

UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR



TRABAJO FIN DE MÁSTER

Recomendación de eventos sociales por texto y colores

Máster Conjunto Ingeniería Informática e i2-TIC

**Autor: Moraleda Ponzola, Miguel
Tutor: Moreno Llorena, Jaime**

FECHA: Septiembre, 2016

Recomendación de eventos sociales por texto y colores

Resumen

Actualmente las interacciones que ocurren entre las personas no solo se producen físicamente, es decir cara a cara, sino también a través de nuevas formas de comunicación, en especial de las redes sociales en internet. Estas interacciones sociales digitales, al igual que ocurre con las no digitales, se llevan a cabo por distintos medios paralelos de comunicación expresando información complementaria que en conjunto completan un mensaje. La importancia de obtener toda la información posible del mensaje, se traduce posteriormente en un mayor conocimiento del tema que se trata en el mismo y del contexto, por lo que es importante tender hacia soluciones multimodales. Actualmente la mayoría de aplicaciones están basadas en texto, dejando sin explotar otros medios de comunicación que los usuarios emiten voluntaria o involuntariamente y perdiendo de esta manera, valiosa información.

En este trabajo, se indaga en una solución para introducir el uso de imágenes dentro de aplicaciones multimodales, permitiendo hacer uso de ellas junto al texto como fuente de conocimiento del usuario. Para ello se van a usar los colores de las imágenes como elemento determinista que permita la clasificación y la inferencia de categorías de una taxonomía. Como temática de clasificación se eligió los eventos sociales, realizando además una clasificación útil para los eventos sociales que se encuentran en internet.

Para validar la investigación realizada y probar su utilidad en una situación real, se realiza una prueba de concepto mediante un robot “bot” de Telegram que permite automatizar la respuesta a eventos en las conversaciones establecidas, almacenando además la información para enriquecer su funcionamiento. En ella se añaden como solución multimodal el análisis predictivo de colores y de texto para la clasificación de eventos sociales. Con estas técnicas se consigue tanto clasificar la información masiva de internet referente a eventos sociales, como la extracción de perfiles de usuario que determinen sus gustos dentro del ámbito de los eventos sociales. Finalmente, el objetivo de la prueba de concepto se orientó hacia la recomendación de eventos a partir de la inferencia de gustos de usuarios.

Los resultados obtenidos demuestran con buenos porcentajes de acierto en las predicciones, que es posible la clasificación de eventos sociales usando colores de imágenes y texto, así como el hecho de poder establecer perfiles de gustos de usuarios a partir de sus interacciones, en una aplicación con soporte de mensajería social, usando estos mismos principios.

Palabras clave

Colores
Análisis de imágenes
Análisis de texto
Recomendación de eventos sociales
Bot Telegram

Abstract

Currently the interactions that occur among people face to face not only physically produced, that is, but also through new forms of communication, especially social networking sites. These digital social interactions, as occurs with non-digital, are held by different parallel media expressing additional information together complete a message. The importance of obtaining all possible information message is subsequently translated into a greater knowledge of the subject is the same and the context, so it is important to move towards multi-modal solutions. Today, most applications are based on text, leaving untapped other media that users emit voluntarily or involuntarily and thus losing valuable information.

In this paper, it is investigated in a solution to introduce the use of images within multimodal applications, allowing to use them next to the text as a source of knowledge of the user. For this will use the colors of images as deterministic element that allows the classification and the inference of categories of a taxonomy. Classification as a theme was chosen social events, in addition to performing a useful social events that are in classification internet.

To validate the investigation and prove useful in a real situation, a concept test is performed by a robot "bot" Telegram to automate the response to events in the established conversations also storing information to enhance their performance. It multimodal solution will combine as predictive analytics and text colors for the classification of social events. With these techniques is achieved both classify the massive internet information concerning social events, such as removing user profiles to determine their tastes within the scope of social events. Finally, the objective of the proof of concept was oriented towards events recommendation from the inference tastes of users.

The results show with good percentages of success in predictions, it is possible to classify social events using color images and text, as well as being able to establish profiles tastes of users from their interactions, in an application support social messaging, using the same principles.

Keywords

Colors

Image Analysis

Text Analysis

Social events recommendation

Telegram bot

Agradecimientos

Agradecer a mi tutor Jaime Moreno la dedicación e ideas aportadas a este Proyecto, sin el cual no habría podido completarse ni llevarse al nivel alcanzado. A la colaboración de Kostadin Koroutchev por guiar el proyecto hacia buen puerto en momentos puntuales y con el cual he aprendido mucho. Al antiguo equipo de ingeniería de Smartvel por hacer de mí el mejor producto que jamás saldrá de esa empresa, enseñándome desde el primer momento y brindándome la confianza y oportunidades con las que cualquier becario soñaría. A mi familia por el apoyo diario, y especialmente a Charly, que sin su alegría habría perdido la esperanza. Y finalmente a Bea, que ha sufrido junto a mí el proyecto y ha proporcionado un apoyo especial, siendo comprensiva con la situación y sin dejar de animar y dar cariño en ningún momento.

Tabla de Contenidos

Resumen.....	i
Palabras clave	i
Abstract.....	ii
Keywords	ii
Agradecimientos	iii
Tabla de Contenidos	iv
Tabla de Ilustraciones	vi
Índice de Tablas	vii
Glosario.....	viii
1. Introducción	1
1.1. Contexto de la investigación.....	1
1.2. Hipótesis de partida	2
1.3. Solución propuesta	2
1.4. Objetivo.....	3
1.5. Ventajas y desventajas	4
1.6. Aspectos no cubiertos o limitaciones.....	5
1.7. Estructura de la memoria.....	5
2. Estado del arte	6
2.1. Estado del arte en el análisis de imágenes sociales	6
2.2. Estado del arte en recomendación social.....	14
2.2.1. Bots de eventos en Telegram	16
2.3. Estado del arte eTurismo.....	16
2.4. Casos de uso del estado del arte	17
2.5. Conclusiones del estado del arte	19
3. Propuesta.....	20
3.1. Requisitos generales funcionales.....	20
3.2. Requisitos generales no funcionales	20
3.3. Análisis y Diseño.....	20
3.3.1. Recolección.....	20
3.3.2. Procesamiento	24
3.3.3. Predicción	28
3.3.4. Servicio.....	34
3.4. Arquitectura	43
3.5. Protocolos	47
4. Desarrollo.....	52
4.1. Proyectos y Repositorios	52
4.2. Tecnologías y lenguajes utilizados	52
4.3. Despliegue	53

4.4.	Dependencias Externas	54
4.5.	Test funcionales y Documentación.....	55
5.	Evaluación.....	56
5.1.	Pruebas funcionales del sistema.....	56
5.2.	Pruebas clasificación y análisis de imágenes.....	56
5.3.	Pruebas clasificación y análisis de texto	59
5.4.	Pruebas bot Telegram	60
6.	Conclusiones	66
6.1.	Conclusiones.....	66
6.2.	Trabajos futuros	66
	Referencias	68
	Anexos	71
1.	Juego de Facebook experimento de filtros Instagram	71
2.	Análisis de filtros Instagram.....	72
3.	Listado de páginas soportadas por el recuperador de eventos	73
4.	Propuesta para el filtrado de pornografía en Instagram	74
5.	Distribución de eventos en categorías	75
6.	Creación de un bot de Telegram	76
7.	Instrucciones de uso para el despliegue	77
8.	Diseño de arquitectura	78
9.	Prueba de uso guiado bot Telegram	79
10.	Encuesta pruebas bot Telegram	83
11.	Resultados de la encuesta de pruebas bot Telegram	84

Tabla de Ilustraciones

Ilustración 1 Módulos de la solución propuesta	3
Ilustración 2 Logo Imgnet	9
Ilustración 3 Relación de imágenes de Imagenet	9
Ilustración 4 Logo Panoramio	9
Ilustración 5 Interfaz de Panoramio	10
Ilustración 6 Logo de Picasa	10
Ilustración 7 Interfaz web de Picasa	11
Ilustración 8 Logo de Snapchat	11
Ilustración 9 Interfaz de Snapchat	12
Ilustración 10 Logo Instagram	12
Ilustración 11 Interfaz de Instagram	13
Ilustración 12 Logo de Flickr	13
Ilustración 13 Interfaz de Flickr	14
Ilustración 14 Análisis de Extracción de imágenes en fuentes de imágenes	21
Ilustración 15 Elementos básicos a definir en un scraper de una fuente de eventos sociales	21
Ilustración 16 Representación modelo de colores HSV	22
Ilustración 17 Tabla de valores de puntuación de feedback	23
Ilustración 18 Calculo del feedback de recomendación	23
Ilustración 19 Procesamiento de imágenes	25
Ilustración 20 Ejemplo visual de SVM	26
Ilustración 21 Ejemplo visual de RBM	26
Ilustración 22 Pipeline de procesamiento de eventos	27
Ilustración 23 Estructura de feedback	27
Ilustración 24 Proceso de análisis de conversaciones	28
Ilustración 25 Entrenamiento de texto	31
Ilustración 26 Formación de set usando test definido	32
Ilustración 27 Formación de sets sin test definido	32
Ilustración 28 Esquema de funcionamiento de algoritmo SVM cascade	33
Ilustración 29 Estructura de comando del bot de Telegram	35
Ilustración 30 Estructura de Inline Query en el bot de telegram	36
Ilustración 31 Diagrama de uso de Inline Query	36
Ilustración 32 Metodología de recomendación	39
Ilustración 33 Diagrama comando events en chat individual	39
Ilustración 34 Diagrama comando events en chat individual	40
Ilustración 35 Diagrama caso de uso inline query	41
Ilustración 36 Diagrama casos de uso inline callback more/less	42
Ilustración 37 Diagrama caso de uso inline callback evaluación	43
Ilustración 38 Arquitectura del sistema	44
Ilustración 39 Arquitectura de servicio del bot	47
Ilustración 40 Protocolo de descarga de imágenes	48
Ilustración 41 Protocolo de carga de imágenes	49
Ilustración 42 Protocolo de descarga de eventos	49
Ilustración 43 Protocolo de actualización de eventos en la fuente	49
Ilustración 44 Protocolo de carga de bordes	51
Ilustración 45 Protocolo de actualización de predictores	51
Ilustración 46 Resultados encuesta interesante vs utilidad	62
Ilustración 47 Resultados encuesta búsqueda ordenador vs facilidad de aprendizaje	63
Ilustración 48 Resultados encuesta recomendación de gustos	63
Ilustración 49 Resultados encuesta referentes a textos del bot	63
Ilustración 50 Resultados encuesta complejidad de la prueba	63

Índice de Tablas

Tabla 1 Hipótesis de búsqueda de artículos de análisis de imágenes	6
Tabla 2 Cadenas de búsqueda de artículos de análisis de imágenes	6
Tabla 3 Tabla resultados estudio de trabajos de análisis de imágenes	7
Tabla 4 Taxonomía de eventos sociales de internet	30
Tabla 5 Inline callbacks permitidos en el bot de Telegram.....	37
Tabla 6 Tabla de Arquitectura de máquinas del sistema	44
Tabla 7 Uso de servicios de Machine Learning en las distintas máquinas	45
Tabla 8 Despliegue de máquinas del sistema	53
Tabla 9 Resultados pruebas de clustering	57
Tabla 10 Grafica pruebas de clustering	58
Tabla 11 Distribución de la clasificación de eventos	59
Tabla 12 Perfiles de pruebas del bot	60
Tabla 13 Gráfico de participantes en las pruebas del bot	60

Glosario

- **Webhook:** es un método de alteración del funcionamiento de una página o aplicación web, personalizando la llamada que esperan unos datos para ejecutar o redirigiendo el flujo de datos de forma personalizada
- **CPU:** unidad central de procesamiento o unidad de procesamiento central, es el hardware encargado de la ejecución de instrucciones dentro de una computadora.
- **ETurismo:** Rama dentro del turismo que se realiza o maneja a través de medios digitales.
- **Machine Learning:** rama de la inteligencia artificial cuyo objetivo es desarrollar técnicas que permitan a las computadoras aprender.
- **Log:** Registro que deja un programa al ejecutarse y que contiene información de la misma.
- **Bot:** Entidad con inteligencia artificial capaz de interactuar con otro tipo de entidad inteligente, ya sea persona o no, y en ocasiones proporciona un servicio.
- **Keywords:** Palabras claves usadas para relacionar y definir fácilmente un concepto.
- **Licencia freeware:** Licencia de comercio de uso libre sin coste impositivo.
- **Hashtag:** Palabra clave e indexada que permite una recuperación rápida de elementos anexionados a ella.
- **Scrapear:** Extracción de información de un medio, ya sea una web, un servidor o un archivo, a otro con una estructura establecida. Los elementos básicos para realizar un scrapeo son: el *crawler*, el *spider* y el *scraper*.
- **Endpoint:** Punto de entrada a un servicio, proceso o sitio web.

1. Introducción

1.1. Contexto de la investigación

Dentro de la sociedad tecnificada en la que vivimos es un hecho el auge que tienen las aplicaciones para soporte de redes sociales. A la hora de clasificar este tipo de redes, se podrían establecer diferencias respecto a la utilidad que tienen, el multimedia que explotan y el tipo de usuario que las usan o al van dirigidas, pero todas ellas tienen en común que consiguen obtener mucha información de sus usuarios. Este hecho ha convertido las redes sociales en un banco de información de valor incalculable, en la cual las empresas invierten en proporcionar funcionalidad y servicios a los usuarios, de forma que estos compartan de forma cómoda y sencilla más aspectos de su vida. Toda esta información obtenida es posible agruparla por temáticas y afinidades, llegando no solo a conocer a los individuos aislados, sino a mostrar la existencia de círculos sociales entre los usuarios, que favorecen e impulsan su participación en dichas aplicaciones. Un usuario habitualmente no suele pertenecer a un solo círculo social, sino que a medida que vuelca información al sistema, va participando en varios de esos círculos, que en definitiva representan la intersección de sus intereses, y mediante el análisis de los mismos se puede determinar la posición que ocupa en ellos y la intensidad de su participación.

En la actualidad están en auge la aplicación de técnicas de análisis del lenguaje natural para identificar dichos intereses y opiniones en los mensajes de texto intercambiados en las redes sociales [4] [1], ya que suele ser el medio de transmisión más habitual que los usuarios utilizan para verter la información. Este medio puede ser el más usado, pero no es el único y no por ello se debe pensar con el análisis del lenguaje la impresión obtenida del usuario es fidedigna o completa. En la actualidad también está en auge el empleo de distintos recursos multimedia junto con el texto tradicional en las redes sociales que llega a ser un medio secundario en algunos casos. Así mismo, están proliferando aplicaciones para redes sociales que permiten compartir recursos multimedia y son numerosas las que están dirigidas expresamente a ello, aunque aún no se han realizado análisis exhaustivos de estos otros medios multimedia más allá del texto. Esto es debido principalmente a los problemas de computación [2] que conlleva el análisis multimedia, la dificultad de la misma y la creencia de que con el texto ya se obtiene una visión suficiente de la información que disponible en estas aplicaciones. Más allá de la explotación de una modalidad u otra de información (texto, imagen, audio, vídeo...), está la utilización combinada de varias de ellas en el análisis, lo que sería optar por una aproximación multimodal, que es una estrategia poco explorada actualmente, aunque seguramente obtendría mayor precisión y calidad de resultados.

Entrando más en el aspecto de la transmisión de la información, se ha demostrado también que las personas por sí mismas, se comunican por diversos canales con distintos propósitos, tanto cuando lo hacen utilizando tecnologías de la información y las comunicaciones (TICs) [3] como cuando no. En una comunicación sin uso de las TICs, la pérdida de información puede ser debida a la incomprensión, falta de atención o simplemente desconocimiento por una de las partes, en cambio en una conversación a través de las TICs depende además de la capacidad de computación de canales que se puedan procesar y enriquecer el flujo de información.

En los últimos años se ha hecho notar la importancia del post-procesamiento masivo de información recopilada (Big Data), dando lugar a grandes beneficios y revelando su importancia [4]. Por lo tanto, es cuestión de tiempo y de avance de las tecnologías que la pérdida de información que ocurre actualmente sea subsanada, aprovechando al máximo toda información aportada por los usuarios. Actualmente se está haciendo uso del resultado obtenido de estas tecnologías, en aplicaciones del ámbito de la recomendación. En ella sistemas autónomos se entrenan con los datos extraídos de las aplicaciones de redes sociales para ser capaces de proporcionar al usuario mejores resultados, experiencia y adaptación a sus características.

Una solución interesante que se está aplicando en la actualidad en la mayoría de aplicaciones con soporte para redes sociales como Facebook, Telegram, etc., es la incorporación de entidades computacionales más o menos inteligentes (IA) como sistemas autónomos que actuarían como un nexo más de estas redes, de forma equivalente a los participantes de las mismas [5]. Estos nexos artificiales procesan el contenido de estas aplicaciones para proporcionar al usuario alguna funcionalidad extra y facilitando con ello una interacción inteligente. Estas entidades que recibirán el nombre de bot, no tienen por qué estar implementados por los desarrolladores de la aplicación, sino que estos dejan abierta esa parte del código para que cualquiera pueda crear sus propios bots de contenidos. Con esta política de abrir funcionalidad a servicios de terceros mediante bots, se consigue la incorporación de nueva funcionalidad que enriquece la aplicación, sin que esta suponga un coste computacional añadido directamente a la misma. En la implantación de este tipo de políticas de integración de servicios de terceros, es recomendable que se realice una mínima supervisión por parte de los responsables de las aplicaciones para evitar problemas de seguridad.

Hasta la fecha se han realizado bots de recomendación para diversas temáticas como deportes, música, política, pero todavía no se han utilizado en el ámbito de eTurismo. La solución propuesta consiste en un bot capaz de tener en cuenta los gustos del usuario y los datos de eventos sociales, para relacionarlos entre sí teniendo en cuenta el momento, la ubicación y las preferencias a la hora de recomendar. Siendo estos tres parámetros mencionados anteriormente fácilmente extraíbles de las aplicaciones con soporte para redes sociales, una solución multimodal que analice todos los medios posibles de transmisión de información, se considera una solución factible e innovadora. La extracción de la información de los eventos requeriría una agregación de la información abundante, aunque dispersa, que existe en internet.

Este proyecto se enmarca como trabajo de fin de los masters de Ing. Informática y de i2-TIC realizados conjuntamente en la Escuela Politécnica Superior de la Universidad Autónoma de Madrid. En él se integran conceptos de los campos de Redes Sociales, Interacción Persona-Ordenador, Big Data y Gestión de Proyectos, entre otras especialidades, siendo conocimientos básicos necesarios para este TFM.

1.2. Hipótesis de partida

Considerando el contexto presentado se plantea el proyecto de investigación de este trabajo considerando las siguientes hipótesis:

- **H1:** Es posible clasificar imágenes sociales mediante el análisis cromático de las mismas.
- **H2:** Es posible clasificar eventos sociales mediante el análisis de textos asociados a ellos.
- **H3:** Es posible realizar una aproximación multimodal para clasificar eventos sociales, considerando imágenes y textos asociados a ellos y cumpliéndose **H1** e **H2**.
- **H4:** Es posible caracterizar a los usuarios de una red social según sus preferencias por eventos sociales, considerando las imágenes y textos que se comparten en los círculos a los que dichos usuarios pertenecen y cumpliéndose **H1** e **H2**.
- **H5:** Es posible realizar una taxonomía general para clasificar los eventos sociales publicados en internet.

1.3. Solución propuesta

De acuerdo al problema expuesto y a las hipótesis de partida definidas, se ha efectuado el análisis y diseño de una solución que es capaz de hacer uso del análisis cromático de imágenes y del lenguaje natural, para servir eventos sociales de internet de forma adecuada al gusto del usuario. La categorización de los eventos, las imágenes y gustos de los usuarios, se realizan mediante procedimientos predictivos. Esta solución propuesta será presentada a modo de prueba de

concepto de la investigación realizada. La solución propuesta estará formada por cuatro módulos principales:

- **Módulo de Recolección autónomo de eventos**, estará dedicado a recolectar eventos de internet de las páginas que se le indique y cuando corresponda, introduciendo nuevos eventos al sistema.
- **Módulo de Procesamiento de eventos**, estará encargado de almacenar los eventos obtenidos y adecuarlos a un formato definido. En el proceso se pasará por un proceso de geolocalización y categorización, dejando los eventos listos para poder servirlos.
- **Modulo Predictor**, concentrará todos los procedimientos de machine learning encargados de predecir elementos de nuestro sistema, así como todo el procesamiento y análisis de imágenes. Este módulo será el que realice los experimentos de investigación correspondientes al proyecto.
- **Bot Telegram**, es el modulo encargado de servir los eventos procesados al usuario y de recolectar información del mismo, para posteriormente poder devolver mejores resultados.

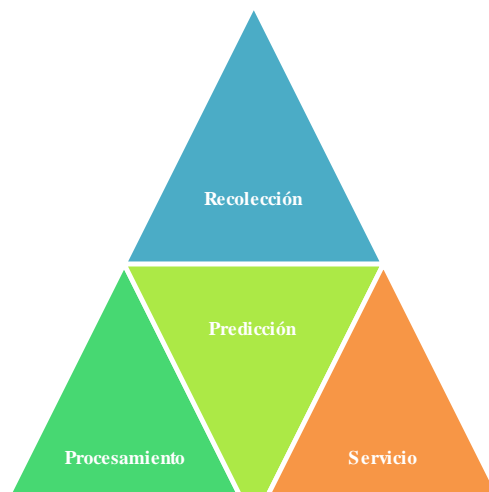


Ilustración 1 Módulos de la solución propuesta

1.4. Objetivo

El objetivo general de este trabajo de fin de máster es indagar en el aprovechamiento de la información multimodal publicada en las redes sociales, en particular de las características cromáticas de las imágenes compartidas con el fin de clasificarlas y etiquetar a los usuarios que las comparten con información relevante.

Para alcanzar este objetivo general se establecen una serie de objetivos más concretos:

1. Clasificar imágenes sociales mediante el análisis cromático de las mismas.
2. Clasificar eventos sociales mediante el análisis de textos asociados a ellos con técnicas de procesamiento de lenguaje natural.
3. Clasificar eventos sociales de forma multimodal, considerando imágenes y textos asociados a ellos.
4. Caracterizar a los usuarios de una red social según sus preferencias por eventos sociales de forma multimodal, considerando las imágenes y textos que se comparten en los círculos a los que dichos usuarios pertenecen.
5. Realizar una prueba de concepto mediante el desarrollo de una aplicación que integre la aproximación propuesta, en particular un recomendador multimodal de eventos sociales, que recolecte y clasifique eventos publicados en internet, que recoja datos de usuarios en

redes sociales para caracterizarlos de forma multimodal considerando los círculos sociales a los que pertenecen y las imágenes y mensajes compartidos en ellos, y que recomiende eventos a usuarios en las propias redes sociales con la información sintetizada, así como la ubicación y el momento particulares.

6. Realizar una taxonomía general para clasificar los eventos sociales publicados en internet

1.5. Ventajas y desventajas

En relación a la solución propuesta, los objetivos planteados y en general a la investigación realizada, las principales ventajas que aporta este trabajo son las siguientes:

- **Análisis imágenes:** el uso de imágenes como elemento predictivo es una gran fuente de información. Las aplicaciones cada vez apuestan más por incorporar funcionalidades relacionadas con la compartición de imágenes por lo que es un filón que se debería explotar.
- **Multimodal:** el explotar los medios de forma multimodal permite extraer mayor información del usuario.
- **Mayor conocimiento del usuario:** Cuantos más atributos se puedan explotar para la extracción de información, más detalles se conocerán del mismo.
- **Categorización de eventos:** El realizar una taxonomía de categorización de eventos sociales de internet que permita clasificarlos puede abrir con el tiempo un estándar para su uso en futuros trabajos.
- **Sistema distribuido y escalable:** Este modelo permite que cada sistema gestione sus competencias y pueda ser escalado en el caso de que se requiera más capacidad.
- **Alimentación dinámica de eventos:** La solución propuesta se alimenta periódicamente de eventos de internet sin necesidad de supervisión.

Y como desventaja posible:

- **Gran coste computacional:** El tener que analizar imágenes conlleva un gran uso de *CPU*, siendo costoso y lento. Por ello se ha ideado la solución pensando en poder escalar todos los módulos.
- **Privacidad:** El recabar más información del usuario sin que él sea consciente, no deja de ser en cierto modo entrometerse en su privacidad a fin de conocerlo más, sin que el haya dado permiso expresamente para ello.
- **Revisión de eventos:** El reconocedor que usa nuestra solución mejorará con el tiempo a medida que se tengan más eventos almacenados. Por ello en el caso de que la solución propuesta sea incapaz de reconocer la categoría de un evento, se deberá realizar revisar a mano.
- **Revisión de imágenes:** Al no existir una base de datos de imágenes sociales y viendo que en las aplicaciones con soporte para redes sociales los usuarios no solo suben imágenes sociales, sino también memes o imágenes de otro tipo, es necesario filtrarlas. Este filtrado en la fase inicial del proyecto necesariamente hay que hacerlo a mano, para poder partir de un entrenamiento aceptable. En futuras actualizaciones del modelo de entrenamiento, se podría usar el propio predictor para filtrar, ahorrándonos tiempo.
- **IP estática:** La solución propuesta hace uso del *webhook* del api de Telegram, la cual requiere una dirección IP estática con la cual establecer comunicación.

1.6. Aspectos no cubiertos o limitaciones

Tras desarrollar la prueba de concepto de este proyecto, han quedado por cubrir ciertos aspectos con los que se debe continuar trabajando:

- Se han usado técnicas de análisis de imágenes y machine learning de forma funcional, sin llegar a la perfección en temas de optimización, de los algoritmos implicados en dichos procesos.
- La recolección y el procesamiento de eventos de forma autónoma por parte del sistema ha quedado sin representación gráfica, por lo que se deberá recurrir a *logs* del sistema, página de administración o línea de comandos, para saber en qué estado se encuentran los eventos dentro del sistema. Aun sin contar con una parte visual que monitorice la actividad del sistema de forma clara, el sistema podrá funcionar con normalidad.
- El recomendador de eventos solo podrá conocer los gustos de una persona a partir de su actividad e interacción en conversaciones grupales, ya que no es habitual que un usuario tenga conversaciones con un bot más allá de intercambiar comandos.

1.7. Estructura de la memoria

La estructura seguida para documentar el proyecto realizado es la siguiente:

La sección 1 contiene la introducción, en la que se cuenta el contexto del proyecto, los objetivos y motivaciones que me llevaron a realizarlo, incluyendo ventajas, desventajas y limitaciones que encontradas o que se espera aportar al tema.

En la 2ª sección se presenta un estudio del estado arte, en el que se tratan varias temáticas. Se aborda la evolución dentro del ámbito del análisis de imágenes, la evolución de la recomendación social, la incorporación de IA en las redes sociales y los eventos sociales de eTurismo en internet.

En la sección 3 se plantea una propuesta basada en mejoras posibles que se han identificado para demostrar las hipótesis planteadas en el proyecto.

La sección 4 contendrá la parte más técnica del proyecto, donde se contarán aspectos del desarrollo llevado a cabo para la prueba de concepto.

En la sección 5 se incluye una planificación detallada de las pruebas a realizadas a fin de demostrar la validez de nuestro trabajo. Las pruebas técnicas se han desarrollado integra y satisfactoriamente. También se prevén realizar pruebas

En la sección 6 se presentan las conclusiones y trabajos futuros, donde se hace una valoración de las ideas extraídas de nuestro proyecto, una interpretación de los resultados obtenidos en aplicación realizada y un esbozo de trabajos futuros con los que mejorar y seguir la línea de este proyecto.

Por último y para cerrar este documento, se incluyen las referencias consultadas, así como una serie de anexos técnicos en relación con el proyecto o de documentación paralela.

2. Estado del arte

2.1. Estado del arte en el análisis de imágenes sociales

Dada la dificultad encontrada para realizar la búsqueda de publicaciones relevantes para este trabajo dentro de la temática de análisis de imágenes sociales, se va a realizar un estudio de varios repositorios, intentando ser lo más exhaustivo posible.

2.1.1. Imágenes sociales

Como primera medida en esta búsqueda será definir el concepto que se busca de imagen social. En las fuentes de información visitadas no se ha encontrado definido el concepto como tal, por lo tanto, en este trabajo se va a realizar una propuesta de definición. Este tipo de imagen es representativa de eventos en las que pueden participar de una a varias personas en la realización de una actividad. Por el mero hecho de que exista interacción de varias personas no se considera una imagen como social, sino que es necesario que la imagen en sí de referencias que indiquen que se está realizando un tipo concreto de actividad. Estos eventos pueden ser desde actividades del día a día espontáneas hasta eventos organizados, pero siempre permitiendo clasificarlas con claridad dentro de un tipo de evento y representando la realidad. Es importante recalcar también que las imágenes sociales deben ser de medios digitales para establecer ese contexto común, ya que podrían existir diferencias al comparar con imágenes de otros contextos.

2.1.2. Hipótesis de búsqueda

Se van a definir una serie de hipótesis de búsqueda, que definan un horizonte al que nuestra investigación de campo debe llegar y seguir como criterio para la consecución de un estudio válido. Las hipótesis definidas son las siguientes:

1.	Es posible clasificar imágenes sociales usando los colores.
2.	La clasificación de imágenes sociales revela información del usuario.

Tabla 1 Hipótesis de búsqueda de artículos de análisis de imágenes

2.1.3. Cadenas de búsqueda

Para realizar la búsqueda de trabajos dentro del mundo académico relacionados con la temática del análisis de imágenes, y teniendo en cuenta las hipótesis de búsqueda anteriormente definidas, se han creado una serie de cadenas de búsqueda:

$C1 = \langle \text{color} \rangle + \langle \text{image} \rangle + \langle \text{concursos} \rangle + \langle \text{atributos} \rangle$ $\langle \text{image} \rangle = [[\text{social} + \text{histogram} + \text{tag}] * \text{image}]$ $\langle \text{color} \rangle = [[\text{clúster}] * \text{color}]$ $\langle \text{concursos} \rangle = [\text{Imagenet} + \text{Imageclef}]$ $\langle \text{atributos} \rangle = [\text{analysis} + \text{classification} + \text{classifier} + \text{classify} + \text{trie}]$
--

Tabla 2 Cadenas de búsqueda de artículos de análisis de imágenes

La búsqueda se realizó usando cadenas de texto en inglés. Dentro de las palabras elegidas para formar las cadenas se han incluido referencias a concursos de clasificación de imágenes (*Imageclef*, *Imagenet*), metodologías de indexación de patrones habitualmente usadas en imágenes (*Trie*) y acciones relevantes con las actividades que se van a realizar en el proyecto (*classification*, *analysis*, *histogram*).

2.1.4. Criterio de evaluación

Las fuentes digitales utilizadas como repositorios de consulta fueron las siguientes:

- Google Scholar
- ACM Digital Library
- Springer Link
- IEEE Xplore Digital Library

2.1.5. Criterio de evaluación

Para la evaluación de los resultados obtenidos de las búsquedas, se analizarán todos los artículos obtenidos, dando una puntuación de 0 si no son relevantes, y de 1 si lo son. Con lo que un artículo que entre en más de un criterio de validez, ya será considerado como apto. Los criterios de validez definidos son los siguientes:

- **Extracción de colores**
 - 0 - Inválido: El artículo no hace referencia a la extracción de colores o histograma en su contenido.
 - 1 - Válido: El artículo hace referencia a la extracción de colores o histograma en su contenido.
- **Clusterización de colores**
 - 0 - Inválido: El artículo no hace referencia a la clusterización de colores.
 - 1 - Válido: El artículo hace referencia a la clusterización de colores.
- **Análisis Imágenes sociales**
 - 0 - Inválido: El artículo no hace referencia a la clusterización de colores.
 - 1 - Válido: El artículo hace referencia a la clusterización de colores.
- **Concurso de imágenes**
 - 0 - Inválido: El artículo no hace referencia a un concurso de imágenes.
 - 1 - Válido: El artículo hace referencia a un concurso de imágenes.

2.1.6. Resultados

Los resultados obtenidos al realizar las búsquedas organizados por los criterios definidos anteriormente fueron los siguientes:

critérios	Extracción colores	Clusterización colores	Imágenes sociales	Concurso
# artículos	5	2	2	0

Tabla 3 Tabla resultados estudio de trabajos de análisis de imágenes

Los resultados obtenidos de las búsquedas con las cadenas de concurso, devolvieron resultados, pero ninguno relevante ya que no casaba con ningún otro criterio de validez. En cuanto al criterio de validez de la clusterización hay unos cuantos trabajos que no revelan explícitamente que metodología han usado, por lo tanto y ante la duda no serán tomados en cuenta.

2.1.7. Conclusión de la búsqueda

El estudio realizado revela la existencia de bastantes trabajos enfocados en el análisis y reconocimiento de expresiones y acciones físicas de los usuarios, durante la realización de acciones sociales, extrayendo de estos sentimientos o emociones de los usuarios. Dentro del mundo académico del análisis de imágenes, parece que se progresa lentamente y se usan concursos a nivel mundial (Imagenet, Imageclef) para tratar de impulsar su avance. Estos concursos suelen ir enfocados a temas médicos [6], plantas [7] o de animales, en los cuales sus aplicaciones pueden proporcionar mayores beneficios económicos. Pocos trabajos se establecían dentro del ámbito de nuestros criterios de búsqueda. La mayoría de estos si trataban el tema de la extracción y análisis de patrones cromático [8] [9] [10], pero solo se consiguieron encontrar dos trabajos en los que se llegara a relacionar la composición cromática de la imagen con atributos del usuario [11] [12]. Esto pone de manifiesto que en este ámbito se prima más el desarrollo técnico y el surgimiento de la técnica por encima de la aplicación de esta. Esto posiblemente sea debido al coste computacional del tratamiento de imágenes, pero no por ello se debe desechar esta tecnología, ya que se pueden sacar grandes beneficios de su empleo.

Respecto a las hipótesis de búsqueda planteadas, queda patente de que sí es posible clasificar una imagen mediante sus propiedades cromáticas, pero no se han realizado trabajos usando imágenes sociales. En cuanto a la información que se puede extraer de las imágenes, en los trabajos encontrados se trataron de inferir, el sexo de la persona que había tomado la imagen y los sentimientos que representaba esta.

2.1.8. Sitios web especializados en imágenes

Actualmente existe un auge en el intercambio de imágenes por internet y esto ha hecho proliferar el surgimiento de aplicaciones con soporte para redes sociales que explotan este medio. Este tipo de aplicación mantiene los mismos principios [13] que se podrían identificar en aplicaciones con soporte para redes sociales de otra índole, simplemente que en las que nos ocupan se le da una mayor importancia a la cuestión de las imágenes. Los sitios web dedicados a imágenes que carecían de características sociales y deseaban ser algo más que una base de datos, se han ido poco a poco actualizándose para proporcionar a sus usuarios funcionalidades sociales que les permita no solo subir imágenes, sino compartirlas, comentarlas e interactuar con los demás usuarios. Dentro de las diferentes aplicaciones enfocadas a imágenes, se observan claramente las diferencias de funcionalidad entre ellas, que es lo que acaba definiendo su propósito y el uso que le da el usuario. Dentro del estudio de campo realizado de este tipo de aplicaciones, se buscaron varias características con el fin de encontrar una buena base de datos de imágenes para entrenar los modelos predictivos. Dichas características son las siguientes;

- **Imágenes sociales:** Se buscan imágenes que representen espontaneidad y actividad dentro de ámbitos sociales reales, no se busca usar las típicas imágenes genéricas.
- **Volumen de imágenes:** Es necesario que la aplicación tenga un gran volumen de imágenes, así como actualizaciones periódicas que añadan nuevas cada poco tiempo y enriquezcan renovando el repositorio.
- **Imágenes calidad:** Es importante, aunque no excluyente que las imágenes estén lo menos tratadas posibles para obtener imágenes lo más fidedignas a la realidad en cuanto a colores.
- **Categorización:** La categorización de las imágenes en la aplicación es importante para poder elegir la categoría que se desea obtener y reducir el tiempo de filtrado manual.
- **Fácil obtención:** La descarga de imágenes masiva debe ser todo lo sencilla posible con límites altos de descarga para que la obtención de imágenes no suponga una ralentización del proceso.

Las aplicaciones especializadas en imágenes más interesantes que han sido analizadas para este proyecto han sido las siguientes:



Imagenet

Base de datos de imágenes organizada de acuerdo a la jerarquía WordNet [14] y usada en el campo de la investigación para experimentos en los que se relacionan imágenes con palabras a varios niveles taxonómicos. La jerarquía de WordNet da a esta base de datos, la opción de obtener una descripción interesante de una imagen recopilando *keywords* y categorías al recorrer su árbol jerárquico. El uso de estas imágenes es de libre acceso, pero requiere de la autorización de la universidad de Princeton, en el caso de querer realizar descargas masivas.

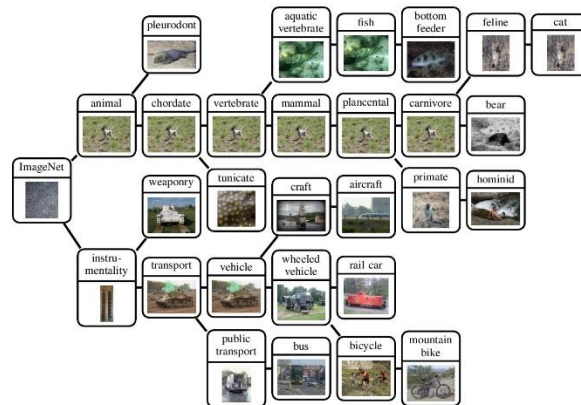


Ilustración 3 Relación de imágenes de Image net

A pesar de ser una aplicación interesante que aún a jerarquías taxonómicas de palabras e imágenes en una base de datos con una gran cantidad de imágenes, se ha descartado su uso por el hecho de ser imágenes genéricas. Se hizo una pequeña prueba de predicción, obteniéndose malos resultados tanto en la predicción (20%) como en su uso como conjunto de test. El hecho de tener una gran variedad de imágenes genéricas sobre una categoría produce dispersión en el espectro de colores confundiendo al predictor y obteniendo malos resultados.



Ilustración 4 Logo Panoramio

Panoramio (Google)

Sitio web creado por dos españoles alicantinos que pretendía crear un mapa mundial con las imágenes de paisajes de sus participantes. En 2007 Google compró el proyecto incorporándolo a Google Maps. En 2014 Google anunció que el cierre del proyecto, incorporando su base de datos y funcionalidad a Google Views.



Ilustración 5 Interfaz de Panoramio

El servicio que ofrecía esta aplicación, consistía en la compartición de fotos geolocalizadas en un punto con el resto de usuarios. Estos podían comentarlas y establecer conversaciones entre ellos a partir de las fotos, pero todo giraba siempre manteniendo el foco en las fotografías y geolocalizaciones por encima de los usuarios. El uso de las fotos con fines profesionales dependía directamente del permiso del fotógrafo. No se ha intentado hacer uso de esta aplicación debido a la temática no social de las imágenes (paisajes).



Ilustración 6 Logo de Picasa

Picasa (Google photos)

Aplicación informática creada por Lifescape en 2002, pasando a las manos de Google en 2004 con licencia *freeware*. Actualmente Google ha anunciado su cierre para incorporar toda su funcionalidad a Google Fotos. Esta aplicación es usada para la edición de imágenes, además de ser una herramienta web para la visualización, almacenaje, la compartición de las mismas. La aplicación está perfectamente integrada en el universo de Google, con todas las posibilidades sociales de compartición y almacenaje gratuito que ofrecen los productos de esta compañía.

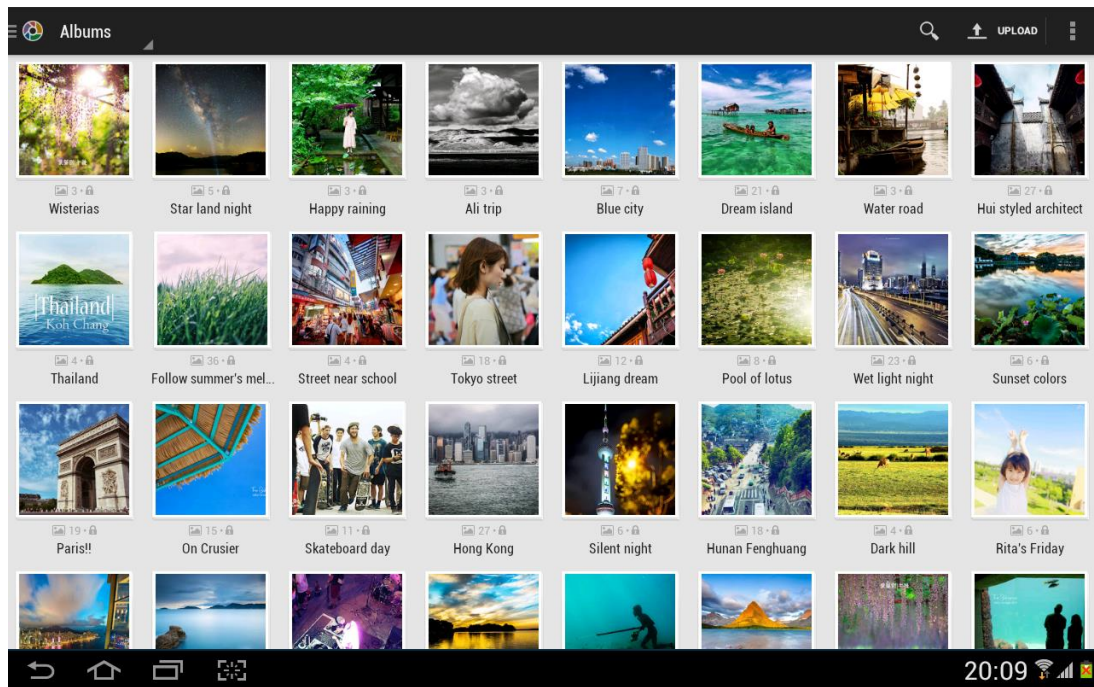


Ilustración 7 Interfaz web de Picasa

A la hora de darle un uso de terceros más allá de lo que ofrece la aplicación, sigue la política celosa de Google de resistencia a la bajada masiva de contenidos de sus plataformas, por lo que, a pesar de tener una gran cantidad de contenido, clasificado y con comentarios de usuarios, las limitaciones para descargar descartan esta plataforma como válida para nuestro proyecto.



Ilustración 8 Logo de Snapchat

Snapchat

Es una aplicación con soporte para red social para Smartphone y Tablet, con varias particularidades que la hacen ser una app de moda. Los mensajes que se comparten en esta aplicación son imágenes o video, a los que se les puede añadir complementos, ya sean texto, stickers o filtros, y tienen la particularidad de supuestamente no dejar rastro en el dispositivo receptor, teniendo unos pocos segundos para visualizar el mensaje. El uso por parte de famosos, junto a la actualización diaria de filtros, la privacidad y la cantidad de usuarios que tiene, la han disparado a ser una de las apps más usadas del momento.



Ilustración 9 Interfaz de Snapchat

Debido a la privacidad de la que hace gala, así como del envío de fotos con filtros y otros complementos, no deja opción para su uso en este proyecto, pero no deja de ser una gran base de datos de fotos diarias de sus usuarios, que internamente podría usar Snapchat para realizar estudios interesantes.



Ilustración 10 Logo Instagram

Instagram (Facebook)

Es una aplicación con soporte para red social, que permite la compartición de imágenes y video entre sus usuarios. La aplicación, propiedad de Facebook desde 2012, fue pionera en la incorporación de filtros fotográficos con éxito. Aprovechando el tirón de los filtros y añadiendo la idea de *hashtags* de Twitter, la visión de los desarrolladores era la de crear un producto similar a Twitter, pero con imágenes. Con el paso del tiempo y viendo la aceptación en la sociedad, ha derivado en una red social con perfiles profesionales de usuarios que se ganan la vida subiendo fotos patrocinados por marcas y usuarios normales que cuenta su vida en forma de imágenes, texto y hashtags.

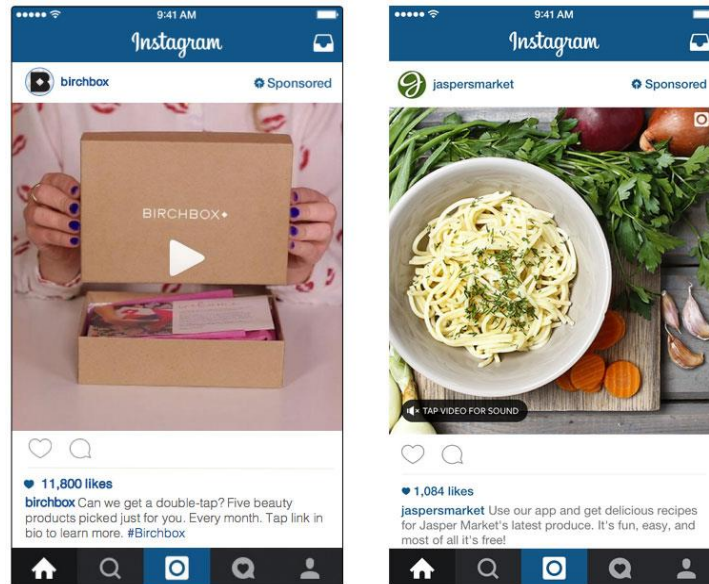


Ilustración 11 Interfaz de Instagram

La cantidad de imágenes diarias subidas a esta red social es enorme, de las cuales según un estudio realizado (Ver anexo [Análisis de filtros Instagram](#)) un 30% de las imágenes llevan un filtro incorporado. Esto deja un 70% de imágenes útiles para nuestro propósito, pero al igual que ocurría con Google y Picassa, Facebook en este caso, es celoso de dejar libre acceso a sus contenidos teniendo que recurrir a la picaresca. Otro inconveniente, es la temática de las fotos y la composición de las imágenes, ya que más de un 40% podrían ser identificadas como *selfies*. Este tipo de composición hace que se pierdan píxeles de información de la imagen ocupadas por la cercanía de los rostros, y no solo eso, sino que muchas veces las imágenes podrían considerarse como *selfies* de comida, ropa o elementos cuya extracción de información útil no queda clara. Otra sorpresa que da esta aplicación es el alto contenido pornográfico existente, se debe suponer que se realizan tareas de revisión de contenidos, revisando si estos son adecuados a sus códigos éticos más allá de las cuentas de usuario famosas o profesionales. El hecho de encontrar imágenes pornográficas durante las pruebas realizadas de descarga masiva de imágenes de esta aplicación, se quiere pensar que es debido a que son imágenes publicadas sin haber pasado aún por sus filtros de revisión (Ver anexo [Propuesta para el filtrado de pornografía en Instagram](#)).



Ilustración 12 Logo de Flickr

Flickr (Yahoo!)

La última aplicación candidata que se analizó en el estudio fue Flickr. A efectos prácticos es parecida a Picasa (Google Fotos), se usa de contenedor de imágenes de los usuarios facilitándoles la compartición y funcionalidad social. Tiene un gran número de imágenes en su base de datos y al usarse como repositorio, se encuentran imágenes puntuales de eventos sociales, más que imágenes del día a día.

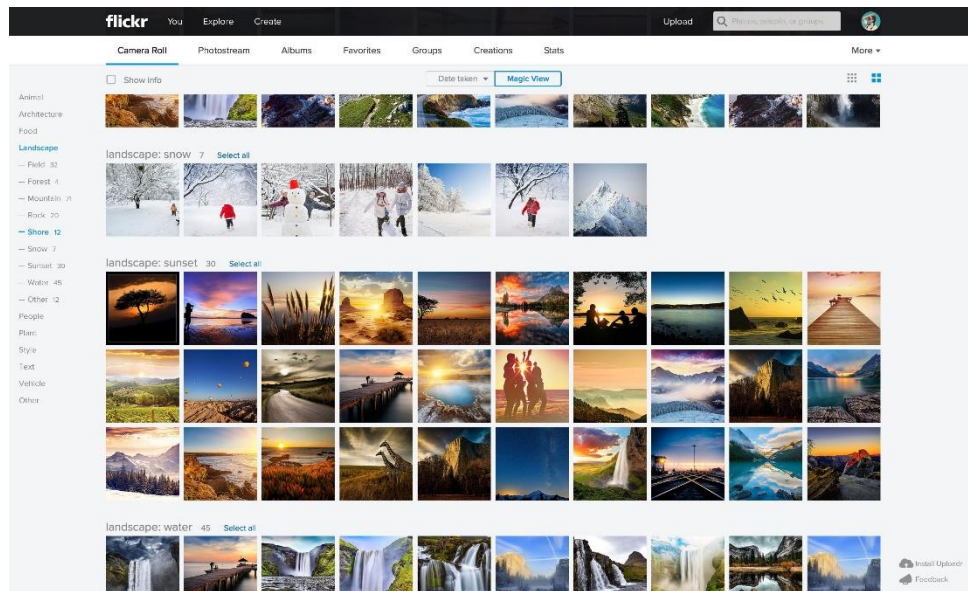


Ilustración 13 Interfaz de Flickr

Flickr es la aplicación que mejor recoge todas las necesidades que se pedían. Esta fuente contiene imágenes sociales, en gran volumen, de buena calidad, de más o menos de fácil extracción y con palabras clave para facilitar su búsqueda.

2.2. Estado del arte en recomendación social

El tema de recomendación hoy en día es de los más populares entre los investigadores y se encuentran muchísimos trabajos al respecto. En este caso de todos los ámbitos existentes dentro del campo de la recomendación se va a centrar en dos temas principales, la recomendación de eventos sociales y la recomendación con bot, que son los que atañen a este proyecto.

Dentro de la recomendación social de eventos se observa que los trabajos existentes realizan recomendaciones sobre los siguientes atributos:

- **Localización**, este atributo permite conocer la posición en la que ocurre un evento y a sabiendas de la localización del usuario el sistema recomendador es capaz de devolver aquellos más cercanos [15]. Hoy en día conocer la localización de un usuario es relativamente sencilla con los dispositivos móviles existentes, pero incluso en dispositivos fijos mediante el uso de internet, se podría obtener una posición relativa. La localización en cuanto a eventos sociales se refiere, es un atributo muy importante a tener en cuenta ya que la asistencia de una persona a un evento dependerá de la distancia a la que se encuentre del evento y la afinidad que este tenga con el evento, decidiendo así si merece la pena recorrer esa distancia. Por lo tanto y teniendo esto en cuenta, es importante realizar recomendaciones en distancias razonables o en todo caso preguntar al usuario cuanto está dispuesto a desplazarse [16].
- **Fecha del evento**, este atributo permite conocer cuándo ocurrirá el evento, localizándolo en el tiempo. Es importante recalcar que un evento tiene una fecha de inicio y otra de finalización, siendo importante a la hora de realizar recomendaciones el uso de una u otra dependiendo del momento. En el caso de usar la de inicio se dejando claro cuando es el primer momento en el que se puede acceder al evento y en caso de usar la de finalización, se informa sobre el momento límite a partir del cual un evento deja de estar disponible. En cualquier caso, es necesario aclarar este concepto al usuario ya que puede llevar a

error. Los trabajos encontrados referentes a atributos temporales, fijan el camino en la posibilidad de servir información de eventos en tiempo real. Este hecho supone un gran salto de calidad y añadiendo la necesidad a los recomendadores de ser capaces de rastrear e identificar eventos sociales de forma rápida [17].

- **Categoría del evento**, este atributo permite clasificar un evento y realizar recomendaciones por etiquetas previamente identificadas. Dentro de los trabajos encontrados, no se encontró referencias a ningún estándar o taxonomía a seguir en cuanto a clasificaciones de eventos sociales. Dentro del mundo empresarial, como se aprecia en distintas aplicaciones de éxito y referentes en el sector como *Eventful* [18], *Ticketea* [19] o *Guiadelocio* [20], entre otras..., cada una ha realizado su propia clasificación acorde a sus necesidades. Los trabajos realizados referentes a la clasificación de eventos recurren al análisis de texto natural para realizar la clasificación de estos [21] [22].
- **Datos del usuario**, este atributo está basado en la cantidad de información que es posible recabar de un usuario de distintas fuentes, a fin de conocerle mejor. El agregar información de terceros da una perspectiva clasificada de información para un usuario y ámbito concreto. Por ejemplo, en [23] se observa que se obtienen datos de su actividad física a partir de sus registros de Nike+ [24], gustos de canciones o temas de lectura de Mylifebits [25], etc. Y de esta manera haciendo uso de técnicas de Big Data puedes agregar grandes cantidades de información de terceros en favor de una recomendación más personalizada al usuario [26]. Los wearables en este caso pueden ser una gran fuente de información y abrir un abanico de posibilidades que no se han tenido en cuenta hasta ahora.
- **Participantes**, este atributo permite conocer que personas implicadas en el evento social. Es habitual que las personas participantes en un evento, tenga gustos afines y en principio el evento al que asisten debería ser de su agrado. Dentro del concepto de participantes, se pueden distinguir dos tipos, los asistentes al evento y los organizadores o implicados en el evento. Se han realizado trabajos en los que se analizaban distintos perfiles de participantes y a partir de estos se recomendaba sobre a qué evento acudir, que evento podría ser acorde al gusto de todos los acompañantes o para un evento concreto, incluso realizar recomendaciones sobre que participantes son los más indicados para asistir [27].

En los trabajos relativos a bot recomendadores, se observa que la dinámica suele ser que estos bots son el fruto de otros trabajos más grandes y son usados como prueba de concepto para probar su funcionalidad en un ámbito real. Es habitual que estos bots tengan una pareja, y uno proporcione la utilidad desarrollada en la investigación y el otro sirva más de recolector de información. Por lo tanto, se pueden establecer dos ámbitos necesarios a la hora de realizar un bot recomendador, uno el ámbito de alimentación con información y otro el del servicio. Un ejemplo de esta metodología es el bot de *health.io* [28] que fue construido a partir del aprendizaje extraído en temas de conversación humana, de otro bot de Telegram y un proyecto de investigación médico de machine learning. Este bot infiere enfermedades y diagnostica posibles causas de malestares en las personas bajo consulta. El entrenamiento del otro bot fue usado para que supiera responder a las personas y la interacción fuese más fluida.

El resto de bot recomendadores que van surgiendo se basan en análisis en tiempo real o uso de otras funcionalidades analíticas para hacer recomendaciones de forma autónoma. Este es el caso de *x.ai* [29], es un bot que hace las veces de secretaria recomendando fechas para quedar con una persona. Es un bot que es capaz de entender *email*, pero su recomendación no va más allá de encontrar un hueco, palabras clave y sacar un resultado posible.

2.2.1. Bots de eventos en Telegram

Telegram es una aplicación de soporte para redes sociales surgida en 2013, que va más allá de ser simplemente un cliente de mensajería. Gracias a su código open source y a sus características en temas de privacidad y seguridad, han consagrado esta aplicación como la clara rival de WhatsApp (Facebook). Dentro de la plataforma de Telegram, se ha realizado un estudio de campo de los bots de eventos existentes. En él se experimentó el tipo de funcionalidad que realizaban, la interacción y la recomendación que realizaba en el caso de que la hiciera. Los bots más relevantes encontrados fueron los siguientes:

@vayaeventbot [30]

Este bot es un gestor y recomendador de eventos sociales. La funcionalidad implementada permite al usuario crear eventos propios y compartirlos con el resto de usuarios. El recomendador hace uso de la localización para mostrar al usuario eventos cercanos a su posición.

@geteventsbot [31]

Este bot es simplemente un servidor de eventos, al cual se le introduce el nombre de una ciudad y devuelve eventos que ocurren dentro de ella. Este evento no pudo ser probado ya que no funcionaba con las ciudades introducidas.

@maxappbot [32]

Este bot es un coordinador de eventos, en el cual se crea un evento, se añaden participantes a la conversación y se encarga de recordar a los asistentes la fecha y lugar del evento. No realiza ningún tipo de recomendación, simplemente hace la función de recordar.

@goalbot [33]

Este bot es un servidor de información referente a eventos deportivos. Este tipo de bot es común dentro de Telegram y su única función es la de recibir información actualizada desde el servidor central. Se lleva a cabo la estrategia *master – slave*, en la cual cada cierto tiempo el esclavo pregunta al maestro sobre actualizaciones de los eventos elegidos por el usuario y este devuelve esa información. En este caso no hay recomendación, solamente servicio de información.

En resumen, los bots existentes realizan tareas de servicio de información, sin ser autónomos ni realizar tareas inteligentes. Uno de los analizados contaba con tareas de recomendación, pero siempre de eventos de su propia base de datos, en ningún caso se tienen en cuenta eventos externos. El hecho de añadir tareas de recomendación inteligentes o llegar a incluir funcionalidad que permita al bot actuar más autónomamente sin depender tanto del usuario, daría a los bot una perspectiva más humana, mejorando su servicio e interacción con los usuarios.

2.3. Estado del arte eTurismo

El concepto de eTurismo se suele relacionar con las actividades del sector servicios relacionadas con el turismo, pero desarrolladas o gestionadas a través de medios digitales. Hoy en día este concepto ha ido creciendo, englobando actividades más allá del turismo como pueden ser los eventos sociales, como por ejemplo conciertos, deportes, restaurantes...etc. pasando de un simple servicio privado de reserva a un cúmulo de información global (Big Data) que en muchos casos contienen información útil e implícita de los usuarios. En este mundo actualmente, se está buscando la mejor forma de explotar toda esta cantidad de información acumulada, a fin de mejorar el servicio al usuario y llevar a cabo nuevos productos. Los puntos clave de esta explotación del Big Data [34] en el eTurismo son los siguientes:

- **Destino Inteligente:** Del popular término de “*Smart city*” se deriva el destino inteligente como su aplicación al turismo. Este término consiste en la necesidad de que el turista no tenga que preocuparse por nada a la hora de recabar información sobre una localización a la que piense ir. Esto engloba desde localización de las cosas, puntos de interés, cultura local... entre otras. La idea de que un destino te proporcione ayuda y sepa adaptarse a un usuario, mejoraría la experiencia considerablemente.
- **Opiniones y Análisis de mercado:** El uso de la información que dejan los usuarios al usar un sistema de eTurismo, es muy valiosa. Este puede ser usada para realizar estudios conjuntos con más usuarios a fin de mejorar el servicio o modificar parámetros del mismo. Analizando esta información no se consigue otra cosa que conocer mejor al usuario, fidelizándolo al saber cómo proporcionarle una mejor experiencia en el siguiente uso que realice del servicio proporcionado.
- **Interacción persona-ordenador:** Se está llegando a una realidad en la cual, para efectuar trámites de reserva o gestión de servicios, se prefiere interactuar con una máquina antes que con una persona física. Este hecho ocurre sobre todo en personas de la franja de edad menor a 60 años. El avance de la tecnología y el diseño, han puesto al servicio del usuario los medios necesarios para proporcionarle una experiencia rápida y cómoda en este sentido. Este hecho explica el auge de los kioscos-totems interactivos y las apps móviles, dentro del eTurismo.
- **Unificar contenidos:** En este sector cada empresa privada en el pasado ha implementado su solución propia para dar un servicio al cliente. Esto ha provocado una dispersión de servicio, con unas metodologías y funcionalidades diferentes a las que el usuario ha debido adaptarse. El surgimiento de aplicaciones que proporcionen una agrupación ordenada y selectiva de estos servicios acorde con las necesidades del usuario, no solo mejora la experiencia del usuario, sino la visibilidad, comodidad y calidad del servicio.

2.4. Casos de uso del estado del arte

Dentro de las tecnologías investigadas se puede ver que cada una de ellas por separado está enfocada a un ámbito específico y abarcando distintos casos de uso o aplicaciones. A continuación, se va a realizar un estudio de los casos de usos individuales de cada una de las tecnologías estudiadas de, y las posibilidades que tendrían juntas.

Análisis de imágenes

El análisis de imágenes es usado sobre todo en actividades relacionadas con el reconocimiento de patrones en las mismas, enfocadas especialmente en gestos, emociones [35] o presencia [36]. Este tipo de aplicación permite la extracción de información residente en las propias imágenes a través de un procesado, ya que por sí solas no sería posible la obtención de dicha información. El fin de todas estas aplicaciones es el dotar a una máquina la capacidad de no solo ver, sino también de entender lo que ve, proporcionando facilidades y otras posibilidades más al usuario a la hora de interactuar persona-ordenador. Actualmente se trabaja sobre todo el reconocimiento en imágenes médicas y animales. Más allá de este ámbito del reconocimiento de patrones, también se realizan análisis de imágenes a partir de los metadatos que esta lleva está incorporada. Esta información puede ser usada más allá del ámbito de las imágenes, pudiendo llegar a contener información sensible como la localización de donde fue realizada la foto entre otras. Estos metadatos suponen un aporte a la par que una posible difusión de información, por lo que se recomienda su limpieza tras el procesado y uso de esta meta información o la restricción del

acceso a fin de preservar su integridad [37]. El procesamiento de este tipo de análisis se realiza sobre todo en indexadores de información o aplicaciones enfocadas para redes sociales.

Recomendación social

En el ámbito de la recomendación social es posible encontrar desde banners inteligentes en páginas web alimentados a través de cookies hasta buscadores inteligentes que se retroalimentan de sus propias búsquedas. El principio de uso principal de este tipo de tecnología es el de basándose en una información que refleje la actuación de un usuario, extraer gustos o ser capaz de establecer un perfil concreto y explotarlo proporcionando recomendaciones acordes. El tipo de perfil extraído depende completamente de la temática de recomendación, aunque no quita que la unión de perfiles de distintos ámbitos pueda mejorar la recomendación formando un perfil más completo o revelando información interesante. Los usuarios suelen beneficiarse de este tipo de servicio de manera ajena a su conocimiento para evitar producir un sentimiento de invasión u observación incomoda constante. Otro ámbito interesante de la recomendación social, en auge con el surgimiento de las aplicaciones con soporte para redes sociales, es la recomendación colaborativa [38]. Este tipo de recomendación es ajeno a reglas de recomendación y basado su éxito y difusión únicamente en el apoyo recibido por parte de los usuarios. Por lo tanto y teniendo este dato en cuenta, se observa que dentro del ámbito de la recomendación social se cuenta con entidades informáticas programadas que realizan este tipo de recomendaciones y, por otro lado, usuarios implicados en opinar y difundir eventos sociales, realizando de esta manera su recomendación particular.

Bots

El uso de bots habitualmente ha estado enfocado en temas de monitorización, gestión o tareas efímeras que consumieran la entidad y se destruyeran tras su uso. Solían ser programas efímeros encargados de la realización de tareas livianas y una vez finalizaban eran eliminados. Tenían aplicaciones en ámbitos de scrapping y la seguridad, siendo un tipo de ataque común.

En la actualidad los bots cada día van teniendo más inteligencia, capacidad de interacción e importancia usándose como interfaz inteligente para proporcionar servicios en distintas plataformas. El enfocar estos bot a fines más socializadores con los usuarios, ha enriquecido su uso y sus aplicaciones. La posibilidad de incorporar bots a chats permite como ya se dijo anteriormente la inclusión de servicios de terceros en la aplicación en cuestión, consiguiendo así una integración sin coste alguno de nuevas posibilidades. El uso de bots sociales en cuanto a tipo de usuarios se refiere, es más usado entre la franja de edades de los 18 a los 35, y enfocado normalmente a la temática del ocio, pero ya se están desarrollando bots enfocados a personas con dificultades especial u otras temáticas como ocurren en [39], que está enfocado al e-learning.

eTurismo

El eTurismo como ya se dijo anteriormente se encuentra en un momento de expansión. Actualmente está enfocado sobre todo a la búsqueda reserva y promoción de ocio, siendo mucho mayores las posibilidades que esta temática ofrece. El uso del BigData en este ámbito permitirá establecer mejores perfiles de usuario y servicios, enfocándose en el desarrollo de nuevas posibilidades y un mejor servicio. El tipo de usuario que usa este tipo de aplicaciones es variado, aunque prima el perfil que busca encontrar una oferta de ocio más barata de su precio habitual, ya sea por reventa u oferta. Desde el punto de vista profesional del sector, cada vez es más recurrente el abrir servicios a las nuevas tecnologías que surgen. Es habitual que vayan de la mano con emarketing, siendo un panorama de individualismos en el que cada uno mira por su servicio y realiza su solución personalizada. Actualmente la necesidad profesional de este sector reside en obtener datos del cliente a fin de hacer una base de datos que les permita establecer perfiles de sus clientes. Para ello recurren a Spam, formularios o historial de datos de visitas anteriores.

2.5. Conclusiones del estado del arte

Como se vio en las distintas secciones del estado del arte, la temática del proyecto parece acertada e innovadora. Aunar en un proyecto clustering de colores con recomendación y eventos sociales, parece una mezcla interesante, cuya prueba de concepto podría tener varias aplicaciones comerciales, con la posibilidad de conseguir un beneficio económico futuro. El uso de los bot de Telegram también parece una solución acertada, tanto por plataforma como por estado de la tecnología en la actualidad, ya que innovar haciendo un smart-bot que incluya ámbitos de recomendación inteligente mejorando la interacción y el servicio parece el siguiente paso dentro de la generación actual de bots.

Si cabe destacar algo extraño dentro del estudio realizado sería lo oscuro que, a opinión del autor de este trabajo, se ha mostrado el mundo del análisis de imágenes. Parece un mundo en el que se avanza a base de incentivos de concursos y en cuanto se consigue un avance, esa persona es contratada por una gran multinacional desapareciendo todos sus futuros trabajos. La búsqueda de trabajos de esta parte fue tediosa, ya que la mayoría de resultados interesantes estaban ocultos bajo otros resultados que salían al usar las cadenas de búsqueda establecidas. Esto era debido a que los trabajos de este ámbito casi no tenían citas o visitas, lo cual hacía que, al buscar, aunque la sentencia de búsqueda fuera correcta, quedarán retrasados estos resultados en pos de otros. Se pidió la ayuda a Kostadin Koroutchev, profesor de la EPS (UAM) con conocimiento avanzados en la materia, para confirmar que las búsquedas que se realizan eran correctas, y poder añadir otras cadenas que devolvieran resultados interesantes.

Dentro de los casos de usos analizados se puede ver que excepto el tema del análisis de imágenes que parece que anda más en horas bajas, debido a su dificultad, costes de computación...etc., el resto son temáticas con muchas aplicaciones y que actualmente se están desarrollando mucha tecnología a su alrededor. Es importante tener en cuenta las últimas aplicaciones y casos de uso realizados en estos temas a la hora de realizar una aplicación en estos ámbitos y que aporte nueva funcionalidad.

3. Propuesta

A continuación, se mostrarán los requisitos, el análisis y el diseño de la solución implementada para este proyecto a fin de resolver los problemas existentes remarcados a lo largo de los puntos anteriores.

3.1. Requisitos generales funcionales

Dentro de los requisitos funcionales se deberán tener en cuenta las siguientes funciones:

- El sistema debe ser capaz de clasificar imágenes y texto según la taxonomía definida.
- El sistema debe procesar los eventos hasta dejarlos listos para servir.
- El sistema debe servir los eventos procesados mediante un bot de Telegram.
- El sistema debe ser capaz de analizar las conversaciones grupales de Telegram y asociar su temática con la taxonomía definida.
- El sistema deberá recibir valoraciones de los usuarios ante las recomendaciones.

3.2. Requisitos generales no funcionales

Dentro de los requisitos no funcionales se deberán tener en cuenta las siguientes funciones:

- El sistema estará dividido en máquinas virtuales según la funcionalidad que desempeñen.
- Se debe proporcionar un servicio rápido entre las máquinas.
- La transferencia de información entre el backend y los productos del frontend se realizará mediante un servicio de Api Rest.
- Se deberá proporcionar seguridad tanto de acceso al sistema, así como al despliegue.

3.3. Análisis y Diseño

En esta sección se va a proceder con el análisis y diseño de las distintas partes que componen el trabajo realizado. En él se van a analizar las cuatro funcionalidades básicas, definidas anteriormente en la propuesta (Recolección, Procesamiento, Predicción, Servicio). Se incluirá un análisis de requisitos en cada una de ellas teniendo en cuenta el estudio de casos de uso habitual realizado anteriormente para cada uno de los ámbitos en los que se divide el proyecto.

3.3.1. Recolección

Análisis de las fuentes de imágenes

La fuente elegida para la extracción de imágenes que se usarán en el entrenamiento del predictor va a ser [Flickr](#), aun así, se va a realizar la implementación para las fuentes [Imagenet](#) e [Instagram](#), para realizar pruebas y experimentos. En principio no se va a realizar mezcla de imágenes de distintas fuentes, ya que los tipos de foto encontrados en cada aplicación, son diferentes entre sí y podría ensuciar los experimentos. La selección de estas tres fuentes es debida a su calidad de imágenes, riqueza de etiquetado y facilidad de extracción.

La extracción de información de estas fuentes se realizará en los tres casos mediante el Api de sitio web, que en cada caso tendrá un proceso de autenticación diferente. Excluyendo la parte específica de autenticación y el mapeo de información de cada fuente, el resto de acciones a realizar serán idénticas, lo cual permitirá abstraer esa funcionalidad y que sea común a las tres fuentes. La periodicidad con la que se obtendrán imágenes será de tres veces por semana como máximo, dejando tiempo a que los usuarios suban nuevas imágenes y optimizando el proceso de descarga.



Ilustración 14 Análisis de Extracción de imágenes en fuentes de imágenes

Análisis de fuentes de eventos

Hay infinidad de páginas de eventos sociales que se pueden considerar fuentes. Por lo general, cada una de estas fuentes está hecha con tipos de datos diferentes, una tecnología diferente, un diseño diferente y en ningún caso siguen ningún formato establecido (ni a veces en la misma página). Aun así, es posible establecer un procedimiento mediante el cual automatizar la extracción de eventos. Existen cuatro formas habituales de servir eventos, desde una página **web sencilla**, desde una **web paginada**, desde una **web con JavaScript** y desde un **Api Rest**. En todos los casos la extracción consistirá en un *scrapeo* del sitio web al nivel de profundidad que sea necesario, por lo tanto, hará falta una solución recursiva. Es habitual que la información de las páginas se sirva rellenando plantillas idénticas dentro de una página, siguiendo patrones estructurales definidos por el creador de la misma. Por ello, definiendo la lógica del scrapeo, la estructura de la información en las páginas, configuración adicional de autenticación si fuese necesaria y el mapeo de la información a la estructura definida de salida, nos permitiría obtener una solución que podría coger cualquier fuente que se quisiera. El resto de funcionalidad será común a todas las fuentes, pudiendo crear una librería de uso común.

Es importante recalcar que, a la hora de extraer información de forma masiva de páginas web, se está sometiendo al sitio web a una sobrecarga de rendimiento de peticiones que podría hacer caer el servicio en caso de no soportarlo o en defensa de este hecho, que se produjera un baneo.

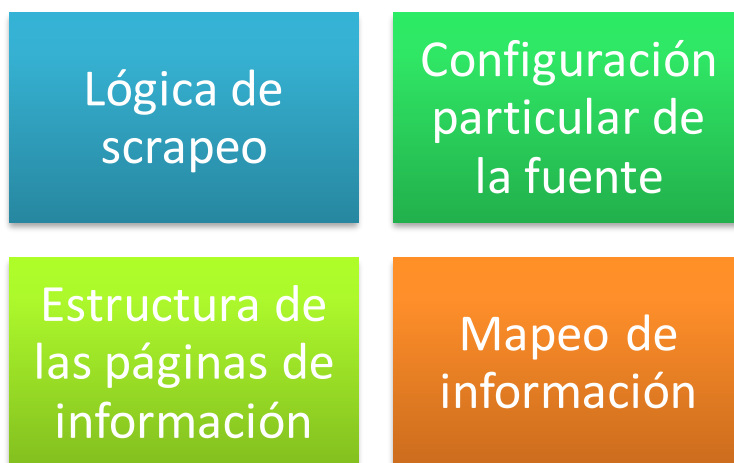


Ilustración 15 Elementos básicos a definir en un scraper de una fuente de eventos sociales

Para la descarga periódica de eventos se establecerá un tiempo de repetición para cada fuente, mediante el cual pasado ese tiempo se scrapeará de nuevo la página. Los eventos una vez procesados irán a parar un sistema de almacenamiento temporal, levantando una señal en el sistema para programar su procesado. En el caso de existir el evento ya en el sistema (coincidencia de código único) se realizará una actualización sobre este y se notificará al sistema central que hay que reprocesarlo.

Colores

Dentro de los trabajos revisados en el estado del arte y relacionados con el análisis cromático, se observó la constante de usar el modelo HSV de colores. Este modelo representa el matiz, la saturación, y el valor para definir un color. El matiz es un valor en grados que va de 0 a 360°, donde en comparación con el modelo RGB se dividen los 3 colores dando lugar a un total de 120° por color. La saturación representa el porcentaje de brillo negro-blanco de un color lo que hará que parezca más o menos decolorado. Por último, el valor representa cuanto blanco-negro tiene la imagen. Es posible convertir de esta modalidad de cromática a otra, pero en este trabajo se hará uso de esta siguiendo la línea marcada por los trabajos anteriores.

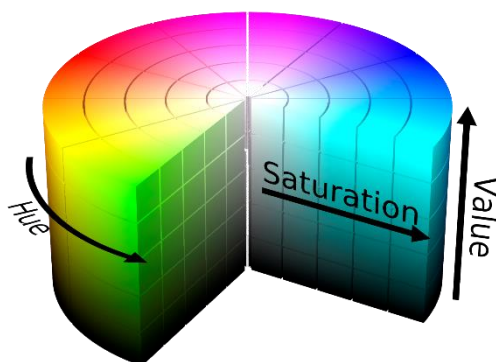


Ilustración 16 Representación modelo de colores HSV

La cantidad de colores de una imagen irá en proporción del tamaño y del número de píxeles que está tenga. En el caso de este trabajo se va a hacer uso la mayoría de las veces de imágenes reducidas para su uso en internet, lo cual disminuye el número de píxeles originales, aun así, por motivos de rendimiento es necesario realizar una simplificación. Dicha simplificación se realizará mediante un clustering de colores, extrayendo únicamente los colores más relevantes de cada imagen. Para ello se hará uso del algoritmo *Kmeans*, el cual se encargará de agrupar colores y devolviendo los más relevantes y el porcentaje de píxeles que cayó en dicho clúster. Es importante ajustar el tamaño del clúster, para no realizar ni aproximaciones muy pequeñas ni muy grandes. Una vez extraído el clúster de todas las imágenes de una categoría se realizará un segundo proceso de clustering a fin de sacar los clústeres a fines a la categoría. Igual de importante que era la elección de un número correcto de clúster a formar en el primero proceso de clustering, lo es para este segundo proceso.

Conversaciones de Telegram

El bot almacenará las conversaciones que ocurren en los chats grupales para su posterior análisis en busca de intereses de los participantes relacionados con los eventos sociales. Se van a almacenar tanto los mensajes de texto como las imágenes adjuntas en la conversación. En el caso de la imagen se almacenará ya analizados los colores y con una predicción ya asignada sobre su pertenencia a una categoría de eventos social. En cuanto al tipo de chat, solo se guardarán los chats grupales, suponiendo que un usuario individual no vaya a tener una conversación de relevancia con el bot en su versión actual. No se considera un límite establecido en cuanto al almacenamiento de chats de momento debido a su fase inicial de implantación.

Feedback de recomendación

Para conseguir que evolucione el sistema de recomendación, es necesario obtener feedback que permita balancear las recomendaciones efectuadas en el sistema. El sistema de feedback consistirá en la extracción de las categorías de eventos pertenecientes a la recomendación realizada y su valoración otorgada. Para ello el usuario introducirá su valoración del 0 al 5 teniendo las siguientes puntuaciones:

Valoración	Puntuación
0	-3
1	-2
2	-1
3	1
4	2
5	3

Ilustración 17 Tabla de valores de puntuación de feedback

Una vez obtenida la valoración se multiplicará la puntuación con el número de eventos de esa categoría y de esa forma se obtendrá la valoración del usuario sobre esta categoría. Posteriormente se enviará para almacenarla en la base de datos de feedback.



Ilustración 18 Cálculo del feedback de recomendación

Análisis de atributos y restricciones

Se van a definir una serie de atributos que definirán la estructura y deberán estar presentes tanto en el ámbito de las imágenes como en el de los eventos sociales, a la hora de tomar estos como válidos. En el caso de que falte algún atributo en alguna de las estructuras, se tomará como inválido y se descartará. En el caso de las imágenes:

- **Escala de grises:** Se desestimará toda imagen en la cual se detecten patrones que indiquen que fue descargada en escala de grises.
- **Texto o palabras no válidas:** Se realizará un análisis para detectar palabras en las imágenes y se comparará con una lista de palabras prohibidas para excluir imágenes inapropiadas.
- **Selfies:** Se comprobará que la fotografía realizada no es un selfie, descartándola en caso positivo.
- **Código Único:** A fin de no repetir el costoso procesamiento de las imágenes, se les asignará un código único que permita distinguir unas con otras.
- **Colores:** Tendrá necesariamente una clusterización de N colores, con su proporción correspondiente de colores contenidos en dicho clúster.
- **Rostros:** Contará el número de rostros que aparecen en la imagen teniendo en cuenta que la imagen no es un selfie.

En el caso de los eventos sociales, más que restricciones por características de atributos se excluirá un evento por la falta de alguno. Los atributos necesarios para los eventos sociales son:

- **Fuente:** Nombre de la fuente de la que se extrajo la información del evento social.
- **Código Único:** Se creará un código único que lo diferencia del resto y permita actualizarlo. A la hora de elegir la información que forme el hash de este campo, es

importante tener en cuenta la lógica de la fuente y sus actualizaciones para elegir bien los campos.

- **Nombre y descripción:** Es necesario que el evento traiga nombre y descripción para poder realizar la predicción de categoría por texto.
- **Imagen:** Es necesario que el evento traiga la url de la imagen y esta siga online en el momento del procesado para poder predecir haciendo uso del análisis cromático de la imagen.
- **Fecha de inicio-fin:** Deberá tener al menos fecha de inicio, en el caso de no tener fecha de fin se supondrá que el evento finaliza al final de ese mismo día a las 23:59.
- **Categoría:** En el caso de encontrarnos con una fuente cuyos eventos sean de una categoría específica se puede especificar directamente la categoría, sin necesidad de pasar por el proceso de predicción.
- **Lugar:** Es imprescindible para poder localizar el evento que se encuentre información sobre su localización, ya sea por coordenadas, por nombre de región o por nombre de lugar y dirección postal.

En el caso de los eventos se podrán almacenar cualquier campo adicional a los presentes proveniente de la fuente, por si en futuras incorporaciones de información tiene sentido, tenerlos ya descargados y sea más fácil su incorporación.

Requisitos funcionales

Los requisitos funcionales definidos para el caso de la funcionalidad de Recolección, son los siguientes:

- El sistema realizará descargas masivas de eventos e imágenes.
- El sistema debe ser capaz de autoabastecerse periódicamente de eventos.
- El usuario podrá programar la recolección de eventos.
- El usuario podrá añadir más fuentes con cierta facilidad.
- El sistema extraerá y validará la información obtenida haciendo uso de las estructuras de datos pensada para cada caso.
- El sistema almacenará tanto conversaciones como valoraciones realizadas a través del bot de Telegram.

Requisitos no funcionales

Los requisitos no funcionales definidos para el caso de la funcionalidad de Recolección, son los siguientes:

- Se diferenciará completamente el flujo de información de eventos e imágenes.
- El sistema debe ser distribuido y escalable en todos sus componentes.
- El sistema deberá incorporar posibilidad de proporcionar anonimidad en el scrapping (proxy).
- El sistema debe ser distribuido y escalable en todos sus componentes.
- El sistema se podrá desplegar de forma automática.
- Se realizarán abstracciones en el desarrollo para facilitar el proceso de añadir más fuentes.

3.3.2. Procesamiento

Es necesario definir procedimientos a fin de mejorar haciendo lo más óptimo posible el procesamiento de las grandes cantidades de información a procesar en este proyecto.

Análisis procesado de imágenes

El procesamiento de imágenes se llevará a cabo en tres fases, una primera de descarga masiva, revisión manual y otra de procesado por clases.

- **Descarga masiva:** Para realizar la descarga masiva, se definirán una serie de palabras de búsqueda para cada clase de la que se quieran extraer imágenes. A continuación, se descargarán las imágenes, agrupadas por clase.
- **Revisión manual:** Se debe realizar una revisión manual a fin de eliminar imágenes que no tengan que ver con la temática o sean inapropiadas y se salten los filtros. En un futuro cuando el predictor esté bien entrenado, llegando a grandes niveles de precisión, se podrá usar para realizar esta fase de revisión de forma automatizada.
- **Procesamiento de clases:** El procesado de las imágenes de una clase se va a realizar en dos fases, primero se van a procesar todas las imágenes asignadas a una imagen extrayendo sus colores más representativos, y a continuación se va a realizar la extracción de los colores más representativos de la clase introduciendo todos los colores más representativos de las imágenes de la clase.

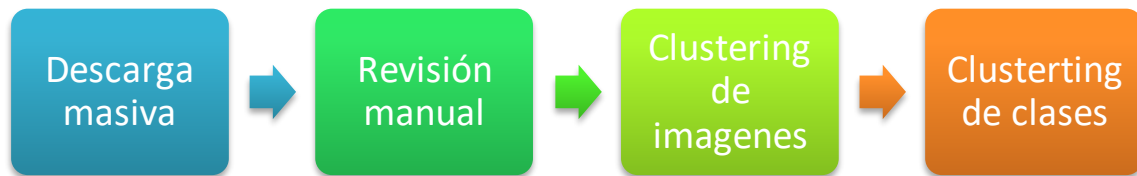


Ilustración 19 Procesamiento de imágenes

Algoritmos de procesamientos de imágenes

Básicamente se va a hacer uso de tres algoritmos de machine learning para realizar predicciones sobre las imágenes. Los algoritmos seleccionados han sido elegidos teniendo en cuenta opiniones de terceros y resultados observados en los distintos trabajos del estado del arte. Los algoritmos de procesamiento son:

- **Kmeans**

Algoritmo de agrupamiento que tiene como objetivo la partición de un conjunto de n observaciones en k grupos en el que cada observación pertenece al grupo más cercano a la media.

$$arg_s \min \sum_{i=1}^k \sum_{x_j \in S_i} \|x_j - \mu_i\|^2$$

En el sistema, es el Algoritmo encargado de realizar una reducción de colores, simplificando el número de colores a manejar tanto en el caso de los colores de imagen como en el de los colores de una clase. Este proceso como ya se dijo antes es necesario por temas de computación ya que no se disponen de medios de alto rendimiento. La reducción del problema aporta simplicidad, a costa de introducir error al sistema. El error introducido depende tanto de los colores de la imagen a reducir, como del número de clúster elegido para su reducción.

- **SVM**

Algoritmo relacionado con problemas de clasificación y regresión, en los que dado un conjunto de entrenamiento es posible etiquetar clases y entrenar una SVM para construir un modelo que prediga la clase de una nueva muestra. Este modelo es una representación de los puntos de la muestra en el espacio, separando las clases en espacios lo más amplios posibles mediante un hiperplano de separación, definido como el vector entre los puntos de las clases más cercanos, al que se llama vector soporte. Por ello a este algoritmo se le conoce como máquina de soporte vectorial.

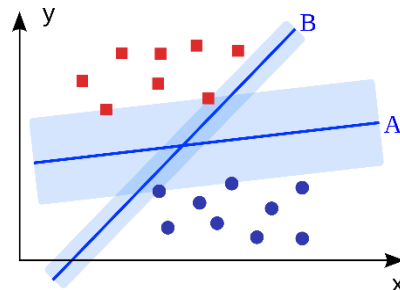


Ilustración 20 Ejemplo visual de SVM

En el sistema se hace uso de este algoritmo para la predicción mediante regresión de un color para obtener su categoría más probable. El entrenamiento se realizará previamente haciendo uso de los colores más representativos de cada categoría obtenidos de su clusterización.

- **RBM**

El algoritmo Restricted Boltzmann Machine (RBM) es una forma particular de logaritmo lineal de Markov Campo Aleatorio (MRF), es decir, para la cual su función de energía es lineal en sus parámetros libres. Para que sea lo suficientemente potente como para representar distribuciones complicadas (es decir, para pasar de la configuración paramétrica limitada a una no paramétrica), hay que tener en cuenta que algunas de las variables no se tienen en cuenta (variables ocultas). Aumentando el número de variables ocultas, se aumentará la capacidad de modelado de la máquina de Boltzmann (BM).

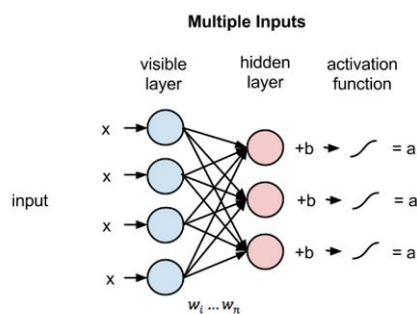


Ilustración 21 Ejemplo visual de RBM

En el sistema se hace uso de este algoritmo como alternativa a SVM. Se han realizado distintos experimentos y tampoco se consiguió determinar que algoritmo devolvía mejores resultados, por lo que queda a gusto del usuario elegir con que algoritmo se realice la predicción. Según la literatura existente [40] a los expertos tampoco les queda claro que algoritmo es mejor a la decantarse por uno en vez de por el otro. Por último indicar simplemente que, en el caso de las redes neuronales aplicadas en el campo de las imágenes, es fácil caer en el sobre-entrenamiento por lo que se deberá realizar un entrenamiento más cuidado.

Análisis procesado de eventos

Para el procesamiento de eventos se va a realizar un pipeline de procesos encadenados con estados de parada en el caso de que un evento no supere un procedimiento. El procesamiento se iniciará una vez recibido el mensaje del scraper, avisando que un nuevo evento ha sido almacenado en el base de datos de descarga. Los procesos a realizar en orden de aparición son los siguientes:

- **Recuperación del evento descargado:** El primer proceso consiste en la recuperación de la información descargada del almacenamiento temporal en el que los deja el scraper.
- **Localización del evento:** Se comprobará que está bien geolocalizado, realizando consultas espaciales contra la base de datos o google Places en el caso de no estar completa o correcta la información.
- **Traducción a múltiples idiomas:** Se traducirá el evento a varios idiomas a fin de poder dar servicio independientemente del idioma del usuario. Se traducirán los nombres y las descripciones con posibilidad de forzar el que no se traduzca el nombre en ciertos casos.
- **Creación de sesiones:** En el caso de que en el evento traiga información relativa a las sesiones, se realizará la generación de las mismas. Esto quiere decir que, si viene la frase “todos los lunes y martes” y sabiendo la fecha de inicio y fin, se podrán crear todos los lunes y martes de entre las fechas establecidas las sesiones correspondientes.
- **Extracción de categoría:** Se concatenarán el nombre del evento y su descripción, pasándolo por el predictor de categoría a fin de clasificarlo.
- **Control de calidad:** Finalmente antes de terminar el pipeline, se realizará una revisión a fin de encontrar eventos similares ya publicados. La manera de comprobar que el evento fue publicado ya, será analizando si hay más de un evento en el mismo espacio-tiempo y además hay coincidencia en sus descripciones y categorías. Con esto se evitará caer en la duplicación de eventos de distintas fuentes.

En el caso de que un evento se quede estancado en una fase del procesamiento, se requerirá de una revisión manual a falta de una solución mejor.



Ilustración 22 Pipeline de procesamiento de eventos

Análisis feedback del usuario

Para poder realizar en un futuro recomendaciones de mayor precisión respecto al gusto de los usuarios, se analizarán las valoraciones dadas por los usuarios. En dichas valoraciones se seguirá el siguiente formato:



Ilustración 23 Estructura de feedback

A la hora de procesar el feedback recibido el sistema juntará todas las valoraciones realizadas teniendo en cuenta si son de un usuario concreto o de un chat concreto. A partir de este agregado de feedbacks se balancearán las cargas de las categorías, estableciendo así los gustos del usuario. A partir de estos gustos se realizará un filtrando más consecuente a la hora de realizar recomendaciones de eventos.

Análisis de las conversaciones

En el sistema de recomendaciones diseñado también se hace uso de la temática de la conversación. Para ello se irá almacenando toda la conversación de un chat en la base de datos y a la hora de realizar una recomendación se recuperará para analizarla. En el análisis de estas conversaciones, se juntarán todos los textos y se usará el predictor de texto para extraer la categoría asociada a las conversaciones pasadas. Una vez extraída se comprobará si existen valoraciones realizadas a fin de establecer si se hablaba negativa o positivamente en la conversación, por ejemplo, si la categoría extraída del texto es fútbol, pero las valoraciones de fútbol en ese chat son negativas, reforzará el sentimiento negativo de esa valoración. En el caso de ser positivo lo reforzará positivamente y en el caso de no existir se añadiría positivamente por defecto. Esta estrategia a falta de pruebas y de un estudio exhaustivo para acertada, aunque se incluirá más adelante en trabajos futuros otras estrategias a tener en cuenta y probar.



Ilustración 24 Proceso de análisis de conversaciones

Requisitos funcionales

Los requisitos funcionales definidos para el caso de la funcionalidad de Procesamiento, son los siguientes:

- El procesado de eventos estará dividido en varios subprocesos, teniendo que pasar por todos en el orden establecido para que un evento sea aceptado como válido.
- Cada subproceso recibirá un evento en un estado y lo devolverá en otro.
- El usuario podrá colocar eventos en la cadena de subprocesos saltando subprocesos.
- Los procesamientos realizados deben ser reproducibles.
- El procesado de imágenes dará como resultado un fichero pickle.

Requisitos no funcionales

Los requisitos no funcionales definidos para el caso de la funcionalidad de Recolección, son los siguientes:

- El sistema tendrá persistencia en todo momento respecto del estado del evento.
- El procesado de imágenes almacenará el resultado del clustering final de cada imagen.
- El sistema montará un sistema de colas para controlar la entrada de eventos en el sistema y evitar su saturación.

3.3.3. Predicción

La predicción de categorías va a ser un elemento muy importante y central dentro del sistema, ya que se hará uso de él en distintos puntos del mismo. La importancia y calidad otorgadas a este módulo es máxima, ya que será proporcional a la calidad final del sistema global.

Procedimiento de predicción

Como ya se dijo anteriormente se van a realizar predicciones sobre colores y eventos para asignarlos a categorías. En el proceso se obtendrá no solo la clase predicha, sino su porcentaje de

confianza respecto al resto, lo cual no permite saber la calidad de la predicción y da la posibilidad de establecer límites de calidad.

El sistema de predicciones deberá seguir un crecimiento incremental de precisión. Para ello a medida que se obtengan más elementos en el sistema se realizarán entrenamientos periódicos con sus test correspondientes, para estimar la mejoría del sistema. En caso de mejora se actualizará con el nuevo. Este procedimiento de actualización debe ser sencillo y permitir volver a versiones anteriores.

Análisis de categorías

A partir de las categorías observadas de las referencias analizadas en el estado del arte, se ha creado una taxonomía propia con dos niveles categóricos, que engloben todos los eventos que puedan venir de internet. El primer nivel engloba temáticas genéricas y el segundo más específicas dentro del primero. La taxonomía desarrollada es la siguiente:

Categoría Padre	Categoría Hija
sports	american_football
	athletics
	australian_football
	baseball
	basketball
	boxing
	cricket
	cycling
	darts
	extreme_sports
	football
	formula_1
	golf
	handball
	hockey
	horse_racing
	motorsports
	other_sports
	polo
	rugby
	running
	sailing
	street_sports
	sumo
	tennis
	water_sports
winter_sports	
wrestling	
art_and_culture	cinema
	contemporary_art
	exhibition
	literature
	painting
	photography
holiday	Sculpture
	buddhism
	christian
	hebrew

	hinduism
	local holiday
	local observance
	muslim
	national holiday
	observance
	orthodox
	other_days
	season
	typical_non_working_days
trade_fairs_and_conferences	automotive
	business_and_networking
	computers_and_technology
	fashion
	medicine_and_health
	other_conferences
	other_trade_fairs
	tourism
music	classical_music
	clubbing_and_party
	concert
	festivals
performing_arts	ballet
	comedy
	dancing
	flamenco
	magic
	musical
	opera
	theater
other_interesting_events	bullfighting
	christmas_market
	courses_and_workshops
	food_and_drinks
	gastronomy
	kids_and_family
	local_celebrations
	other_events
	religious
	street_markets

Tabla 4 Taxonomía de eventos sociales de internet

Como se puede ver en cada temática padre se encuentran las categorías más representativas de eventos que se pueden encontrar en internet dentro del género. En casi todas las categorías se ha dejado una categoría *other_XXX*, en el que se incluirían aquellas subtemáticas raras que se han quedado sin representación debido a su rareza en internet. En el caso de los eventos que hay ambigüedad entre varias categorías o no quede claro a cuál pertenece, se ha decidido crear una categoría padre llamada *other_interesting_events* que los englobaría, como por ejemplo los eventos de temática taurina.

Extracción de categorías

Para la extracción de categorías se habilitará a modo de servicio Api Rest con cada uno de los predictores. En el caso de los eventos se esperará la concatenación del nombre y la descripción,

en formato texto, siendo estos los campos representativos del evento. En el caso de las imágenes se esperará un path o url, que contenga la imagen. Posteriormente se descargará dicha imagen, extraerán sus colores y se predecirá su categoría. En ambos predictores el resultado será una categoría de la taxonomía mostrada anteriormente o nada en el caso de que no quedará claro o hubiese algún fallo en el proceso.

Entrenamiento

Es necesario entrenar el sistema de predicción para su funcionamiento. En ambos casos, tanto en imágenes como en texto, el sistema deberá incorporar más información para el entrenamiento hasta encontrar el punto de sobreentrenamiento. En ese momento en el caso de querer mejorar la predicción será necesario realizar tareas de filtrado o procesamiento que mejoren el entrenamiento. A continuación, se explica el procedimiento de entrenamiento para las dos fuentes de datos existentes:

Texto

Para el entrenamiento de texto se va a hacer uso de eventos existentes en el sistema cuyas categorías hayan sido ya asignadas y hayan llegado a publicarse (pueden estar expirados). De estos eventos se van a usar los campos del nombre, la descripción, las keywords y su categoría. Es importante recalcar que en la temática de predicciones sobre texto es recomendable contar tanto de textos largos como con una gran cantidad de ellos. En el caso de los eventos, existe la posibilidad de que algunos vengan con una escasa descripción (por la naturaleza de la fuente de la que se extrajeron) por lo que puede ser complicada la predicción. Para el entrenamiento se van a repartir los eventos seleccionados en 2 sets (entrenamiento y test) de forma aleatoria y se van a probar distintos algoritmos a fin de que compitan y finalmente se acabe eligiendo el mejor de ellos para usar en el sistema.

En el proceso de entrenamiento, primero se limpiarán los textos de palabras sin relevancia (determinantes, preposiciones...) dejando las palabras de utilidad para la clasificar. A continuación, se van a ponderar las palabras asignándoles una relevancia según a la categoría más probable a la que hagan referencia y posteriormente se establecerá una clase a la que pertenece ese evento. Una vez realizado este proceso, se observarán los resultados de precisión y se decidirá poner en uso el pickle correspondiente al algoritmo con mejor precisión para ese servicio de clasificación.



Ilustración 25 Entrenamiento de texto

Imágenes

Para el entrenamiento de las imágenes, como ya se dijo antes, se van a realizar descargas de imágenes sociales de fuentes de internet. Una vez descargadas se realizará un filtrado manual para seleccionar imágenes representativas de la temática correspondiente (entiéndase como validación no como sesgo). Dentro de la temática del análisis de imágenes no solo es necesario muchas imágenes, sino también lidiar con problemas típicos de estas, para ello se van a realizar una serie de pre-filtrados o acciones configurables a realizar a la hora de aceptar una imagen para entrenar. Estos pre-filtrados son los siguientes:

- **Redimensión:** Se establecerá una serie de parámetros a fin de redimensionar todas las imágenes a una dimensión común entre todas. De esta manera las proporciones de todas serán idénticas y se podría hacer uso de esta acción con fines de optimización.

- **Blanco y negro:** Se filtrarán imágenes en blanco y negro, dejándolas de lado en la selección debido a su falta de colorido y utilidad por lo tanto en este proyecto.
- **Reconocimiento de palabras:** A fin de evitar introducir en el sistema imágenes pornográficas sobretodo, se definirán una serie de palabras a modo de lista negra, que en el caso de reconocer dichas palabras en la imagen retirarán dicha imagen del proceso de selección.

Como se explicó anteriormente en 3.3.2, la carga y procesamiento de imágenes es independiente del entrenamiento, pudiendo elegir a la hora de realizar la selección de imágenes para entrenar, el tirar de la base de datos con imágenes ya procesadas o hacer una selección propia que pasará primeramente por procesado y luego por entrenamiento. Una vez obtenida la selección se van a formar 3 sets en este caso (entrenamiento, test de entrenamiento y test). La posibilidad de formar estos 3 sets dependerá de si se quiere controlar el test final, pudiendo variar de esta manera la formación de dichos conjuntos. A continuación, se muestra gráficamente las dos configuraciones posibles:

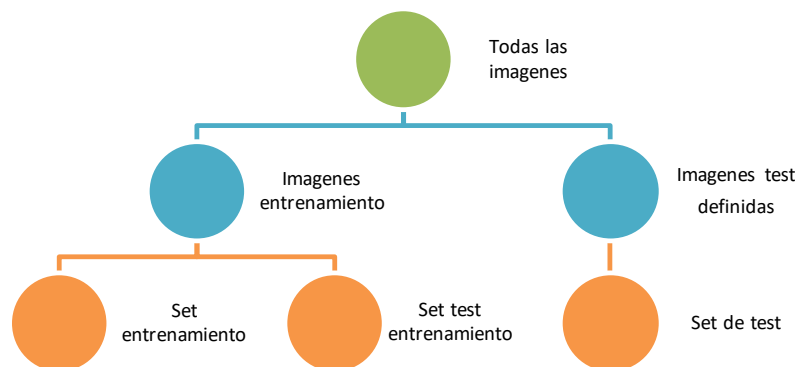


Ilustración 26 Formación de set usando test definido

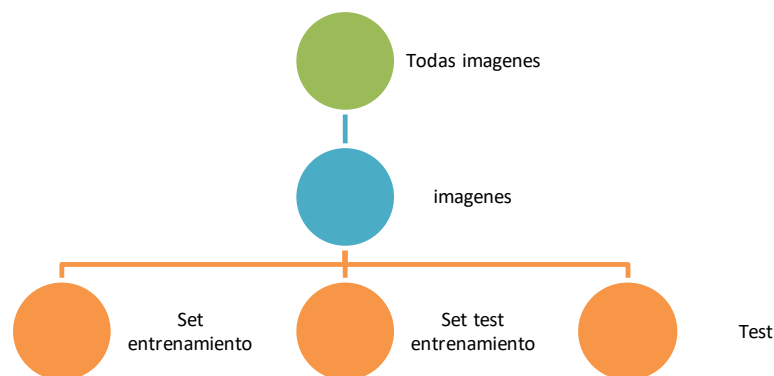


Ilustración 27 Formación de sets sin test definido

Una vez establecidos los sets se procederá a la ejecución de los algoritmos seleccionados (es posible seleccionar más de uno) de entre los siguientes: svm, rbm o svm_cascade que es una aproximación realizada para este proyecto. Las dos primeras son simplemente aplicaciones se los

algoritmos ya implementados y a continuación se procederá a comentar el algoritmo desarrollado con el nombre svm_cascade:

Svm cascade

Este algoritmo es una aplicación del svm tradicional en el cual se realiza una combinación de todas las posibilidades de clases posibles para predecir a modo de cascada. Para ello se sigue la siguiente algoritmia en cada imagen:

- **Fase de barajar de clases:** En esta fase se repartirán las clases existentes en el entrenamiento en grupos de máximo 3 clases, haciendo una especie de sorteo clasificatorio.
- **Fase de entrenamiento:** En esta fase se realizará una combinatoria entre todas las clases del grupo y se entrenarán las clases obteniendo un porcentaje de precisión para cada clase, respecto de la combinatoria entrenada. Con esta estrategia se pretende poder elegir de forma automática el mejor predictor para cada clase posible. Se guardará el entrenamiento realizado por si en un futuro fuese necesario repetir ese grupo evitar tener que volver a entrenar.
- **Fase de predicción:** En esta fase se intentará predecir la categoría de la imagen objetivo usando el entrenamiento realizado para cada uno de los grupos.
- **Fase de verificación:** En esta fase se verificará que la clase predicha coincida con uno de las clases integrantes del grupo, añadiéndose esa clase a las clases a promocionar en el siguiente ciclo de repetición del proceso. En el caso de que tras verificar todos los grupos se tenga una sola clase, se considerará esa clase como la clase finalmente predicha y terminará el proceso.

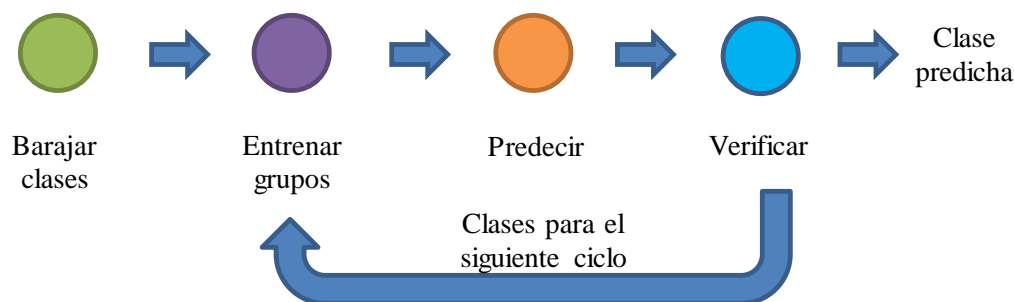


Ilustración 28 Esquema de funcionamiento de algoritmo SVM cascade

Este algoritmo no es recomendable finalmente para este proyecto ya que al aumentar el número de clases se va haciendo cada vez más lento e ineficiente, aun así, es una propuesta interesante para optimizar y usar en un futuro ya que con este algoritmo se obtuvieron los mejores resultados al hacer uso de pocas clases.

Requisitos funcionales

Los requisitos funcionales definidos para el caso de la funcionalidad de Predicción, son los siguientes:

- Se debe definir una taxonomía válida de clasificación para eventos de internet.
- El sistema agrupará todos los procesos predictivos en un servicio Api Rest.
- Las predicciones realizadas deberán ser reproducibles.
- El entrenamiento deberá ser independiente del servicio.

Requisitos no funcionales

Los requisitos no funcionales definidos para el caso de la funcionalidad de Predicción, son los siguientes:

- Cada proceso predictivo tendrá su propio fichero de entrenamiento.
- Los ficheros de entrenamiento deben ser fácilmente gestionables.
- El sistema debe poder desplegarse de forma automática.

3.3.4. Servicio

Servicio de Api

El servicio de eventos se realizará mediante un Api Rest. Este Api servirá tres tipos de contenidos inicialmente, los eventos publicados, las regiones publicadas y las taxonomías. Como medida para incrementar la velocidad de servicio de eventos en el sistema, el Api hará uso de un indexador de contenidos para facilitar esta labor. En ámbito de seguridad del Api, el sistema se encargará de proporcionar a los usuarios registrados una clave en forma de *Token* para su uso en peticiones al Api.

Análisis bot Telegram

Uno de los puntos fuertes de esta aplicación con soporte para redes sociales, es la posibilidad de incorporar entidades autónomas e inteligentes de terceros (bots) que realizan alguna función. El framework desarrollado viene preparado para el establecimiento de comandos y la reacción asíncrona ante introducción del nombre del bot tras una '@' llamado internamente como 'inlinequery'. Por defecto, el sistema de Telegram gestiona internamente los mensajes recibidos en un bot sirviéndolos tras un endpoint al cual es posible acceder tras una simple autenticación. También en el caso de querer gestionar y procesar personalmente los mensajes, permite la opción de establecer un webhook. En el caso de elegir esta opción hay que saber que Telegram redirigirá los mensajes entrantes al endpoint indicado y en caso de fallar o no estar disponible se perderá. Para establecer el webhook es necesario un endpoint estático, así como habilitar el protocolo *https* y compartir el certificado con Telegram (Ver anexo 6 [Creación de un bot de Telegram](#)).

El procedimiento que se va a realizar tras recibir un mensaje difiere dependiendo del tipo de mensaje, tipo de chat y el contenido del mismo. A continuación, se detallará el procedimiento a realizar en cada caso.

Tipos de chat

La diferencia entre el tipo de chat nos hará plantearnos cuestiones entorno a la utilidad de la información expuesta en este tipo de chats, así como la necesidad de su análisis y almacenamiento.

- **Individuales**: Un chat individual es aquel en el cual se va a encontrar únicamente el usuario y el bot en la conversación. Este tipo de chats se usan simplemente para obtener un servicio del bot y en el cual la conversación existente se realiza mediante intercambios de comandos y respuesta con el bot. Para este tipo de chats el bot no guardará un registro de los mensajes del usuario, ya que se presupone que no existirá una conversación rica en contenido.
- **Grupales**: Se definen como chats grupales aquellos en los que haya más de dos personas. En este caso la conversación surgida si es rica en contenido pudiendo revelar datos sobre los usuarios. Este tipo de conversaciones serán almacenadas filtrando los comandos de bot existentes en ella. Una vez almacenada se hará uso de esta información, cada vez que

se realice una consulta de comando, con el fin de obtener recomendaciones de eventos que se ajusten más a la temática del grupo.

Tipos de mensaje

El tipo de mensaje va a definir la acción a realizar dentro del bot. Los tipos de mensajes aceptados son los siguientes:

- **Mensaje de chat**: Contendrá información sobre la conversación sin ningún trabajo para el bot más allá de su almacenamiento.
- **Comando**: Entre los mensajes de chat el usuario puede enviar comandos que serán procesados por el bot. La sentencia de un comando sigue la secuencia habitual dentro del ámbito informático formada por los siguientes elementos:



Ilustración 29 Estructura de comando del bot de Telegram

La sentencia comenzará con un “/” que indicará que el nombre a continuación es un comando a interpretar con el bot. Cuando se añade el carácter ‘/’ la aplicación de Telegram por defecto te muestra los comandos disponibles para los bots incluidos en la conversación. La gestión de argumentos ya depende íntegramente del desarrollo del comando, aunque deberá saber distinguir y asignar los argumentos introducidos al tipo de dato correspondiente. En el bot desarrollado en este proyecto se van a contemplar los siguientes comandos:

Comando	Argumentos	Descripción
/help	No tiene	Devuelve información sobre el bot y los comandos disponibles en él.
/events	<ul style="list-style-type: none"> • Nombre de la región: Nombre de la región de la que se quiera obtener eventos. • Coordenadas: Tendrá el formato latitud, longitud. • Fechas: Tendrá el formato YYYY-mm-dd. En el caso de introducir dos fechas se considerará la recomendación de eventos entre esas fechas. 	Devuelve una lista de eventos para los argumentos introducidos.
/recommend	<ul style="list-style-type: none"> • Nombre de la región: Nombre de la región de la que se quiera obtener eventos. • Coordenadas: Tendrá el formato latitud, longitud. 	Devuelve una recomendación basada en las evaluaciones y temática de la conversación.

	<ul style="list-style-type: none"> • Fechas: Tendrá el formato YYYY-mm-dd. En el caso de introducir dos fechas se considerará la recomendación de eventos entre esas fechas. 	
/myrecomendationinfo	No tiene.	Devuelve las categorías de las cuales se tienen valoraciones, ordenadas por gusto en orden descendente.
/ <i><id del evento></i>	No tiene.	Devuelve una información más detallada sobre el evento, al pulsar sobre el enlace que viene en la lista de eventos recomendados.

- **Inline Query:** Este tipo de mensaje se recibirán en el bot cuando el usuario llame al bot utilizando la sentencia de inline query. Esta sentencia consistirá en introducir la siguiente estructura de consulta en la entrada de mensajes:



Ilustración 30 Estructura de Inline Query en el bot de telegram

Una vez introducida el bot devolverá distintas opciones a modo de inline keyboard que contendrá distintas consultas a realizar variando el tiempo de la misma. Este método a diferencia del comando, su sentencia no deja huella en el chat al introducirla (se introducirá un texto descriptivo previo a los resultados devueltos) y puede ser utilizada, aunque el bot no esté en la conversación. Resaltar que para realizar una predicción analizando los mensajes del grupo es necesario incluir al bot dentro de la conversación.

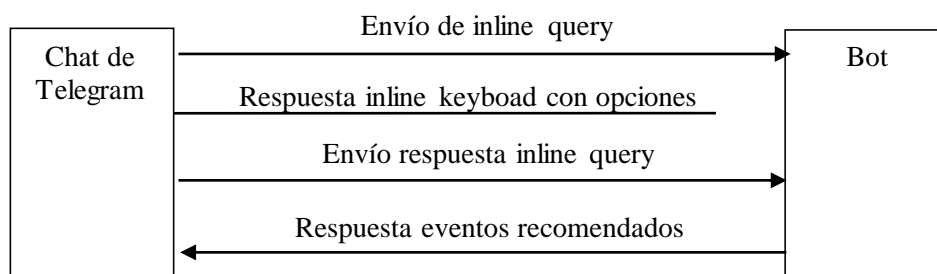


Ilustración 31 Diagrama de uso de Inline Query

- **Response Inline Query:** Como ya se comentó en el inline query, una vez elegida una opción dentro de las presentadas en el inline keyboard al realizar una inline query, el sistema realizará el proceso de recomendación de eventos a partir de la opción

seleccionada. En el caso de estar el bot en la conversación, al usar el comando /more, será esta consulta la identificada como última por el sistema.

- **Inline Callback:** Este tipo de mensaje se produce como respuesta a consecuencia de accionar algún mecanismo ya sea un Inline Button o de otro tipo. En este tipo de mensaje se establece un id-valor a devolver al bot cuando se acciona dicho procedimiento. En este proyecto se van a producir inline callback en consecuencia de que el usuario un pulse inline button. Los inline callbacks que se pueden recibir son:

Inline Button	Callback	Descripción
Botón más eventos	More	Este callback devolverá eventos de la recomendación realizada, trayendo eventos del api correspondientes a la siguiente página. En caso de no existir más eventos devolverá un mensaje de “no hay más eventos” y no mostrará más el botón correspondiente a este callback.
Botón eventos anteriores	Less	Este callback devolverá la página anterior del Api de la recomendación realizada. En caso de no haber página anterior, el botón no saldrá.
Botón Valorar recomendación	eval	Este callback obtendrá las categorías de la recomendación realizada y mostrará un teclado del 0 al 5 para valorarlos, siendo el 0 el menor valor posible y 5 el máximo.
Botón de valoración realizada	evald	Este callback obtendrá la valoración realizada de una recomendación y la procesará para proceder con su almacenamiento.

Tabla 5 Inline callbacks permitidos en el bot de Telegram

Tipos de contenido

El tipo de contenido va a definir la acción a realizar dentro del bot. Los tipos de mensajes aceptados son los siguientes:

- **Texto:** El tipo más común de los mensajes enviados será de tipo texto. Su almacenamiento será sencillo y a la hora de recuperarlo para predecir categorías, simplemente se agruparán los textos de la conversación y serán enviados al servicio de predicción en formato texto.
- **Imagen:** Es frecuente que se envíen imágenes en las conversaciones referentes a la conversación o a la temática del chat. El análisis de estas imágenes nos puede dar

información valiosa a la hora de extraer gustos y realizar mejores recomendaciones. Las imágenes quedarán almacenadas en el servidor de Telegram, permitiéndonos usar la url de este para manipular la imagen. Al servicio de predicción en este caso se le enviará la url y él ya se encargará de descargarla, analizarla y devolver la predicción. A la hora de guardar la información de la imagen, se guardará la url, la predicción realizada pero no la información de los colores procesados.

- **Feedback:** Tras realizar una recomendación se recibirá la valoración otorgada a los eventos evaluados. Esta recomendación será almacenada y procesada en la siguiente recomendación a realizar.

Metodología de recomendación

El bot va a realizar recomendaciones personalizadas ante petición por parte del usuario. Estas recomendaciones irán enfocadas al chat, siendo en el caso de los chats individuales, recomendaciones personalizadas al usuario. Para realizar las recomendaciones el bot hará uso de las siguientes fuentes de información:

- **Mensajes enviados:** Las conversaciones almacenadas del chat suponen un flujo de información constante que permite conocer a los usuarios participantes [41]. Estas conversaciones reflejan los focos de interés permitiendo establecer una relación entre temas de interés y espacio temporal. Es habitual que en chat sociales no varíe mucho la temática, pero es una característica que depende únicamente de la naturaleza de los integrantes. Es posible definir 4 tipos de modelos de usuario para definir todos los casos posibles:
 - **Individual:** Este tipo de usuario se encuentra solo en el grupo y por lo normal no tendrá interacción con el bot.
 - **Grupo actualidad:** Este tipo de conjunto de usuarios varían el tema de conversación acorde con la actualidad de su alrededor, siendo complicado establecerles una etiqueta concreta.
 - **Grupo abierto y curiosos:** Al igual que pasaba en el grupo de actualidad, este grupo es muy cambiante en cuanto a temática, pero no siguen ninguna línea actualidad global, sino los temas que van surgiendo de forma espontánea.
 - **Grupo temática cerrada:** Este tipo de conjunto de usuarios por lo general no varían el tema de conversación y siguen una temática fija a lo largo del tiempo
- **Valoraciones realizadas:** Las valoraciones realizadas reflejan la opinión del usuario y pueden ser usadas para conocer los gustos del mismo. En este bot las recomendaciones están centradas en las temáticas, por lo cual en ningún caso se centrará la atención en la valoración del evento como tal.

El procedimiento a seguir a la hora de realizar recomendaciones será el siguiente:

- **Recuperación de información:** Se obtendrá la información de los mensajes antiguos del chat y de las valoraciones realizadas.
- **Análisis de mensajes:** Se analizarán los mensajes con el predictor a fin de inferir su temática.
- **Calculo de gustos:** Las valoraciones realizadas tendrán un valor para cada temática, de la que el usuario o grupo de usuarios, dieron su opinión y será usada como base de la recomendación. A partir de esta base se actualizarán las valoraciones con las temáticas

extraídas de los mensajes. Esta actualización potenciará la tendencia existente en la valoración en caso de haberla. En el caso de no haberla tendrá tendencia positiva por defecto.

- **Obtención de eventos:** A partir del cálculo de gustos realizados se realizará una petición de eventos al servicio de Api, fijando las categorías a elegir con más valoradas resultantes tras el cálculo de gustos.



Ilustración 32 Metodología de recomendación

Casos de uso

Se van a definir una serie de casos de uso realizando acciones básicas que permite el servicio de bot.

- **Comando events**

Objetivo: Un usuario del chat quiere obtener los eventos que hay en una ciudad concreta entre dos fechas y compartirlo con los componentes del mismo.

Interacción: El usuario abrirá la conversación donde se encuentra el bot y podrá introducir el comando con sus parámetros en modo texto. Es requisito indispensable que el bot esté incluido como usuario dentro de la conversación.

Procedimiento: El sistema recibe el mensaje y lo procesa como texto. Comprueba si en el texto hay un comando valido para el sistema. De ser así, usa como callback ese método para procesar el comando del mensaje y sus argumentos y emitir una respuesta al usuario.

Resultado: Se obtendrá una lista de eventos mostrando fecha de inicio y fin, horas de inicio y fin, título del evento, lugar y dirección del evento, categoría de evento y comando para ampliar la información. Además, se mostrarán los botones de interacción correspondientes (more, less y evaluación)

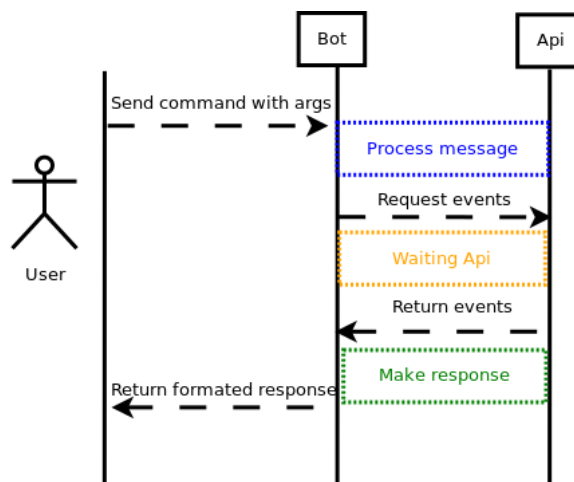


Ilustración 33 Diagrama comando events en chat individual

- **Comando recomend**

Objetivo: Un usuario del chat quiere obtener los eventos que hay en una ciudad concreta entre dos fechas, compartirlo con los componentes del mismo y que además sean acordes a los gustos de ese chat o usuario concretos.

Interacción: El usuario abrirá la conversación donde se encuentra el bot y podrá introducir el comando con sus parámetros en modo texto.

Procedimiento: El sistema recibe el mensaje y lo procesa como texto. Comprueba si en el texto hay un comando valido para el sistema. De ser así, usa el callback de ese método para procesar el comando y sus argumentos y emitir una respuesta al usuario. A continuación, obtendrá los mensajes antiguos del chat para analizarlos y las valoraciones anteriormente realizadas. A partir del análisis de estas dos fuentes de información se establecerán los gustos del chat (en el caso de ser individual coincidirán con los del usuario) y se realizará la petición de eventos teniendo en cuenta estos gustos.

Resultado: Se obtendrá una lista de eventos mostrando fecha de inicio y fin, horas de inicio y fin, título del evento, lugar y dirección del evento, categoría de evento y comando para ampliar la información. Además, se mostrarán los botones de interacción correspondientes (more, less y evaluación)

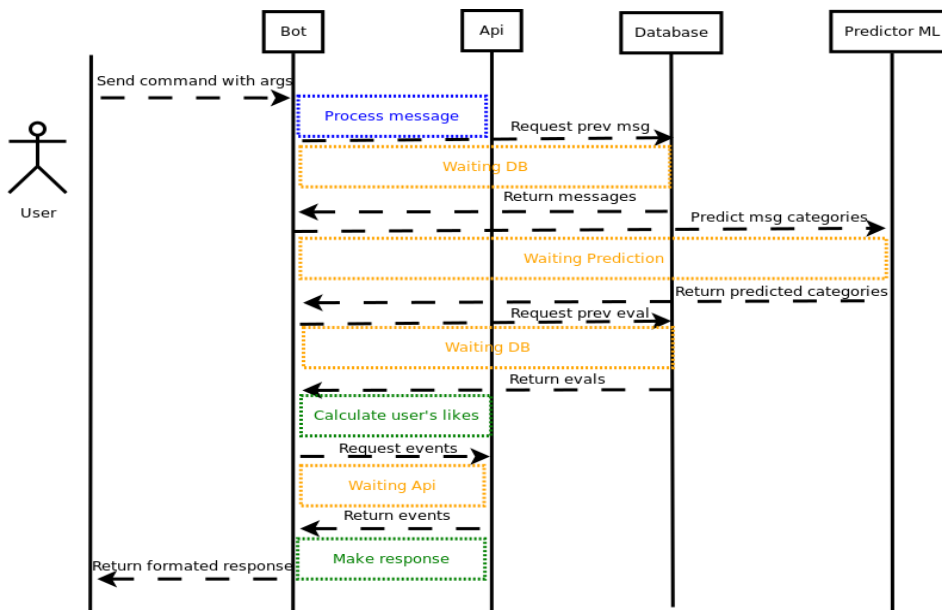


Ilustración 34 Diagrama comando events en chat individual

- **Comando de inline query**

Objetivo: Un usuario quiere obtener los eventos que hay en una ciudad concreta entre dos fechas mediante inline query en un chat en el que no está incluido el bot.

Interacción: El usuario abrirá la conversación en la que quiera compartir el contenido e introducirá el inline query con sus parámetros en modo texto. Una vez procesado el bot le devolverá distintas opciones en modo de inline keyboard. El usuario elegirá una de las opciones y el bot devolverá los resultados de dicha elección.

Procedimiento: El sistema recibe el inline query, lo procesa y genera las opciones pertinentes enviándoselas de nuevo al usuario. A continuación, el bot recibirá la opción elegida por el usuario que será procesada y se devolverán los resultados. En el caso de ser un chat en el que el bot no haya sido incluido, no se tendrá información del mismo.

Resultado: Se obtendrá una lista de eventos mostrando fecha de inicio y fin, horas de inicio y fin, título del evento, lugar y dirección del evento, categoría de evento y comando para ampliar la información. Además, se mostrarán los botones de interacción correspondientes (more, less y evaluación)

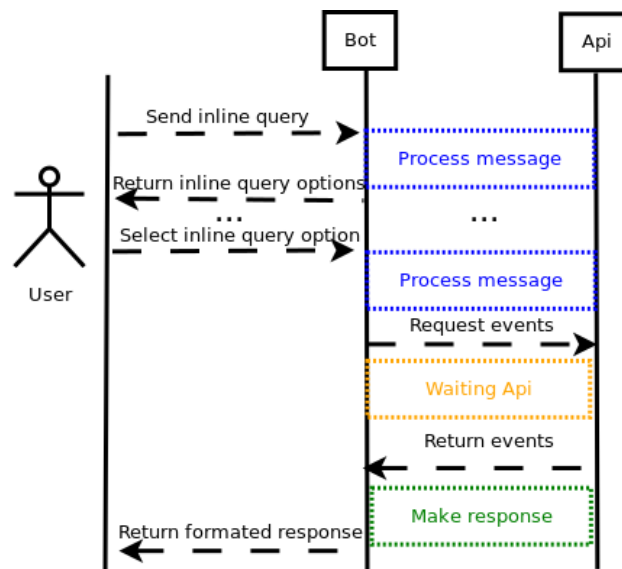


Ilustración 35 Diagrama caso de uso inline query

Inline button de more/less

Objetivo: Un usuario quiere obtener más eventos o volver a los eventos anteriores que hay en el sistema tras realizar una petición anterior.

Interacción: El usuario pulsará el botón more o less para realizar la acción deseada, enviando al sistema un callback query con la elección. Tras pulsarlo el bot devolverá los resultados correspondientes a la acción elegida.

Procedimiento: El sistema recibe el callback query, lo procesa y generará la respuesta correspondiente. A continuación, el bot recibirá la opción elegida por el usuario que será procesada y se devolverán los resultados. En el caso de ser un chat en el que el bot no haya sido incluido, no se tendrá información del mismo.

Resultado: Se obtendrá una lista de eventos mostrando fecha de inicio y fin, horas de inicio y fin, título del evento, lugar y dirección del evento, categoría de evento y comando para ampliar la información. Además, se mostrarán los botones de interacción correspondientes (more, less y evaluación)

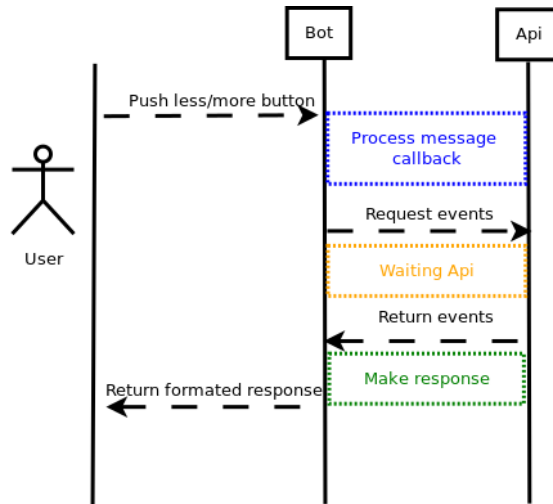


Ilustración 36 Diagrama casos de uso inline callback more/less

- **Inline button de evaluación**
- **Objetivo:** Un usuario quiere realizar una valoración sobre una recomendación de eventos realizada por el bot.
- **Interacción:** El usuario pulsará el botón de evaluar y tras pulsarlo le aparecerá en el cuadro de mensaje unos botones del 0 al 5. El usuario elegirá que valoración darle a la recomendación pulsando el botón con el valor correspondiente. Tras pulsarlo el bot el bot informará de que su valoración ha sido almacenada.
- **Procedimiento:** El sistema recibe el callback query con la petición de evaluar del usuario. El bot extraerá las categorías de los eventos y calculará las puntuaciones de cada opción para cada categoría, incluyéndolas como argumentos ocultos al usuario y se enviarán los botones con las valoraciones correspondiente. A continuación, el bot recibirá la opción elegida por el usuario que será almacenada.
- **Resultado:** Devolverá un mensaje de aceptación de la valoración realizada.

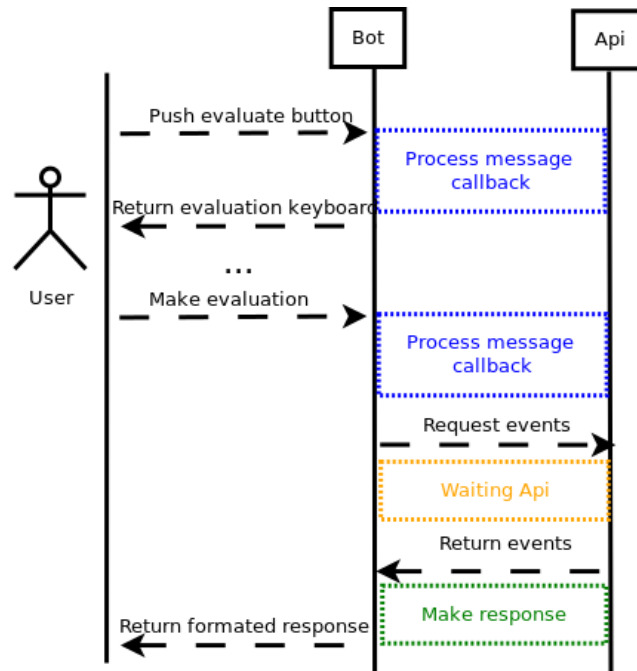


Ilustración 37 Diagrama caso de uso inline callback evaluación

Requisitos funcionales

Los requisitos funcionales definidos para el caso de la funcionalidad de Servicio, son los siguientes:

- El bot debe almacenar los mensajes.
- El bot debe permitir enviar comandos.
- El bot debe permitir enviar inline queries.
- El bot tendrá habilitados inline buttons para facilitar la interacción.
- El bot manejará tanto texto como imágenes.
- El bot se podrá añadir a conversaciones tanto individuales como grupales.

Requisitos no funcionales

Los requisitos no funcionales definidos para el caso de la funcionalidad de Predicción, son los siguientes:

- La plataforma elegida será Telegram, pero se podría implementar para otro tipo de bots como Facebook...etc.
- El sistema debe poder desplegarse de forma automática.
- El sistema contará con varios nodos de servicio.
- El bot estará comunicado con el Api de eventos.
- El bot estará conectado con el servicio del sistema de predicción.
- El bot deberá hacer uso de Webhook proporcionado por Telegram.
- El bot no debe tener acciones bloqueantes ni memoria de anteriores acciones.

3.4. Arquitectura

Teniendo en cuenta los requisitos y casos de uso presentados en el análisis de la solución propuesta, se ha elaborado el diseño del sistema que muestra la arquitectura del mismo (Ver ampliado en Anexo [Diseño de arquitectura](#)). En ella se ve la perfecta confluencia del sistema de

aprovisionamiento de información, junto con la parte de investigación de Machine Learning y la parte de servicio que hace las veces de prueba de concepto de la investigación realizada.

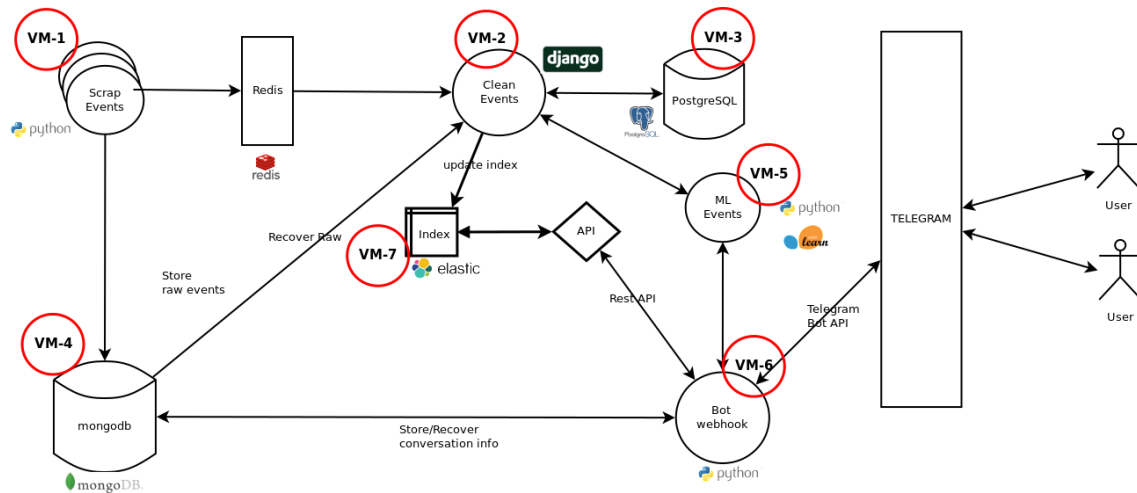


Ilustración 38 Arquitectura del sistema

En el diseño se ven definidas siete máquinas virtuales que constituirán nuestro sistema. Se podría modificar el diseño agrupando algunas de las máquinas, pero se ha decidido separar los módulos del sistema por funcionalidades a fin de mejorar la escalabilidad. La distribución de funcionalidad-máquina es la siguiente:

Máquina virtual	Funcionalidad
VM – 1	<ul style="list-style-type: none"> • Descarga Masiva de eventos sociales de internet.
VM – 2	<ul style="list-style-type: none"> • Procesamiento de eventos sociales
VM – 3	<ul style="list-style-type: none"> • Almacenamiento de eventos sociales
VM – 4	<ul style="list-style-type: none"> • Almacenamiento de eventos descargados • Almacenamiento de colores • Almacenamiento de bot de Telegram
VM – 5	<ul style="list-style-type: none"> • Predictores de Machine Learning • Análisis de Imágenes
VM – 6	<ul style="list-style-type: none"> • Webhook del bot de Telegram
VM – 7	<ul style="list-style-type: none"> • Índice de eventos

Tabla 6 Tabla de Arquitectura de máquinas del sistema

3.4.1. Arquitectura de aprovisionamiento

Como se ha podido ver en la ilustración anterior, el aprovisionamiento permitirá añadir todos los nodos de scrapping necesarios para consumir tareas de la cola de scrapping. Estos nodos únicamente tendrán conexión con la base de datos temporal de descarga de información de las fuentes y con las colas de tareas de scrapping y de aviso al sistema central de información nueva descargada. Con esta configuración se cumplen los requisitos de escalabilidad del sistema. Remarcar que es importante a la hora de crear nodos de scrapping, la necesidad del uso de proxys o el uso de direcciones Ip diferentes para evitar baneos de las fuentes.

3.4.2. Arquitectura de machine learning

El sistema de machine learning que cuenta con parte de la investigación realizada en este proyecto, se encuentra en una posición central dentro del diseño realizado ya que la mayoría de las instancias del sistema harán uso de su servicio. Es importante remarcar que en la ilustración anterior de arquitectura se muestra el flujo del sistema y no las conexiones totales entre las máquinas. A continuación, se proporciona una lista de máquinas y servicios del módulo de machine learning que usan:

Máquina	Servicio ML
VM-1	<ul style="list-style-type: none"> Inferencia de idioma. Traducción de texto.
VM-2	<ul style="list-style-type: none"> Inferencia de idioma. Traducción de texto. Inferencia de categorías en texto. Extracción de sesiones.
VM-6	<ul style="list-style-type: none"> Inferencia de categorías en texto. Inferencia de categorías en imágenes.

Tabla 7 Uso de servicios de Machine Learning en las distintas máquinas

3.4.3. Arquitectura de servicio

A la hora de dar servicio se diseñó teniendo en cuenta varios detalles para poder dar una tasa de respuesta rápida y que los usuarios se sientan cómodos con la interacción en la prueba de concepto. A continuación, se comentan las medidas tomadas para llevar este propósito a cabo.

- **Api + Índice**

Como ya se ha comentado anteriormente se va a usar un Api Rest para servir la información del sistema central y conectar todos los productos resultantes de este. En este caso el único producto a conectar es la prueba de concepto realizada en forma de bot de Telegram. En el caso de tener requerimientos más altos de servicio se podrá incluir varios puntos de Api tras un balanceador de carga a fin de mejorar el rendimiento y dar servicio a todos los puntos del sistema.

El Api se alimenta de un índice en vez de la base de datos central directamente. Esta medida otorga al sistema de una rapidez considerable en el proceso de recuperación de información por parte del Api, a costa de la gestión del mantenimiento. El tipo de índice usado en este proyecto (*Elasticsearch*), necesita de una solución propia y particular para

realizar el mantenimiento del índice [42], llevándose a cabo como una tarea del sistema central, que se produce tras la actualización de un evento.

- **Bot**

Dentro del bot se ha optado por realizar una solución propia y descartar el uso de librerías de terceros, ya que estas no se adaptaban a la idea de rendimiento seguida en todo el sistema. En nuestro bot se va a separar la funcionalidad principal del bot en dos submódulos. Por un lado, estará el enrutador de mensajes y por otro el procesador.

Nginx: El punto de entrada de la máquina que aloje la parte enrutadora del bot, se conectará con Telegram a través de Nginx. Este servidor nos permitirá olvidarnos de los requerimientos de puertos (443, 80, 88, 8443) que tiene Telegram a la hora de establecer un webhook y configurar la conexión entre nuestro bot y Nginx por el puerto que se quiera. En el caso del entorno actual donde se desplega el sistema no existen problemas, ni restricciones complicadas de puertos, pero en otros casos puede ser una buena solución para resolver este inconveniente.

Enrutador de mensajes: Como su propio nombre indica se va a encargar de recibir las actualizaciones de mensajes que provienen de Telegram y a partir del tipo de mensaje que sea, se colocará en una cola u otra como argumento de una tarea del sistema. El tiempo que lleva realizar la tarea de enrutamiento de mensajes es muy breve por lo que no parece suponer un cuello de botella. En caso de serlo se podrían levantar instancias y balancear carga entre ellas, pero es requisito que únicamente haya un punto de entrada de mensajes con Telegram.

Procesador de mensajes: Esta parte del bot se encargará de recoger las tareas de las colas, procesarlas correspondientemente y generar una respuesta en caso de ser necesario. En principio no se han definido instancias específicas para un tipo de cola, pero podría hacerse en caso de haber cuello de botella, así como levantar más instancias. En esta parte no hay limitación alguna en cuanto a las instancias, ya que son simplemente instancias de procesamiento que consumen tareas de Celery.

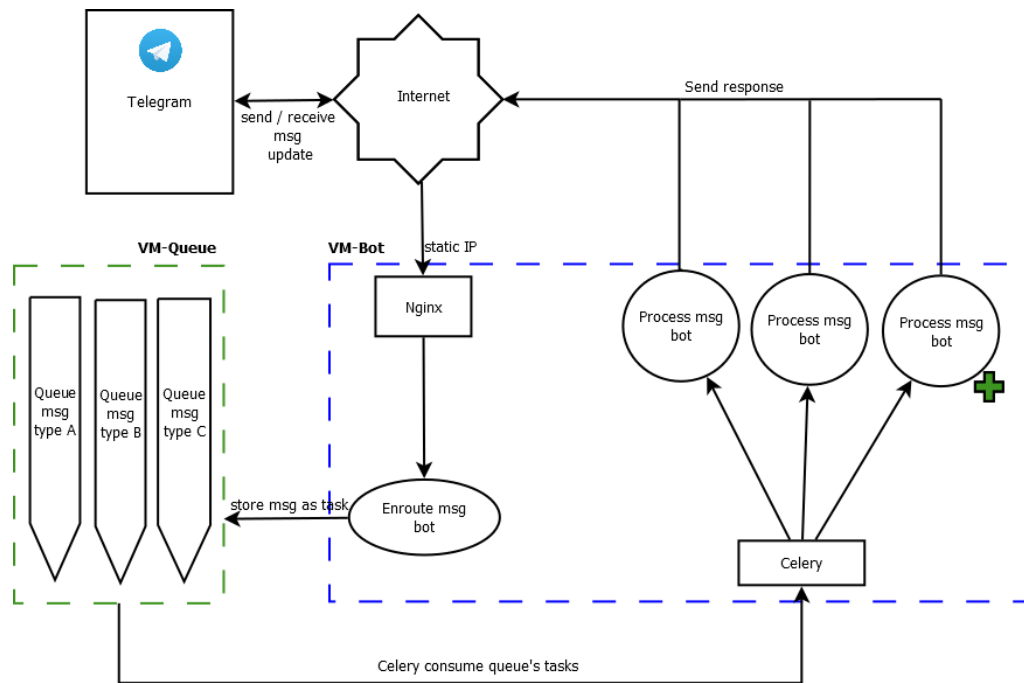


Ilustración 39 Arquitectura de servicio del bot

Seguridad

Se debe preservar la integridad del sistema restringiendo el acceso a las máquinas del mismo únicamente a personal autorizado. Cada máquina será independiente a la hora de permitir a un usuario el acceso. Estos permisos de acceso se darán durante el despliegue de la máquina y se actualizarán en caliente en caso de ser necesario.

En cuanto al acceso al repositorio de código se han definido roles de acceso y se dará de alta a los usuarios según corresponda. Se habilitará además una clave de despliegue en producción para controlar este ámbito y proteger el código.

Logs

Cada máquina gestionará de forma autónoma sus logs, almacenándolos en una ruta que seguirá el siguiente formato: `/srv/<nombre del proyecto>/logs/`. De momento el sistema no incorpora un sistema de extracción, unificación y servicio estadístico de logs, pero un futuro sería una idea interesante a realizar.

3.5. Protocolos

Se van a definir una serie de protocolos de funcionamiento a la hora de realizar ciertas tareas en el sistema.

Funcionalidad del sistema

Para fomentar la modularidad y escalabilidad del sistema, el sistema va a hacer uso *Celery* como herramienta de gestión de tareas. Esta herramienta permite la distribución y manejo de tareas almacenadas en colas, así como la definición de tareas periódicas usadas sobretodo en ámbitos de mantenimiento. Como cola de mensajes se hará uso de *rabbitmq*, elegida por delante de *redis* debido a caídas del sistema. A continuación, se muestran las tareas gestionadas por este sistema:

- **Tareas de Scrapping**

Se enviarán tareas de scrapping de forma periódica cuando la programación así lo indique desde el sistema central. Estas tareas serán consumidas por las instancias de máquinas de scrapping existentes en el sistema. El trabajo a realizar será el de descargar la información de una fuente de eventos. Para ello se pasará como argumentos el nombre de la fuente a scrapear y algún parámetro específico de la fuente o de filtrado que necesite.

- **Tareas de recuperar eventos**

Una vez scrapeada una fuente se enviará una tarea al sistema central de eventos, avisando que hay eventos almacenados en la base de datos temporal de descargas, para ser procesados e introducidos en pipeline del sistema. Como argumentos se pasarán los identificadores de los eventos a recuperar e introducir en el sistema.

- **Tareas del pipeline de eventos**

Las distintas tareas de procesamiento de eventos del pipeline también pueden ser derivadas al pipeline en caso de saturación, tareas costosas de procesamiento o necesidad de hacer uso de varias máquinas. Para ello cada proceso tendrá una cola y se deberá habilitar la opción, ya que por defecto el sistema central es único, existiendo solo un pipeline.

- **Tareas de mantenimiento**

Los eventos requieren una serie de tareas de mantenimiento periódicas a fin de tener curada y limpia la base de datos que se va a servir. Estas tareas están relacionadas con el tiempo. Es importante contar con el hecho de que el paso del tiempo afecta a los eventos, por lo que se deberán retirar del servicio aquellos eventos cuya fecha haya pasado. Esta tarea los des-publicará y los quitará del índice de servicio.

- **Tareas de recepción de mensajes.**

En el bot todos los mensajes recibidos serán transformados en tareas. Según el tipo de mensaje se conformará una tarea u otra, introduciendo dichas tareas en una cola diferente para cada tipo. A continuación, los procesadores de eventos consumirán las tareas de todas las colas según su disponibilidad.

Descarga de imágenes

Como ya se comentó anteriormente es importante respetar los plazos de descarga de imágenes, tanto en temas de denegación de acceso como en temas de optimizar la descarga, obteniendo la mayor cantidad de imágenes posibles en cada ejecución. En el proceso de descarga, únicamente se guardará una referencia en la base de datos de la imagen y se obtendrá la imagen de la url proporcionada, almacenándola junto a las otras imágenes de la categoría a descargar en cuestión.

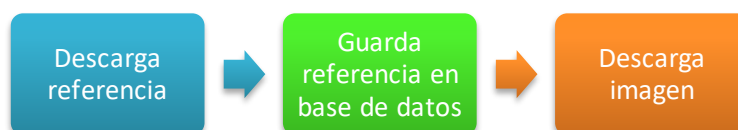


Ilustración 40 Protocolo de descarga de imágenes

Carga de imágenes

Previo a la carga de imágenes, es importante realizar un filtrado manual de las imágenes descargadas, evitando meter imágenes irrelevantes respecto a la temática elegida. Una vez realizada la limpia de imágenes, se seleccionarán las imágenes a cargar y el sistema las procesará actualizando con el resultado obtenido, la referencia en la base de datos. Es importante resaltar que el resultado se almacenará para cada número de clúster con el que se procese una imagen. De esta manera se ahorrará tiempo a la hora de procesar de nuevo imágenes cada vez que se quieran obtener los colores predominantes de una clase. En el caso de que una clase no tuviera ese número de clúster procesado, habría que realizar el proceso entero y actualizar la referencia.

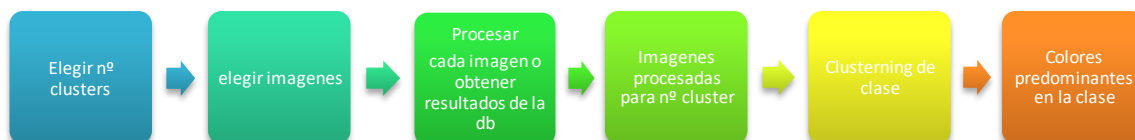


Ilustración 41 Protocolo de carga de imágenes

Descarga de eventos

Dentro del sistema central habrá un módulo que ordene al scraper de eventos, que se active para descargar la información de una página. Dicho modulo tendrá la programación periódica de las fuentes a scrapear y los argumentos que pasar en cada ejecución. La descarga de eventos de varias fuentes se puede realizar en paralelo siempre que no sature los recursos de la máquina. El proceso de descarga seguirá siempre la misma línea de ejecución, primero se obtendrá el endpoint a scrapear realizando la petición correspondiente. Una vez obtenida la información se pasarán por la plantilla de la página donde se indica la localización de la información a extraer. A continuación, se pasa al formato establecido y es almacenada en la base de datos, enviando una señal al sistema de procesamiento para que planifique su entrada.

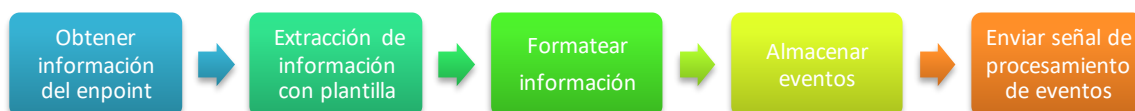


Ilustración 42 Protocolo de descarga de eventos

Actualización de eventos en la fuente

Es habitual que, en las fuentes de eventos de internet ocurran cambios frecuentes en la información de los propios eventos por circunstancias varias como, por ejemplo, la variación de la fecha o del precio. En estos casos el propietario de la fuente actualizará la información y el sistema la obtendrá actualizada durante el siguiente proceso de scraping programado. Una vez obtenida se actualizará en la base de datos de descarga y levantará la señal en el sistema para volver a procesar ese evento. En el caso de no haber ningún cambio en el evento no se levantará ninguna señal. Este proceso depende en gran parte de la programación de scrapeo de las fuentes, ya que una buena programación organizada permitirá tener actualizada la base de datos de eventos de forma eficiente y dando un servicio de calidad.



Ilustración 43 Protocolo de actualización de eventos en la fuente

Actualización de eventos en el sistema

Una vez levantada la señal de procesamiento, el sistema como hace habitualmente, cogerá de la base de datos de descarga el evento en cuestión para su procesamiento. En este proceso se comprobará la existencia del evento en la base de datos y en el caso de existir se iniciará el procesamiento con la nueva información. Este proceso implica la pérdida desactualizada de la información del evento del sistema, así como de su estado en el sistema, es decir, si un evento se encuentra publicado y llega información actualizada de él, se despublicará e iniciará el procesamiento desde el principio. Presumiblemente a menos que la información actualizada de problemas en el procesamiento, debería igualar o mejorar el estado antiguo del evento en el sistema.

Publicación de eventos

Al contar en los requerimientos con la necesidad de dotar al sistema de una alta tasa de respuesta frente a peticiones, se ha incluido en el sistema un indexador de contenido. Este permite un rápido acceso a los datos reduciendo la tasa de respuesta. En el indexador de servicio únicamente se introducirán los eventos sociales publicados que hayan pasado por todo el procesamiento del sistema, siendo labor del propio sistema la actualización y limpieza del mismo ante cambios en los eventos [42].

Geolocalización de eventos en el sistema

La geolocalización de los eventos en el procesamiento del sistema consiste en completar la información geográfica de los eventos para su servicio a través de este parámetro. Para poder geolocalizar un evento es necesario tener los bordes del mundo previamente introducidos a varios niveles (país, región, subregión, ciudad) y obtener su localización geográfica para posicionarlo. Para ello se hará uso del Api de Google Places [43] o en su defecto del Api de Bing [44]. En ellas se buscarán las direcciones o informaciones obtenidas del evento en la fuente referentes a su localización. En el caso de tener ya las coordenadas geográficas desde la fuente, se haría uso este método para completar su dirección en caso de no tenerla. Una vez que se tiene toda la información de la localización del evento, se realizarán consultas sobre los bordes existentes en la base de datos obteniendo una jerarquía de regiones en la que se encuentran sus coordenadas. Por ejemplo, si se tiene un evento de un concierto en Madrid, con las coordenadas (40.4, -3.68) el sistema devolverá que está en Madrid, Comunidad de Madrid, España. El sistema únicamente llega a nivel de ciudad en cuanto a profundidad geográfica se refiere.

Carga de bordes de regiones

Como se vio en el apartado anterior sobre geolocalización de eventos, es imprescindible tener dentro de nuestra base de datos los bordes de las distintas entidades geográficas de nuestro sistema. Hay dos formas de obtener los bordes, dibujándolos uno a uno a mano o descargarlos de un repositorio online. En este caso se ha optado por hacer uso de las siguientes bases de datos online de bordes geográficos:

- **Geonames** [45]: Base de datos geográfica que será usada para la obtención de los nombres y jerarquía de las entidades geográficas, estableciendo toda la estructura geográfica del sistema, teniendo únicamente que completarlo más tarde con los bordes. La información contenida en este servicio es de gran calidad y completitud, siendo necesarios pocos retoques. Es importante tener en cuenta que, a pesar de la gran cantidad de detalle existente, en las bases de datos de bordes no existe tanto detalle, por lo que se deberá cargar únicamente los elementos de interés para nuestro sistema.
- **Gadm** [46]: Base de datos de bordes que contiene información sobre todo el mundo a distintos niveles. Es necesario recalcar en algunos países la falta de algún nivel o de la precisión de estos, siendo necesario hacerlos o retocarlos.

- **Koordinates** [47]: Base de datos que contiene mapas de bordes de distinta índole entre los que se encuentran los bordes de países, regiones y ciudades. Los bordes encontrados en este servicio son extremadamente precisos, hasta el punto de tener cuidado de que no solapen con los de Gadm.

El proceso de carga de bordes se realizará de país en país especificando los niveles a cargar en cada caso. Primeramente, se montará la jerarquía de país, región, ciudad y luego sobre esta, se añadirán los bordes. El cruce entre las entidades y los bordes se hará mediante el nombre de las mismas. Una vez terminada la carga de bordes es importante realizar una revisión manual en busca de fallos. Únicamente se realizarán creaciones o modificaciones de bordes en el caso de ausencia o imprecisión de algún borde. La mejor herramienta para tratamiento, revisión y administración de bordes es Qgis [48], permitiendo además sincronizar con la base de datos de forma sencilla y realizar consultas geográficas a través de interfaz.



Ilustración 44 Protocolo de carga de bordes

Actualización de predictores

Los predictores del sistema harán uso de la estrategia de usar ficheros del tipo *pickle* para guardar el entrenamiento realizado. El fichero *pickle*, es un tipo de fichero del lenguaje Python, que representa una serialización de una estructura de objeto, permitiendo almacenarlo manteniendo su estado, para en un futuro volver a usarlo. Haciendo uso de este fichero se podrán guardar los entrenamientos realizados de forma que no haya que procesar de nuevo toda la información para realizar una predicción [49]. También facilita mucho la actualización del modelo ya que únicamente modificando el fichero *pickle*, se realizará el cambio de modelo predictivo, pudiendo tener incluso control de versiones. Es importante comparar efectividad de predicción entre modelos antes de sustituir uno por otro, realizando el test con el mismo conjunto de prueba.



Ilustración 45 Protocolo de actualización de predictores

4. Desarrollo

4.1. Proyectos y Repositorios

Se va a crear un proyecto y repositorio por cada módulo del sistema, separado de los demás y un proyecto que será el encargado de desplegar el entorno. Con este proyecto de despliegue es posible poner en funcionamiento el proyecto tanto en local como en producción de manera que no dependa de un servicio externo. Este proyecto hará uso de las tecnologías Vagrant y Ansible para llevar a cabo todo el proceso de levantar las máquinas virtuales de cada módulo y configurarlas sin tener que descargar ningún repositorio más. El proyecto de despliegue está alojado en https://bitbucket.org/EPS_UAM_TFM_MMP/tfm_bender.

4.2. Tecnologías y lenguajes utilizados

Parte del éxito del desarrollo del proyecto realizado ha sido el poder juntar distintas tecnologías aprovechando los beneficios de cada una de ellas. Se han utilizado las siguientes tecnologías para la realización de este sistema:

- **Python**

Se ha utilizado este lenguaje de programación para realizar todo el sistema. Ha sido el lenguaje elegido por su gran cantidad de librerías y módulos existentes, que facilitan mucho la labor de programación en ámbitos como el Machine Learning.

- **Django**

Framework de Python que permite el desarrollo de programas web con entorno de administración de forma sencilla. A partir del diseño de las estructuras de datos del sistema, el framework proporciona persistencia a través de su mapping ORM y junto a la alta capacidad de reutilización de elementos, hacen de este framework una herramienta ideal para ser el centro de este proyecto. En este caso la base de datos usada junto a Django fue Postgresql.

- **Django restframework**

Framework de Django que proporciona funcionalidad Rest al sistema y la posibilidad de crear un web Api sencillamente. Este módulo es capaz de gestionar tanto el servicio de los elementos como la seguridad (métodos de autenticación de los usuarios).

- **Celery**

Es un sistema de gestión de tareas a través de distintas implementaciones de colas de mensajes, permitiendo realizar sistemas escalables y con multiprocesamiento. En nuestro sistema se va a usar en muchos módulos y para distinta índole (ver [Funcionalidad del sistema](#)). El funcionamiento de este elemento depende del funcionamiento de las colas que lo alimentan.

- **Scikit-learn**

Librería de machine learning para Python que proporciona una abstracción completa sobre los algoritmos de machine learning que alberga. El uso de la librería puede llegar a ser complejo, teniendo que realizar mejoras continuas en la lógica desarrollada a medida que se aprende su funcionamiento. La documentación existente es excelente, aunque extensa, más se recomienda leer previamente a su uso.

- **Postgresql**

Sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Se usará en el sistema esta implementación de base de datos SQL como almacenamiento central de eventos procesados.

- **Postgist**

Módulo de Postgresql que añade funcionalidad para la realización y gestión de consultas geo-espaciales.

- **Mongodb**

Base de datos no relacional (NOSQL) de código abierto orientada a documentos que representa una solución ágil y escalable a la hora de guardar estructuras de datos. En el sistema se usará para almacenar la descarga de eventos, el procesamiento de las imágenes y los chats recopilados por el bot.

- **Elasticsearch**

Indexador de información que permite el servicio de información de forma rápida. En este proyecto se va a usar de indexador para almacenar la información a servir en el Api. En el caso de poner el sistema en producción bajo un contrato económico, sería recomendable indexar los logs de todo el sistema en una máquina con Elasticsearch y hacer uso de una solución como Kibana, para la visualización gráfica de los mismos.

- **Rabbitmq**

Sistema de colas elegido para usar en el sistema y en el cual se almacenarán las distintas tareas del sistema a consumir por Celery. Se eligió esta solución por delante de Redis por su estabilidad, ya que Redis en ocasiones sufría caídas (fallo sin determinar el origen) y paralizaba el sistema.

- **Vagrant y Ansible**

Permiten la creación, despliegue y configuración de máquinas virtuales a partir de la definición de roles. En este proyecto se han creado roles de despliegue específicos de cada máquina y roles de servicio comunes a varias, permitiendo la reutilización de roles entre máquinas.

4.3. Despliegue

La máquina de despliegue será el servidor con dirección **helicon.ii.uam.es** habilitado dentro de la sala de servidores de la EPS UAM. En dicho servidor se ha procedido a crear un usuario, descargar el repositorio de despliegue y proporcionarle a este una clave ssh habilitada únicamente para despliegue del proyecto en esa máquina. Como ya se explicó antes, el proyecto desplegará las máquinas necesarias que se comunicarán entre sí mediante direcciones internas. La lista de máquinas y sus características técnicas se muestra a continuación:

Dirección IP	Nombre de la máquina	Memoria Asignada
192.168.168.20	Mongo	1024
192.168.168.30	Flash	2048
192.168.168.40	Roger	1024
192.168.168.50	Messagequeue	1024
192.168.168.60	db	1024
192.168.168.70	Albert	2048
192.168.168.80	Indexer1	2048
192.168.168.85	Nikola	1024
192.168.168.90	Bichibot	1024

Tabla 8 Despliegue de máquinas del sistema

De todas las máquinas a desplegar, únicamente la máquina de **bichibot** necesita *port-forwarding* para conectar el webhook de Telegram. El *forwarding* lo hará sobre el puerto **8443**. En caso de existir algún tipo de firewall en el servidor es necesario habilitar el acceso.

El sistema al completo puede ser reproducido

Se ha añadido un anexo a este documento (Ver Anexo [Instrucciones de uso para el despliegue](#)) en el que se incluyen las instrucciones básicas para realizar el despliegue de este proyecto con éxito.

4.4. Dependencias Externas

Todas las máquinas hacen uso de software externo de terceros para su funcionamiento. Para su instalación de forma automática se tomarán las siguientes medidas

Creación de roles de despliegue

Se van a crear roles de Ansible para el despliegue de los componentes de las máquinas. Estos roles se podrán combinar y reutilizar en varias máquinas aportando de esta manera modularidad y sencillez a la hora de configurar los componentes y herramientas que usará cada máquina. Estos roles harán uso de variables comunes definidas, así como de otras específicas al módulo concreto. Dentro de los roles generados para este proyecto se pueden distinguir dos tipos:

- **Roles de despliegue**

Serán los roles encargados de desplegar la configuración de cada máquina del sistema haciendo uso de roles dependientes.

- **Roles de herramientas**

Serán roles genéricos que instalaran recursos de terceros que hará uso la máquina a desplegar. Estos roles se añadirán a los roles de despliegue como roles de dependencia dentro de los roles de despliegue, a menos que se aúnen varios servicios completamente independientes en una máquina con el fin de optimizar el número de máquinas en el sistema.

Todos los roles desarrollados van a seguir una estructura de organización basada en carpetas y ficheros yml propios de Ansible. Cada rol tendrá su propia carpeta y contará con las siguientes subcarpetas en su jerarquía:

- **Defaults:** Contiene la configuración por defecto del rol a menos que se le definan otros valores en el proceso de despliegue.
- **Handlers:** Contiene manejadores de eventos que permiten el control de acciones durante el despliegue. Por ejemplo, si tras instalar un servicio se debe reiniciar este antes de realizar otro paso del despliegue, se creará un manejador de eventos que este pendiente y lo realice.
- **Meta:** Contiene las dependencias con otros roles del rol en cuestión. Los roles dependientes se desplegarán antes del rol objetivo.
- **Tasks:** Tareas a realizar para el despliegue del rol. En este proyecto se han agrupado en *main* (contiene las tareas principales), *deploy* (contiene tareas de despliegue de código, repositorios...etc.) y *packages* (contiene la instalación de librerías necesarias en el rol).
- **Templates:** Contiene plantillas de archivos de configuración que se remplazarán durante el despliegue del rol por los archivos de configuración originales, para configurar el rol. En las plantillas se suelen modificar valores a través de variables las variables definidas tanto global como localmente.

- **Vars:** Contiene variables propias del rol a definir para su configuración. Además de variables de configuración del rol, también se encuentra la definición de variables de entorno del rol.

Dependencias Python

Dentro de todos los proyectos desarrollados en Python se incluirá una serie de ficheros que definirán las dependencias del mismo en cuanto a librerías se refiere. Los ficheros se almacenarán dentro de la carpeta *requirements* en la raíz de los proyectos y en su interior albergará todos los ficheros de dependencias. Estos ficheros mantendrán una estructura de archivo de una dependencia por línea, en la cual se seguirá el formato de *<nombre de la dependencia>=<versión>*. Manteniendo las dependencias controladas con esta metodología, al desplegar un proyecto se instalarán las dependencias correctamente, permitiendo reproducir el sistema en cualquier momento y evitando problemas de incompatibilidad por actualización de alguna de ellas.

4.5. Test funcionales y Documentación

Cada proyecto cuenta de forma propia de test funcionales y de generación automática de documentación técnica referente al proyecto en cuestión. En el caso de los test, la estructura seguida ha sido la propia de las buenas prácticas de Python [50], en la cual cada módulo tendrá sus propios test dentro de una carpeta *'tests'*. Se ha seguido la nomenclatura de nombrar a los ficheros de testing igual que al fichero que contiene la funcionalidad que testean.

En cuanto a la generación automática de documentación se va a usar *sphinx* para generar la documentación a partir de una definición de reglas y recogiendo los comentarios incluidos dentro del código. Estas reglas se encontrarán en una carpeta llamada *'doc'* que se encontrará siempre en la raíz principal del proyecto. Dentro de las diferentes opciones que ofrece *sphinx* se podrá elegir el formato (pdf, html, epub...) que más se adecue a las necesidades del usuario a la hora de exportar la documentación.

Las dependencias de funcionalidad tanto de testing como de la documentación se han incluido en las dependencias.

En todos los proyectos se ha incluido un fichero Makefile que contendrá funcionalidad para realizar el testing y la documentación, con unos parámetros ya definidos por defecto. En el caso del testing se ha añadido el módulo de coverage que saca por pantalla el porcentaje de cobertura que otorgan los test realizados. En el caso de la documentación está fijado que el formato por defecto en el que se muestre la documentación sea html, dejando la documentación dentro de la carpeta *'doc'* en una carpeta llamada *'_build/html'*.

5. Evaluación

5.1. Pruebas funcionales del sistema

El sistema cuenta con test funcionales en la mayoría de proyectos. Como ya se dijo anteriormente, en el caso de haber módulos dentro del proyecto se ha seguido una estructura modular y se han realizado empleando la funcionalidad de testing propias de Python y el uso de la técnica de *mock* de funcionalidades. Este tipo de test consiste en reproducir toda la funcionalidad del sistema, “falseando” o “controlando” (‘mockeando’) la salida de funciones externas para llegar al escenario a reproducir. A continuación, se muestra una tabla con el resultado de los test y el porcentaje de cobertura en cada uno de los proyectos:

Nombre de proyecto	Nº de tests	Cobertura	Estado
Roger	54	99%	Correcto
Albert	314	78%	Correcto
Flash	74	76%	Correcto
Bichibot	40	74%	Correcto

La realización de test ha sido evolutiva a medida que se construía el sistema y como se puede comprobar los últimos módulos tienen una gran deficiencia en este aspecto a corregir como trabajo de futuro. El hecho de no tener test no indica el mal o incorrecto funcionamiento del sistema, sino que hace hincapié en la falta de certificación del correcto funcionamiento del mismo.

Se ha probado la generación automática de documentación en todos los proyectos, siendo satisfactorio el resultado.

5.2. Pruebas clasificación y análisis de imágenes

Para la realización de pruebas de clasificación sobre imágenes es necesario la descarga previa de dos sets diferenciados de imágenes. Dichas imágenes provendrán de la misma fuente y con las mismas categorías. El primero de los sets se usará para entrenar el modelo y debe ser más grande que el segundo, sin llegar al sobre-entrenamiento. Es importante a la hora de realizar los test que puedan ser posteriormente reproducibles por lo que se almacenará tanto la configuración de imágenes utilizada para entrenar y testear, así como el clasificador obtenido en formato de pickle.

Pruebas Instagram

Las pruebas de Instagram no fueron de gran calidad, llegando en el mejor de los casos a un porcentaje del **30%** de precisión. Este mal porcentaje de precisión se atribuyó a la mala calidad de las imágenes de la fuente. Como ya se dijo antes, las imágenes encontradas en la fuente de Instagram, contienen selfies, pornografía, anuncios y muchas de las imágenes llevan filtros que modifican el color de estas (Ver [Análisis de filtros Instagram](#)). Al no ser relevantes estas pruebas de clasificación se decidió cambiar de fuente y dejar las imágenes de Instagram para experimentos de otra índole.

Pruebas Imagenet

Las pruebas realizadas sobre Imagenet, representaban un gran pozo de esperanza debido a que es una base de datos usada en el mundo de la investigación a la hora de obtener recursos de imágenes clasificadas con una ontología extensa según la naturaleza que representan. Se realizaron 2 experimentos usando esta fuente de imágenes, en el primero se usaron tanto las imágenes de esta fuente tanto para el entrenamiento como en el test, y en el otro experimento se usó la fuente de Flickr como entrenamiento e Imagenet de validación. La temática explorada en estas pruebas fue algo diferente, ya que se enfocaron a la temática de deportes. Los resultados obtenidos en ambos casos fueron decepcionantes, obteniendo un **32%** de precisión en el caso del uso de imágenes de

Imagenet como set de entrenamiento y test y bajando la precisión hasta un **18%** en el caso de usarlas como como validación de la fuente de Flickr. El motivo de los malos resultados radica en dos aspectos:

- **Fuente genérica:** La fuente de imágenes de Imagenet parece una buena solución en temas de investigación debido a su cantidad de imágenes clasificadas y a su ontología de clasificación que permite obtener imágenes genéricas que representan una categoría. En el experimento se buscaron imágenes bajo la categoría de deportes, y en este concepto se observó que en la base de datos había muchas imágenes representativas de objetos más que de acciones que representarían un deporte en sí, lo cual llevo que al comparar imágenes con otra fuente o entre la misma fuente, dirán resultados malos. El ejemplo claro de este problema se puede ver por ejemplo en la categoría futbol. Dentro de esta categoría había muchas imágenes de pelotas, de distintos colores, formas y en entornos diferentes, pero ninguna representaba en sí que se estuviera jugando al futbol que era algo fundamentalmente para poder considerarla imagen social según nuestra definición enunciada.
- **Mezcla de fuentes:** El hecho de mezclar fuentes en un experimento no ha dado resultados positivos en este trabajo y como se puede ver en este caso sigue sin darlos. Esto es debido a la diferencia de naturaleza de las fotos que alberga cada base de datos. En el caso de Flickr se podrían definir casi todas las imágenes como imágenes sociales representativas, pero en el caso de Imagenet se podrían catalogar como imágenes de representativas de escaparate, que dan una visión de elementos relacionados pero que no definen una acción social. Debido a esto, es complicado realizar los distintos sets con imágenes de distintas fuentes. Podría hacerse los sets mezclados, pero aun así existiría la problemática de la mala calidad de imágenes en cuanto a representación de imágenes sociales dentro de algunas fuentes.

Pruebas Clustering

Las pruebas de clustering se realizaron en una fase temprana de desarrollo del sistema de entrenamiento por ellos los porcentajes resultantes no llegan a ser altos, pero son indicativos de la mejor configuración a utilizar en el clustering. En la prueba realizada se enfrentaban el número de clúster usado en la clusterización de imágenes frente al número de clúster usado en la clusterización de cada categoría. La muestra elegida era de unas 300 imágenes de entrenamiento y 150 de test. Los resultados elegidos fueron los siguientes:

Clúster1 / Clúster2	C216	64	80
16	51%	53%	49%
64	55%	61%	58%
80	50%	52%	42%

Tabla 9 Resultados pruebas de clustering

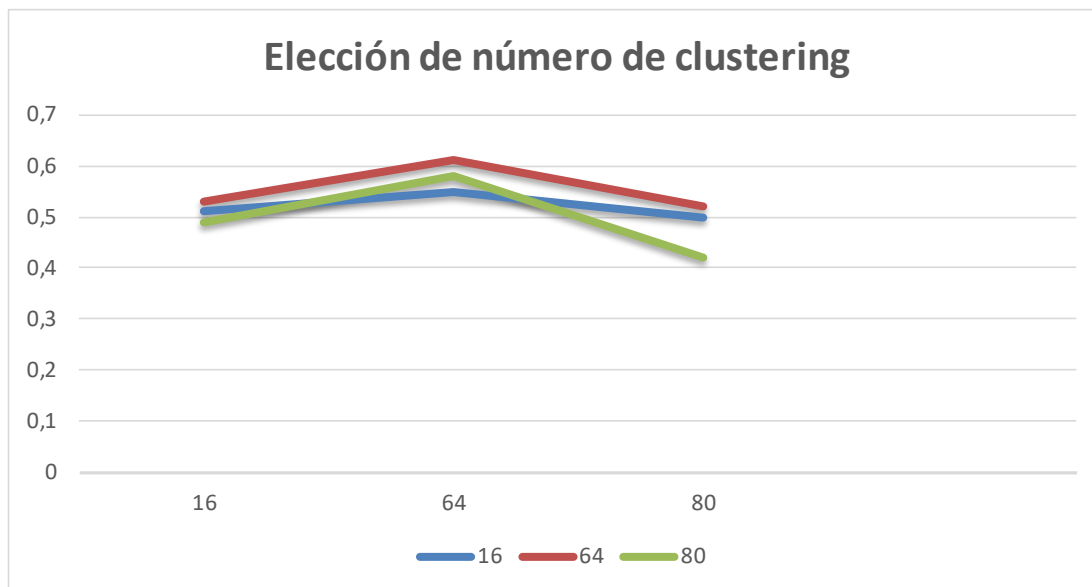


Tabla 10 Grafica pruebas de clustering

Los resultados obtenidos indican que la mejor configuración se conseguía haciendo uso tanto de 64 clúster en el clustering de imágenes como en el de categorías. No se realizaron más pruebas al respecto debido a que posteriormente a mi experimentación sobre clustering, se hizo una publicación [51] con la que se pudieron validar los resultados obtenidos. En el trabajo publicado por la universidad de Tushima se obtenían las mismas conclusiones a la hora de clusterizar imágenes, usando 64 como valor de clustering y siendo este el que les otorgaba mejores resultados.

Pruebas Flickr

Como ya se dijo anteriormente Flickr iba a ser la mejor fuente elegida en cuanto a la alimentación del reconocedor de imágenes se refiere. Para ello se irán descargando imágenes progresivamente y añadiéndolas al conjunto de entrenamiento hasta notar que se acerca en sobreentrenamiento. Para las pruebas se generarán 2 sets de imágenes, uno de entrenamiento y otro de test, pudiendo elegir la formación aleatoria de estos sets o forzar que sean de descargas diferentes. Se han realizado pruebas usando la generación de sets aleatoria y forzando a que sean de descargas diferentes, consiguiendo en ambas llegar a resultados similares, por lo tanto, parece irrelevante. En estas pruebas como ya se dijo anteriormente deberán ser reproducibles, guardando los siguientes ficheros:

- **Fichero de imágenes usadas en set:** Se guardará un fichero que contendrá las imágenes usadas en cada uno de los sets (entrenamiento, test).
- **Fichero de clustering del set:** Se guardará un fichero que contendrá los resultados del clustering computacional realizado para cada set.
- **Pickle:** Fichero con los datos del entrenamiento a sustituir en el predictor para hacer uso del entrenamiento realizado.
- **Resultados obtenidos:** Informe de datos obtenidos parciales para una configuración de entrenamiento concreta o iteración.
- **Resultados Globales obtenidos:** Informe de datos globales donde en el caso de realizar varias iteraciones se podrán observar todos los resultados elegidos a fin de detectar el mejor predictor.

Como se dijo anteriormente la carga de imágenes en cuanto a categorías se refiere en el predictor será progresiva. Para esta memoria se ha realizado una prueba con el conjunto existente en ese momento en el cual se obtuvo un **70%** de precisión haciendo uso de 7 categorías un banco de 8600 imágenes. Este resultado es muy mejorable y aún queda lejos de su límite y objetivo final, pero debido al coste de obtener imágenes y filtrarlas, como ya se dijo anteriormente, se usarán resultados parciales en la predicción. En trabajos revisados recientes [52] se conseguían precisiones muy inferiores quedándose en un **45%** de precisión, por lo que es posible afirmar que estamos ante una solución aceptable, aunque aún lejos de la perfección.

5.3. Pruebas clasificación y análisis de texto

Las pruebas de clasificación de eventos por análisis de texto se realizarán usando los eventos ya clasificados y revisados de la base de datos. Es necesario partir de un gran número de eventos para que la prueba sea de mayor calidad. Este test devolverá un informe con los resultados del entrenamiento y el test, así como el fichero pickle obtenido. En caso de ser más exitosa la prueba que el anterior modelo de entrenamiento, se procederá a sustituir el fichero pickle.

Para realizar las pruebas de clasificación se han usado 800.000 eventos que había en el sistema. La muestra obtenida tiene el sesgo de haber muchos eventos de futbol y música, ya que la mayoría de eventos de internet sigue esta temática (Ver Anexo [Distribución de eventos en categorías](#)).

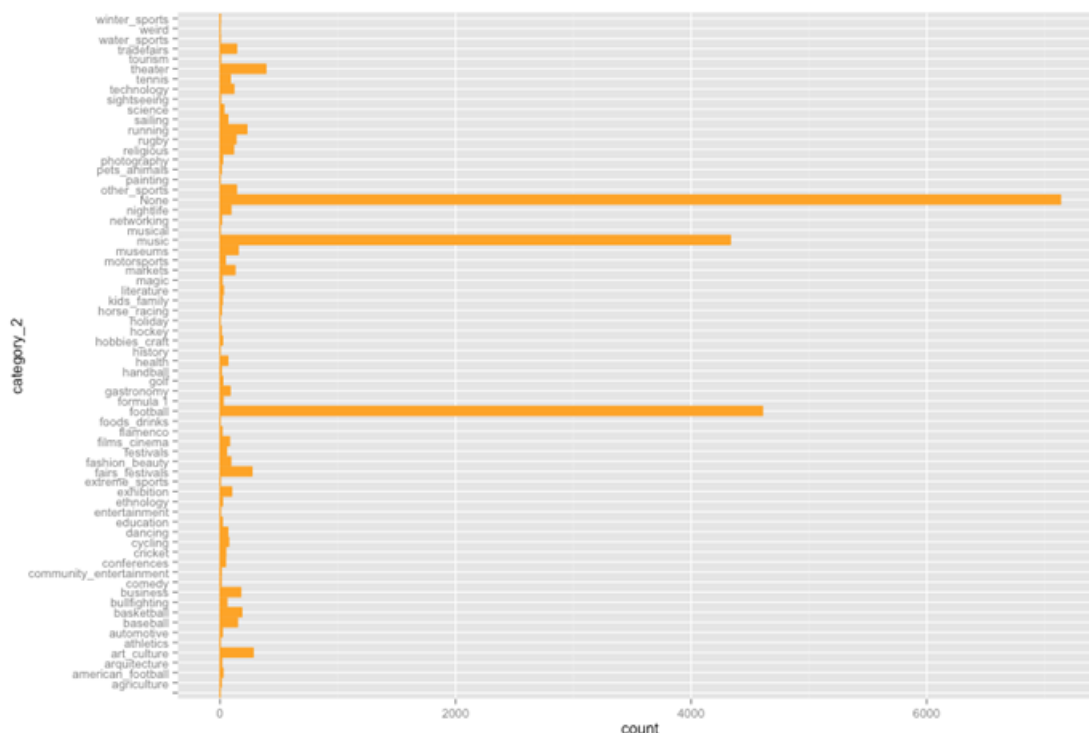


Tabla 11 Distribución de la clasificación de eventos

Aun teniendo este factor en cuenta, al realizar el entrenamiento se han obtenido valores entorno al 70-85% de precisión para los distintos algoritmos, siendo el pickle elegido para la predicción de texto de un **82%** de precisión. Este porcentaje aún necesita mejora y por lo tanto el sistema aún requiere de revisión manual periódica, pero con la incorporación de nuevos eventos a la base de datos, el sistema de predicción irá madurando y mejorando.

5.4. Pruebas bot Telegram

Participantes y grupos

En cuanto a los participantes se refiere, se va a aceptar dentro del test a cualquier persona que desee participar. De los participantes elegidos es posible establecer una serie de agrupaciones a partir de sus aptitudes. Los grupos definidos que se tendrán en cuenta en este trabajo son los siguientes:

Grupo	Edad	Conocimiento Telegram
Joven inexperto	18 – 35 años	No
Joven experto	18 – 35 años	Si
Senior inexperto	35 – 65 años	No
Senior experto	35 – 65 años	Si
Anciano	+ 65 años	No

Tabla 12 Perfiles de pruebas del bot

Los perfiles definidos tienen en cuenta la edad y conocimiento/uso de Telegram para diferenciarse entre sí. Los rangos de edades elegidos fueron definidos acordes a evitar problemas legales en cuanto a que el bot no tiene en cuenta la edad del usuario a la hora de realizar la recomendación, ya que es un dato que desconoce habitualmente. En cuanto a los rangos de conocimiento, se considera que se tienen conocimientos sobre la herramienta siempre y cuando se sepa usar la funcionalidad de la misma al completo (bots incluidos que suele ser un aspecto aun poco usado). Se ha descartado la opción de una persona mayor a 65 años que sepa usar Telegram debido a que poca gente que con esa edad usa como cliente de mensajería social algo distinto a *Whatsapp*. La búsqueda de participantes se ha realizado en círculos internos siendo siempre personas ajenas al desarrollo. El número de participantes fue sólo de trece personas por las limitaciones de recursos y de tiempo, pero incluye representación más o menos numerosa de los distintos grupos establecidos. La gráfica de participación obtenida por grupos es la siguiente:

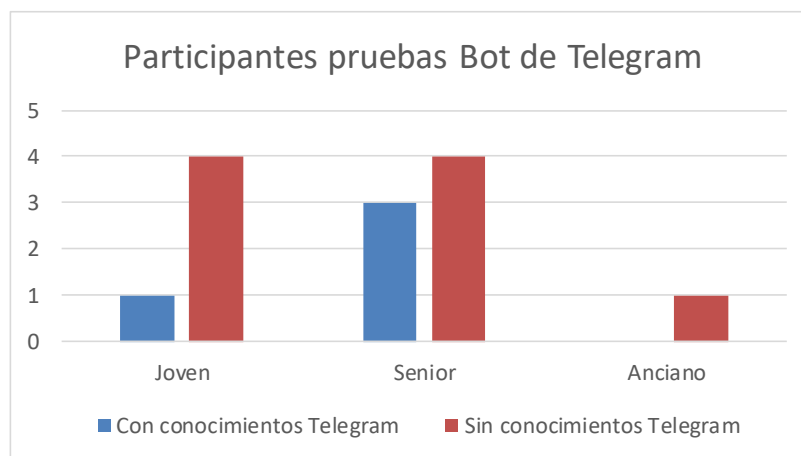


Tabla 13 Gráfico de participantes en las pruebas del bot

Diseño de pruebas

La prueba a realizar para valorar la interacción y el uso del bot desarrollado como prueba de concepto de este proyecto, será una prueba de uso semi-guiado con un cuestionario tipo test final. En ella al usuario se le dará una serie de acciones a realizar, mediante las cuales el usuario irá descubriendo las distintas funcionalidades y opciones que proporciona nuestro sistema. Previamente a realizar la encuesta el usuario dispondrá de tiempo para hacer las pruebas que considere necesarias y ampliar la experiencia con el bot a partir de su uso libre. Esto es necesario

no solo por el hecho de otorgarle al usuario libertad sino porque hay elementos del bot que necesitan tiempo de adaptación al usuario. Entre las acciones a realizar se pueden diferenciar los siguientes tipos de acciones que se sucederán en el orden mostrado a continuación:

Acciones instalación

Objetivo: guiar al usuario en la instalación y configuración de su entorno para poder hacer uso de la funcionalidad del bot.

Grupo de usuario: Joven inexperto, Senior inexperto y Anciano.

Duración: 5-10 minutos

Tareas:

- *Instalar Telegram:* Instalar la aplicación de Telegram en el entorno del usuario en el caso de no tenerla instalada.
- *Inicializar bot:* Buscar el bot desarrollado entre los bot de Telegram existentes e inicializarlo para ese usuario.
- *Añadir a conversación:* Se enseñará al usuario la manera de añadir un bot a una conversación grupal.

Acciones con comandos

Objetivo: Introducción al usuario al uso de comandos, explorando sus posibilidades y utilidades.

Grupo de usuario: Todos los grupos

Duración: 10 - 15 minutos

Tareas:

- *Introducción de comandos:* Introducción y presentación de uso de los comandos de un bot en Telegram.
- *Uso Comando help:* Uso y utilidad del comando **help** en el bot desarrollado.
- *Uso Comando events:* Uso y utilidad del comando **events** para los distintos atributos permitidos en el bot desarrollado.

Acciones con botones

Objetivo: Introducción al usuario al uso de los botones desarrollados y que forman parte de la interfaz de interacción del bot. Para realizar este bloque de acciones es necesario haber realizado previamente las acciones con comandos.

Grupo de usuario: Todos los grupos

Duración: 3 - 5 minutos

Tareas:

- *Introducción a los botones:* Introducción y presentación de uso de botones de un bot en Telegram.
- *Uso de los botones more/less:* Uso y utilidad de los botones more / less que aparecen en la interfaz al ejecutar el comando events.
- *Uso del botón de valoración:* Uso y utilidad del botón de valoración que aparece en la interfaz al ejecutar el comando events.

Acciones con inline query

Objetivo: Introducción al usuario al uso de comandos, explorando sus posibilidades y utilidades.

Grupo de usuario: Todos los grupos

Duración: 5 - 10 minutos

Tareas:

- Introducción al inline query: Introducción y presentación de uso de los inline query de un bot en Telegram.
- Uso del inline query: Uso y utilidad del inline query del bot desarrollado.

Acciones de recomendación

Objetivo: Introducción al usuario al uso de comandos de recomendación, explorando sus posibilidades y utilidades.

Grupo de usuario: Todos los grupos

Duración: 5 - 10 minutos

Tareas:

- Uso del comando recomend: Uso y utilidad del comando **recomend** del bot desarrollado.
- Uso del comando myrecomendationinfo: Uso y utilidad del comando **myrecomendationinfo** del bot desarrollado.

A partir de estos grupos de acciones, se le proporcionará al usuario una visión global de todas las posibilidades, permitiéndole un grado avanzado de uso y la capacidad de valorar la aplicación. Es preciso recalcar de nuevo que tras estas acciones se invitará al usuario a usar el bot a su libre albedrío a fin de que temas como por ejemplo la recomendación se enriquezcan del usuario y puedan desplegar más potencial.

La guía de acciones desarrollada se encuentra en el Anexo 9, [Prueba de uso guiado bot Telegram](#) y la encuesta en el Anexo 10, [Encuesta pruebas bot Telegram](#)

Resultados obtenidos

Los resultados completos obtenidos de las encuestas de valoración realizadas se encuentran en el Anexo 11, [Resultados de la encuesta de pruebas bot Telegram](#) de los cuales se han querido recalcar los siguientes:

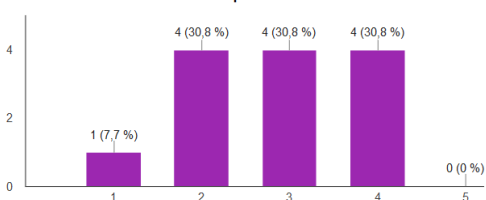
En cuanto a la temática y la búsqueda de una prueba de concepto que tuviera una utilidad real más allá de refutar la investigación, parece que se acertó. Como refleja la gráfica la mayoría de los participantes quedaron agradados en estos dos aspectos.



Ilustración 46 Resultados encuesta interesante vs utilidad

Respecto a la facilidad de aprendizaje reveló aspectos interesantes ya que se nota disparidad de opiniones que cruzan además con la pregunta de si encontraban mayor facilidad a la hora de buscar eventos en comparación con un medio tradicional como podía ser un ordenador. En este caso la lectura que se puede hacer de las cuestiones es que en cuanto a la facilidad de aprendizaje está muy presente el grado de uso de los usuarios con Telegram y frente a la búsqueda sobre medios tradicionales puede ocurrir en muchos casos que los usuarios no tengan muy claro dónde encontrar esos eventos, por lo cual han podido responder teniendo en cuenta ese factor más allá de la interacción o el uso del bot les pueda parecer más sencillo. Luego es cierto que a la pregunta de si requería poco esfuerzo la mayoría de los encuestados lo afirmó, pues realmente en cuatro interacciones recibes mucha información, pero también remarcaron que sin una ayuda inicial les habría sido complicado comprender el funcionamiento del bot.

Me resulto más sencillo que buscar en el ordenador.



Me resulto fácil de aprender a utilizarlo.

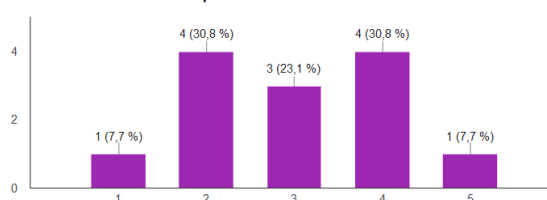
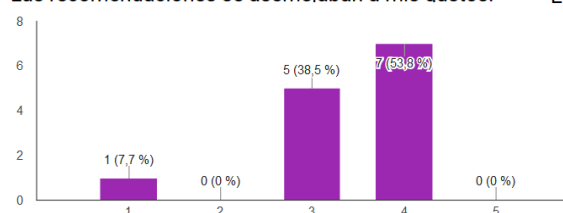


Ilustración 47 Resultados encuesta búsqueda ordenador vs facilidad de aprendizaje

En cuanto a la recomendación de gustos parece que la gente estuvo conforme y los que tampoco estuvieron tan de acuerdo, por lo visto en los comentarios, se entiende que les faltó tiempo de uso y ajuste de la recomendación para afinar su precisión. Se recomendó a los participantes el darle uso libre antes de realizar la encuesta, pero es probable que en muchos casos el uso no haya sido suficiente.

Las recomendaciones se asemejaban a mis gustos.



El bot supo extraer mis gustos correctamente.

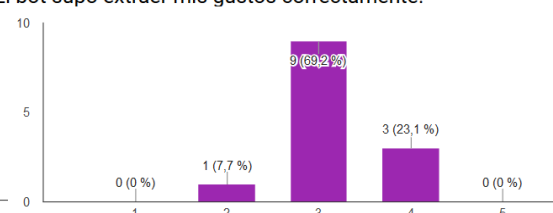
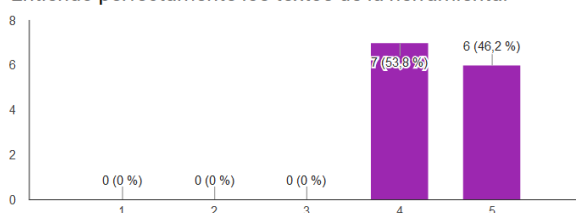


Ilustración 48 Resultados encuesta recomendación de gustos

La información de texto del bot tuvo una buena valoración en cuanto a comprensión en la encuesta, pero se vieron varios comentarios haciendo referencia a que necesitaba más claridad. El usuario habitual de aplicaciones móviles es un usuario acostumbrado a lo visual y no a representaciones puras en texto, lo cual es uno de los motivos chocantes, aun así, si la parte comprensiva es aceptable queda buscar mejoras en cuanto a la estructuración visual de la misma.

Entiendo perfectamente los textos de la herramienta.



La información presentada era fácil de entender

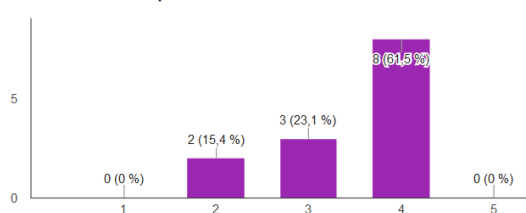
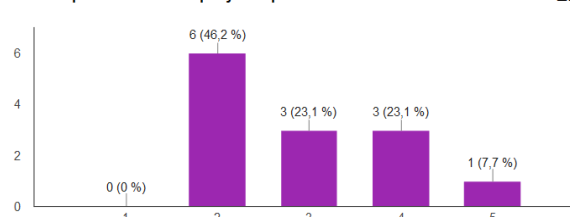


Ilustración 49 Resultados encuesta referentes a textos del bot

Por último y como reflejan las siguientes gráficas la prueba pareció algo compleja, aunque clara en cuanto a instrucciones se refiere. Es posible que los usuarios participantes encontraran algo pesada la prueba en cuanto al tiempo de dedicación, pero a fin de cuentas era un proceso necesario para dar a conocer toda la funcionalidad y se pudieran probar todos los aspectos de la aplicación.

Me ha parecido compleja la prueba.



Entendí las instrucciones de la prueba.

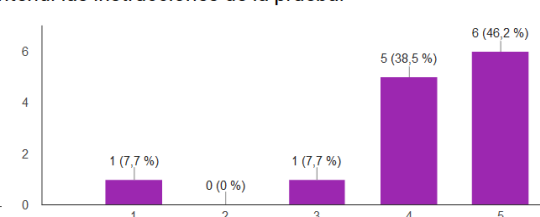


Ilustración 50 Resultados encuesta complejidad de la prueba

Conclusiones

En las pruebas realizadas se han observado varios detalles a tener en cuenta a la hora de mejorar el bot, tanto en cuanto a la interacción como a la hora de usar la plataforma de Telegram. Estos detalles a mejorar serían:

- **Interacción con Telegram**
La interacción desarrollada no acaba de ser del todo cómoda, imbuyendo al usuario en una metodología de hacer uso de comandos, más propios de entornos informáticos. Este hecho puede hacer que, a parte de los usuarios no tan técnico, informáticamente hablando le pueda parecer extraña la forma de funcionar de los bots. La interacción no está desacoplada de Telegram, en el sentido de que sigue muy al pie de la letra sus metodologías, siendo imposible realizar un cambio en la interacción. Cualquier cambio de interacción que se pretenda llevar a cabo va a depender completamente del sistema Telegram y de las posibilidades soportadas por su entorno y Api, por lo tanto, la innovación en este ámbito no depende de la implementación de bots en sí.
- **Comando events**
En relación con lo antes explicado de la interacción, se ve que a varios de los participantes no les quedó claro el uso de argumentos y en la ayuda no venían definidas todas las opciones posibles.
- **Comando de ayuda**
Sería necesario hacer el comando de ayuda más extenso o incluso interactivo, permitiendo de esta manera dar información más específica y detallada al usuario.
- **Teclados de valoración**
La disposición del teclado en vertical en vez de horizontal queda visualmente pobre y en terminales de pantalla reducida obliga a realizar desplazamiento.
- **Formato de salida de los eventos**
En cuanto a la representación de la información de los eventos, no está mal, aunque se podrían añadir mejoras como, por ejemplo: emoticonos, imágenes de las imágenes, o representación gráfica de las localizaciones del evento.
- **Añadir más contenidos**
Los contenidos actuales muestran una cantidad considerable de eventos, pero sería necesario aumentar este número, así como la cantidad de regiones publicadas, pues en ocasiones al realizar búsquedas sobre regiones menos populares, no se arrojan resultados.
- **Orden de los eventos**
En ocasiones al mostrar la lista de eventos tras una petición el orden de aparición prima a mostrar los más antiguos primeros, es decir si hay un musical todo un año y un concierto solo un día, parece que se le da prioridad al musical frente al concierto. Sería necesario retocar
- **Posibilidad de editar datos de recomendación**
En el caso de que un usuario quiera reestablecer y borrar su información almacenada de usuario en cuanto a valoraciones o mensajes almacenados para analizar, se debería habilitar un comando o forma de que el usuario por sí solo pudiese gestionar esta tarea. También podría estudiarse el dotar al usuario de posibilidad de definir gustos más allá de las valoraciones efectuadas.
- **Posibilidad de seleccionar idioma**
Este punto no recibió queja por parte de los participantes, pero es un punto importante en cuanto a la expansión del bot. En cuanto a la parte técnica este punto no es problema ya

que el Api de eventos proporciona los eventos en distintos idiomas, por lo que se debería poder elegir el idioma de estos, mejorando así la experiencia de usuario.

Aun con estos detalles a mejorar, la prueba de concepto del bot de Telegram reúne las funcionalidades esperadas y se considera una buena aproximación con pequeños detalles a limar en temas de interacción. Se debe suponer que en algunos casos se ha juntado la curva de aprendizaje de los bots de Telegram con la de la propia implementación del bot, requiriendo un esfuerzo adicional por parte de los usuarios.

6. Conclusiones

6.1. Conclusiones

En este trabajo se ha propuesto una solución para aprovechar e integrar el uso de imágenes sociales como medio para extraer atributos o características de sus propietarios o quienes las divulgan y crear una prueba de concepto multimodal que use esta aplicación junto con clasificadores de textos. La búsqueda del tema idóneo para este trabajo fue finalmente los eventos sociales, debido a la gran cantidad de información, la no existencia de trabajos realizados en esta temática y la perfecta adaptación a las necesidades del proyecto.

Para trabajar con esta temática se estableció una taxonomía para eventos sociales que fue refutada con éxito a la hora de extraer información, sin echar en falta en ningún momento algún añadido más y confirmando exitosamente **H5**. Una vez teniendo la taxonomía establecida se partió de una extracción masiva de imágenes sociales obtenidas de una fuente online con etiquetas propuestas por el usuario que las definían, para realizar clusterizaciones cromáticas, asignando una serie de colores a cada elemento dentro de la taxonomía y confirmando **H1**. A continuación y a fin de realizar una solución multimodal, se realizó un procedimiento similar con textos, y sin que esto fuese una gran novedad se confirmó **H2**. Finalmente, al aunar las dos partes en la prueba de concepto realizada se consiguió aprovechar las dos tecnologías (**H3**) de forma multimodal para etiquetar las interacciones de los usuarios dentro de una aplicación para soporte de mensajería social, confirmando así **H4**.

La implementación realizada aporta un grado alto de ingeniería al proyecto, juntando de forma muy modular y bien diferenciada, distintas áreas de la informática como es el Web Scraping, Machine Learning y Servicio Api Rest. El uso de Telegram como plataforma fue agradable y sencillo, evitando tener que realizar un mayor desarrollo e inventar una plataforma entera.

A través de la prueba de concepto realizada se observó que con porcentajes de precisión superiores al **75%** tanto en el clasificador de texto como en el clasificador de imágenes, es posible realizar buenas recomendaciones a usuarios realizando una predicción de gustos, usando sus imágenes y sus mensajes de texto como fuente.

6.2. Trabajos futuros

Este trabajo ha dejado atrás opciones y variantes a lo largo de su realización. El tomar decisiones sobre su devenir muchas veces cerraba puertas y abría otras, pero no por ello se deben tomar como malas. Un aspecto que ha dado mucha reflexión y trabajo de decisión es sobre la temática sobre la que realizar la prueba de concepto y aplicar las ideas de la investigación. A continuación, se enumerarán los temas aparcados y los motivos que en trabajos futuros sería muy interesante recuperarlos:

Sentimientos

Uno de los temas estrella de este proyecto en su comienzo era el de los sentimientos. El poder etiquetar sentimientos a partir de colores puede ser un trabajo fascinante que puede enfocarse desde muchas perspectivas e incluso rebatir dogmas sociales establecidos en temas cromáticos. La mayor dificultad encontrada respecto a esta temática fue el problema de encontrar imágenes clasificadas o con etiquetas relevantes a sentimientos. Por otro lado, también se puso en duda si esto era posible, ya que los sentimientos, aunque puedan ser colectivos en algunos casos, por lo general tienden a la subjetividad.

Ropa y tribus urbanas

El caso de la ropa también parecía interesante y en la sociedad actual existiendo relaciones entre estilos de ropa y tribus urbanas, parecía razonable poder unir todo en una red permitiendo

etiquetar a personas por el colorido de su vestimenta. El problema fue una vez más la dificultad a la hora de extraer imágenes, debido a la falta de etiquetado relevante y dificultad de ser objetivo. El hacer uso de imágenes no sociales por ejemplo pases de modelos o fotos de catálogo, no ayudaba a la hora de comparar con imágenes del día a día. La dispersión de colores dentro de las variantes de cada estilo hacia que tampoco quedarán bien definidos los colores propios de cada clase.

Filtros de Instagram

El estudio de los filtros de Instagram relacionado con sentimientos, es otro trabajo futuro de interés que se debería llevar a cabo. Hoy en día miles de personas sobre todo jóvenes de todo el mundo hacen uso de Instagram como diario fotográfico. Hasta ahora y según los trabajos revisados, el análisis de información únicamente ha considerado el texto y poco más, siendo las imágenes lo más representativo de esta aplicación para redes sociales. Al inicio de este trabajo se formuló la teoría de que parecía que la gente hacía uso de unos determinados filtros según su estado emocional y mediante el análisis de estos, se podría extraer los sentimientos del propietario de la imagen. Al final debido a lo engorroso de la extracción de imágenes de Instagram y el trabajo de revisión que necesitaba al igual que pasaba en el tema de los sentimientos, se dejó el tema para el futuro.

Al margen de las temáticas que se han explorado en este proyecto como trabajos futuros dentro de lo ya realizado, habría bastante mejora técnica o de optimización que realizar. Se podría también hacer más complejo el sistema de recomendación o introducir nuevas técnicas a fin de hacerlo mejor, más precisa y que requiriera menos tiempo de ajuste. Sería también deseable ampliar los servicios del bot añadiendo más opciones y retocando todo mencionado como mejoras en las encuestas realizadas. Además, sería interesante que la ampliación del bot, que ya analiza imagen y texto también los haga con video y audio, para conseguir un bot multimodal completo que explotara todas las herramientas y medios proporcionados por Telegram. El uso de Telegram por otra parte podría ser otra de las mejoras a realizar, cambiando el bot de plataforma o adaptándolo a otras nuevas.

Referencias

- [1] P. Bhattacharya, S. Ghosh, J. Kulshrestha, N. Ganguly y M. B. Zafar, «Exploring Topical Groups in Microblogs at Scale,» *CSCW*, 2014.
- [2] R. A. Hanneman, y M. Riddle, *Introduction to social network methods.*, 2005.
- [3] J. B. Walther, «Anticipated ongoing interaction versus channel effects on relational communication in computer-mediated interaction,» de *Human communication research*, 20(4), 1994, pp. 473-501.
- [4] S. LaValle, E. Lesser, R. Shockley y M. Hopkins, «Big data, analytics and the path from insights to value,» *MIT sloan management review*, vol. 52, nº 2, 2011.
- [5] . M. Varvello y G. M. Voelker, «Second life: a social network of humans and bots.,» *Proceedings of the 20th international workshop on Network and operating systems support for digital audio and video ACM.*, pp. 9-14, 2010, June.
- [6] H. Müller, A. G. S. de Herrera, J. Kalpathy-Cramer, D. Demner-Fushman, S. Antani y I. Egel, «Overview of the ImageCLEF 2012 Medical Image Retrieval and Classification Tasks. In CLEF,» pp. 1-16, 2012.
- [7] S. Arivazhagan, R. N. Shebiah, S. S. Nidhyandhan y L. Ganesan, «Fruit recognition using color and texture features,» *Journal of Emerging Trends in Computing and Information Sciences*, , vol. 1, nº 2, pp. 90-94., 2010.
- [8] P. Haffner y V. Vapnik, «Support vector machines for histogram-based image classification,» *IEEE (Volume:10, Issue:5)*, pp. 1055-1064, 1999.
- [9] S. Sergyan, «Color histogram features based image classification in content,» *Applied Machine Intelligence and Informatics*, pp. 221-228, 2008.
- [10] R. Z. Greg Pass, «Histogram Refinement for Content-based image retrieval,» *Applications of Computer Vision WACV '96.* , pp. 92-102, 1996.
- [11] A. H. Jana Machajdik, «Affective image classification using features inspired by psicology,» *MM '10 Proceedings of the international conference on Multimedia*, pp. 83-92, 2010.
- [12] S. Siersdorfer, E. Minack, F. Deng y J. & Hare, «Analyzing and predicting sentiment of images on the social web,» *ACM*, pp. 715-718, 2010.
- [13] J. Scott, *Social network analysis*, Sage, 2012..
- [14] C. Fellbaum, «WordNet web,» Princeton University, 2005. [En línea]. Available: <http://wordnet.princeton.edu/>. [Último acceso: 14 5 2016].
- [15] D. Quercia, N. Lathia, F. Calabrese, G. Di Lorenzo y J. Crowcroft, «Recommending social events from mobile phone location data,» *IEEE International Conference on Data Mining*, pp. 971-976, 2010.
- [16] A. Noulas, S. Scellato, N. Lathia y C. Mascolo, «A random walk around the city: New venue recommendation in location-based social networks.,» *IEEE International Conference on and 2012 International Confernece on Social Computing*, pp. 144-153, 2012.
- [17] H. Khrouf, V. Milicic y R. & Troncy, «Mining events connections on the social web: Real-time instance matching and data analysis in EventMedia.,» *Web Semantics: Science, Services and Agents on the World Wide Web*, nº 24, pp. 3-10, 2014.
- [18] «Eventful,» [En línea]. Available: <http://www.eventful.com>. [Último acceso: 14 06 2016].
- [19] «Ticketea,» [En línea]. Available: <http://www.ticketea.com>. [Último acceso: 15 6 2016].
- [20] «Guia del ocio,» [En línea]. Available: <http://www.guiadelocio.com>. [Último acceso: 14 6 2016].
- [21] A. Agarwal y O. Rambow, «Automatic detection and classification of social events.,» de *In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics.*, 2010.

- [22] T. Reuter y P. Cimiano, «Event-based classification of social media streams.» . In *Proceedings of the 2nd ACM International Conference on Multimedia Retrieval*. ACM., vol. 22, 2012.
- [23] D. Lee, S. E. Park, M. Kahng, S. Lee y S. G. Lee, «Exploiting contextual information from event logs for personalized recommendation,» *Computer and Information Science* Springer Berlin Heidelberg., pp. 121-139, 2010.
- [24] N. company, «Nike+,» [En línea]. Available: https://www.nike.com/us/en_us/p/activity.
- [25] G. Bell, «Mylifebits,» Noviembre 2001. [En línea]. Available: <https://www.microsoft.com/en-us/research/project/mylifebits/>. [Último acceso: 16 5 2016].
- [26] K. Niemann, M. Wolpers, G. Stoitsis, G. Chinis y N. Manouselis, «Aggregating social and usage datasets for learning analytics: data-oriented challenges,» de *Proceedings of the Third International Conference on Learning Analytics and Know*, 2013.
- [27] J. Y. Jiang y T. L. C., «Analyzing Social Event Participants for a Single Organizer,» In *Tenth International AAAI Conference on Web and Social Media.*, 2016.
- [28] D. Calvo Martin-Corral, «health.io,» 1 2 2016. [En línea]. [Último acceso: 17 7 2016].
- [29] D. R. M. & Co, «<https://x.ai/>,» [En línea]. [Último acceso: 17 7 2016].
- [30] «Link de referencia de vasyaeventbot bot,» [En línea]. Available: <https://storebot.me/bot/vasyaeventbot>. [Último acceso: 6 7 2016].
- [31] «link de referencia del bot geteventsbot,» [En línea]. Available: <https://telegram.me/geteventsbot>. [Último acceso: 5 7 2016].
- [32] «Link de referencia de maxappbot bot,» [En línea]. Available: <https://storebot.me/bot/maxappbot>. [Último acceso: 6 7 2016].
- [33] «Link reference of goalbot bot,» [En línea]. Available: <https://storebot.me/bot/goalbot>. [Último acceso: 20 7 2016].
- [34] INVAT-TUR, «BIG DATA: Retos y oportunidades para el turismo,» 2016.
- [35] C. Shan, S. Gong y P. W. McOwan, «Facial expression recognition based on local binary patterns: A comprehensive study.,» *Image and Vision Computing*, pp. 803-816, 2009.
- [36] M. A. Turk y A. P. Pentland, «Face recognition using eigenfaces,» *Computer Vision and Pattern Recognition CVPR'91 IEEE* , pp. 586-591, 1991.
- [37] R. P. Morris, «Role-based access to image metadata». Washington, DC: U.S Patente U.S. Patent No. 6,871,231., 2005.
- [38] M. Balabanović y Y. Shoham, «Fab: content-based, collaborative recommendation,» *Communications of the ACM*, vol. 40, n° 3, pp. 66-72., 1997.
- [39] M. H. bin Ismail, «ELISA: E-Learning Integrated Short Announcement.,» *Colloquium in Computer and Mathematical Sciences Education (CCMSE 2015)*, p. 110, 2015.
- [40] V. Kecman, «Learning and soft computing: support vector machines, neural networks, and fuzzy logic models,» de *Learning and soft computing: support vector machines, neural networks, and fuzzy logic models*, MIT press, 2001.
- [41] S. C. Herring, «Dynamic topic analysis of synchronous chat,» In *New Research for New Media: Innovative Research Methodologies Symposium Working Papers and Readings.*, 2003.
- [42] M. L. Hinman , R. Gheorghe y R. Russo, «ElasticSearch in Action,» de *ElasticSearch in Action*, Manning.
- [43] Google, «Google Api Places,» [En línea]. Available: <https://developers.google.com/places/web-service>. [Último acceso: 2 4 2016].
- [44] Microsoft, «Bing Api places,» [En línea]. Available: <https://msdn.microsoft.com/en-us/library/ff701715.aspx>.
- [45] «Geonames,» [En línea]. Available: <http://www.geonames.org/>. [Último acceso: 14 6 2016].

- [46] «Gadm,» [En línea]. Available: <http://www.gadm.org/>. [Último acceso: 14 6 2016].
- [47] «Koordinates,» [En línea]. Available: <https://about.koordinates.com/>. [Último acceso: 14 6 2016].
- [48] «Qgis Website,» [En línea]. Available: <http://www.qgis.org/es/site/>. [Último acceso: 15 5 2016].
- [49] B. Van Merriënboer, D. Bahdanau, V. Dumoulin, D. Serdyuk, D. Warde-Farley, J. Chorowski y Y. Bengio, «Blocks and fuel: Frameworks for deep learning,» *arXiv preprint*, 2015.
- [50] «The Hitchhiker's Guide to Python,» [En línea]. Available: <http://docs.python-guide.org/en/latest/writing/tests/>. [Último acceso: 14 5 2016].
- [51] A. Fujisawa, K. Matsumoto, M. Yoshida y K. Kita, «An Illustration Image Classification Focusing on Infrequent Colors,» *International Journal of Advanced Intelligence*, 2016.
- [52] Š. R. y B. I., «Image Classification Based on Hierarchical Temporal Memory and color features,» Institute of measurement science, Bratislava, Slovakia, 2015.
- [53] H. Ma, «On Measuring Social Friend Interest Similarities in Recommender Systems,» *Microsoft Research WA 98052*, 2014.

Anexos

1. Juego de Facebook experimento de filtros Instagram

Se realizó paralelamente al proyecto, un pequeño juego experimental para Facebook que consistía en recabar información del usuario, sobre el uso que hacía de los filtros de Instagram para posteriormente compararlo.

Dinámica del juego

La dinámica del juego era sencilla, se presentaba una foto y en la parte inferior miniaturas de los distintos filtros de Instagram. El usuario debía elegir que filtro le gustaba más para cada foto y pasar a la siguiente.

Experimento

El experimento consistía en establecer relaciones entre tipos de filtro y temáticas de imágenes, por ejemplo, en las fotos de futbol siempre se elegía el filtro A y en las de baloncesto el B.

Desarrollo

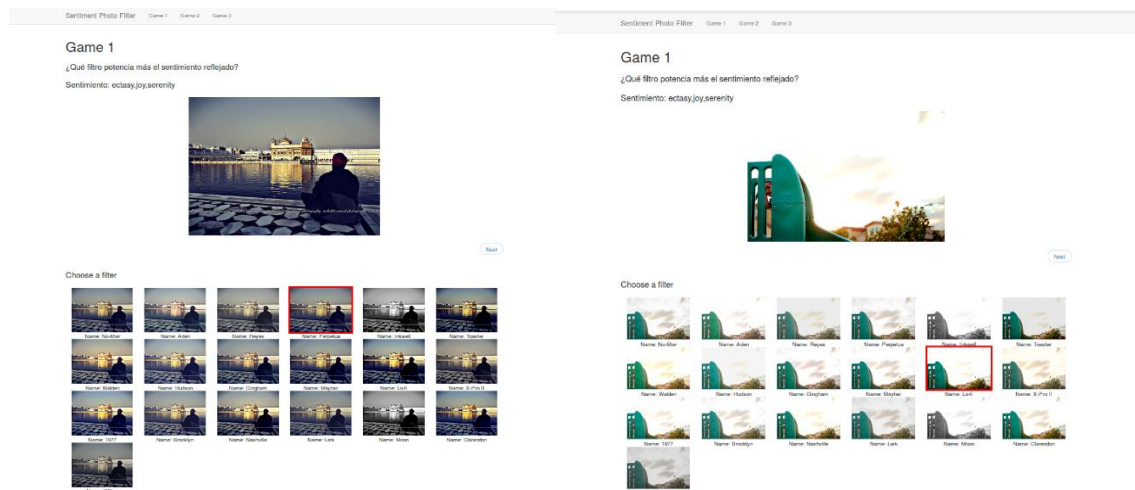
El sistema está dividido en dos partes, un api que sirve las imágenes a mostrar en el juego con la temática a la que pertenecen, y guarda las respuestas emitidas por los usuarios. Y por otro lado el juego en sí, que muestra las imágenes al usuario y realiza la interacción.

Despliegue

El juego estaba alojado en un servidor de Amazon y referenciado a Facebook a través de un webhook. El desarrollo consistía simplemente en un sitio web que embebía el juego realizado Angularjs dentro de la página y a su vez dentro de Facebook. Las respuestas y las imágenes del juego se guardaban en una base de datos de *Mongodb*.

Conclusión

No se llegó a realizar el experimento, a pesar de haberlo preparado, porque se abordaron temas de mayor prioridad y por otro lado estaba la dificultad de encontrar participantes.



2. Análisis de filtros Instagram

A continuación del experimento del anexo 1, se realizó un estudio sobre imágenes de Instagram para ver si era posible establecer la relación entre una temática y un filtro. Para ello se descargaron dos sets de imágenes diferentes, de 8K imágenes cada uno y sobre la búsqueda de *basketball*, *concert*, *soccer* y *theater* dentro de Instagram.

Filtro	basketball	concert	soccer	theater	basketball	concert	soccer	theater	TOTAL %	TOTAL RESULT	COMP	RESULT	Filtro	basketball	concert	soccer	theater	basketball	concert	soccer	theater	TOTAL %	TOTAL
Rise	15	8	7	18	31.25%	16.67%	14.58%	37.50%	48	0.60%	theater	theater	Rise	13	11	4	23	25.49%	21.57%	7.84%	7.84%	51	0.63%
Willow	2	13	2	10	7.43%	48.15%	7.43%	37.04%	7	0.34%	concert	theater	Willow	10	12	2	20	22.73%	27.27%	4.55%	45.45%	44	0.55%
Unleash	1	3	1	2	14.29%	42.86%	14.29%	28.57%	7	0.09%	concert	??	Unleash	4	4	1	4	30.77%	30.77%	7.69%	30.77%	13	0.16%
Mayfair	20	13	16	22	28.17%	18.31%	22.54%	30.99%	71	0.88%	theater	theater	Mayfair	24	20	8	30	29.27%	24.39%	9.76%	36.59%	82	1.02%
Hudson	11	7	5	15	29.9%	19.42%	13.04%	39.47%	38	0.47%	theater	??	Hudson	16	9	3	16	36.36%	20.45%	6.82%	36.36%	44	0.55%
Moon	15	30	4	20	20.00%	40.00%	5.33%	34.67%	75	0.93%	concert	concert	Moon	14	21	2	19	25.00%	37.50%	3.57%	35.71%	56	0.69%
Lark	53	34	13	48	35.61%	22.97%	8.78%	32.43%	148	1.84%	basketball	basketball	Lark	52	32	28	49	32.30%	19.88%	17.39%	30.43%	161	2.00%
Hale	21	17	15	22	29.00%	22.97%	20.00%	26.33%	75	0.93%	theater	theater	Hale	16	18	9	20	27.66%	27.66%	13.65%	30.73%	65	0.81%
Braman	1	3	0	3	14.29%	42.86%	0.00%	42.86%	7	0.09%	??	basketball	Braman	4	3	2	2	36.36%	27.27%	18.18%	18.18%	11	0.14%
Earlybird	6	4	0	5	40.00%	26.67%	0.00%	33.33%	15	0.19%	basketball	theater	Earlybird	2	3	2	6	15.38%	23.08%	15.38%	46.15%	13	0.16%
Cretia	12	20	6	15	22.4%	37.74%	11.32%	28.50%	53	0.66%	concert	theater	Cretia	16	26	5	27	21.62%	35.14%	6.76%	36.40%	74	0.92%
Sierra	5	16	7	10	11.36%	36.36%	15.51%	36.36%	44	0.55%	??	basketball	Sierra	15	10	5	14	34.09%	22.73%	11.36%	31.82%	44	0.55%
Riyes	7	9	3	15	29.59%	26.47%	5.62%	44.12%	34	0.42%	theater	theater	Riyes	7	9	5	13	20.58%	26.47%	14.71%	35.24%	34	0.42%
Ludwig	39	36	10	47	29.55%	27.27%	7.58%	35.61%	132	1.64%	theater	theater	Ludwig	25	23	14	31	26.88%	24.73%	15.05%	33.33%	93	1.16%
Siraon	1	2	0	1	20.00%	50.00%	0.00%	25.00%	4	0.01%	concert	theater	Siraon	1	1	1	4	14.29%	14.29%	14.29%	37.14%	7	0.09%
Ashby	1	4	0	5	10.00%	40.00%	0.00%	50.00%	10	0.12%	concert	??	Ashby	2	1	1	2	33.33%	16.67%	16.67%	33.33%	6	0.07%
Junio	49	44	23	50	29.52%	28.51%	13.86%	30.12%	186	2.05%	theater	theater	Junio	34	39	22	59	22.08%	25.32%	14.29%	38.31%	154	1.91%
Normal	1423	1456	1747	1260	24.42%	24.49%	29.78%	21.21%	5066	72.84%	soccer	soccer	Normal	1423	1456	1734	1294	24.08%	24.67%	29.25%	21.90%	5909	73.29%
Kelvin	1	1	0	3	20.00%	20.00%	0.00%	60.00%	5	0.06%	theater	theater	Kelvin	1	1	0	3	20.00%	20.00%	0.00%	60.00%	2	0.02%
Nashville	14	13	4	10	34.38%	31.71%	9.79%	24.39%	41	0.51%	basketball	basketball	Nashville	19	8	4	17	39.58%	16.67%	8.33%	39.42%	48	0.60%
Suzo	3	3	0	4	30.00%	30.00%	0.00%	40.00%	10	0.12%	theater	concert	Suzo	2	3	0	1	33.33%	50.00%	0.00%	16.67%	6	0.07%
Maven	0	0	0	2	0.00%	0.00%	0.00%	100.00%	2	0.02%	theater	theater	Maven	0	0	2	0	0.00%	0.00%	0.00%	100.00%	2	0.02%
Clarendon	142	106	66	157	30.15%	22.51%	14.01%	33.33%	471	5.95%	theater	basketball	Clarendon	135	110	77	125	30.20%	24.61%	17.23%	27.96%	447	5.44%
Perpetua	6	11	4	14	17.4%	31.43%	11.43%	40.00%	25	0.43%	theater	theater	Perpetua	5	10	4	12	16.13%	32.26%	12.90%	36.71%	31	0.38%
Amaro	22	15	8	25	31.43%	21.43%	11.43%	38.71%	70	0.87%	theater	theater	Amaro	19	14	10	31	25.68%	18.62%	15.51%	41.89%	74	0.92%
Teaser	0	0	1	0	0.00%	0.00%	0.00%	100.00%	1	0.01%	theater	??	Teaser	1	0	0	1	50.00%	0.00%	0.00%	50.00%	2	0.02%
Slumber	9	10	5	14	23.68%	26.32%	13.16%	36.84%	38	0.47%	theater	concert	Slumber	7	12	5	9	21.21%	36.36%	15.15%	27.27%	33	0.41%
Dogpatch	0	1	0	4	0.00%	20.00%	0.00%	80.00%	5	0.06%	theater	concert	Dogpatch	1	3	0	1	20.00%	60.00%	0.00%	20.00%	5	0.06%
Stylize	0	2	0	4	0.00%	26.67%	0.00%	66.67%	7	0.09%	theater	??	Stylize	2	1	2	1	28.57%	14.29%	28.57%	28.57%	6	0.07%
Aden	10	12	7	15	22.73%	27.27%	15.91%	34.09%	44	0.55%	theater	theater	Aden	14	13	2	17	30.43%	28.26%	4.35%	36.96%	46	0.57%
Valencia	19	18	14	25	23.7%	22.50%	17.50%	36.25%	80	0.99%	theater	theater	Valencia	16	13	8	38	22.68%	19.37%	11.43%	47.14%	70	0.87%
Gingham	41	40	12	38	31.36%	30.53%	9.16%	29.01%	131	1.63%	basketball	theater	Gingham	39	37	9	43	30.47%	28.91%	7.03%	33.59%	128	1.59%
Popcorn	1	1	0	1	33.33%	33.33%	0.00%	33.33%	3	0.04%	??	??	Popcorn	1	1	0	1	33.33%	33.33%	0.00%	33.33%	2	0.04%
Lo-fi	0	0	2	0	0.00%	0.00%	50.00%	50.00%	4	0.05%	theater	concert	Lo-fi	0	0	2	0	0.00%	0.00%	100.00%	100.00%	2	0.02%
Levi	15	17	6	24	24.19%	27.42%	9.68%	28.71%	62	0.77%	theater	concert	Levi	23	25	4	15	34.33%	37.21%	9.97%	22.29%	87	0.93%
X-Pro II	21	33	6	41	20.79%	32.07%	5.84%	40.59%	101	1.25%	theater	theater	X-Pro II	23	40	12	27	22.50%	39.23%	11.76%	32.71%	102	1.27%
Helena	0	0	0	4	0.00%	0.00%	0.00%	100.00%	4	0.05%	theater	theater	Helena	0	0	0	2	0.00%	0.00%	0.00%	100.00%	2	0.02%
Waalen	0	2	0	3	0.00%	40.00%	0.00%	60.00%	5	0.06%	theater	??	Waalen	2	2	1	2	28.57%	28.57%	14.29%	28.57%	7	0.08%
Giinza	1	1	0	1	33.33%	33.33%	0.00%	33.33%	3	0.04%	??	concert	Giinza	1	3	2	1	14.29%	42.86%	28.57%	14.29%	7	0.09%
Vesper	0	5	0	3	0.00%	62.50%	0.00%	37.50%	8	0.10%	concert	theater	Vesper	0	1	0	2	0.00%	33.33%	0.00%	66.67%	3	0.04%
Brooklyn	0	0	0	1	0.00%	0.00%	0.00%	100.00%	1	0.01%	theater	theater	Brooklyn	1	0	1	1	33.33%	0.00%	33.33%	33.33%	3	0.04%
TOTAL	8053	8005	1000%										8053	8005	1000%								

Al comparar los filtros usados para cada una de las temáticas en los dos sets de pruebas descargados de Instagram, se puede ver representado en la imagen en color verde cuando coincidían, en rojo cuando no coincidían y en amarillo cuando faltaban imágenes para tener un resultado claro en alguno de los sets o en los dos.

El estudio reveló resultados interesantes y que dan esperanza a que por esta vía sea posible relacionar filtros con temáticas dando buenos resultados. Se debería intentar repetir el experimento usando un mayor número de imágenes y de temáticas. Este experimento solo tiene sentido en el caso de que las temáticas elegidas tengan relación entre sí, es decir, en este caso todas las temáticas usadas se podrían englobar dentro de actividades, en cambio sí se usan palabras de temáticas que no guardan relación como por ejemplo *basketball*, *sunny* y *apple* seguramente los resultados serían malos o sin sentido.

3. Listado de páginas soportadas por el recuperador de eventos

48horasbadajoz	entradascom_api	lvbp	teatrosiglodeoro
800cl	esmadrid	madein	teleticket
abudhabievents	espanaesultura	manual	theguideistanbul
agendagranada	espn	menorca	thevisitorpanama
agendahoy	estacerca	meteoalarm	ticketac
agendaideal	estoy	mexicoescultura	ticketcorner
agendalx	euromuse	miamiandbeaches	ticketea
agendavenezia	euskadikulturklik	miamibeach	ticketea_api
algeria	eventbrite	miamiherald	ticketmasterBE
allconferences	eventful	mistorneosdegolf	ticketone
almeriacultura	eventimDE	motogp	ticketpluspanama
alternativateatral	eventseye	motorsportarena	ticketsfc
andalucia	expodatabase	muenchen	timeanddate
aokdk	facebook	mundotoro	timeoutdubai
artindusseldorf	farodevigo	museosespana	timeoutistambul
artsmallorca	feriade	musicopera	timeoutmexico
atp	feriasinfo	mutualart	timeoutsapaulo
babelfan	foodieandtours	navarracultural	toroticket
badajozdirecto	frankfurtrhein	nferias	transfermarkt
bandsintown	frankfurttourismus	nyc	trip4real_api
bankholidays	genevetourisme	nytimes	tsnn
baseballdecuba	googlecalendar	ocioon	tuboleta
bookmyshow	googleplaces	onqatar	tuentrada
buenosairesgob	gruposerra	pamplona	turgalicia
bugbog	guiadelocio	panatickets	turimedelleida
bvents	guiauol_brasil	pitalmeria	turismoasturias
cairo360	guidebe	portalferias	turismodecantabria
canarias7	haroturismo	portalfiestas	verybello
carreraspopulares	hostedstatsbasket	prensaiberica	viagogo
catracalivre	hostedstatssoccer	primerafila	viennaconcerts
ciclismoenmallorca	hostedstatstennis	qpp	viralagenda
classictic	huelgalendario	quehacerensantiago	visitberlin
clubbingSpain	iamsterdam	redbull	visitbrussels
concertboom	ibizaspotlight	rfegolf	visitdenmark
conferencealerts	ilovetoledo	rfev	visitdubai
conocecuba	importio	runedia	visitlondon
cuandopasa	infocongressi	running4runners	visitporto
culturagal	inyourpocket	salamancalia	visitstockholm
cyclingnews	ireland	santiagocultura	viveleon
destinia	istambul	santiagoturismo	vocento
diariodenavarra	italiatoday	servitoro	voilafr
donostiakultura	juntadeandalucia	soccerway	vuenosairez
dotoday	kedin	songkick	warszawa
dss2016	kherdja	songkick_api	weekly
dubaicalendar	kultunaut	sortiraparis	westticket
edreams	kulturalna	spaininfo	wheremilan
educationukholidays	laciudadcadizes	suenacubano	wikido
elgirhoy	lanyrd	sympla	zuerich
enlima	livenation	taquillacom	
entradascom	londontown	tastingeurope	

4. Propuesta para el filtrado de pornografía en Instagram

Se va a realizar una propuesta sobre una metodología para filtrar imágenes pornográficas de Instagram con el fin de realizar extracciones de imágenes más limpias de esta aplicación para red social.

Problema existente

Actualmente en Instagram se cuelan muchas imágenes pornográficas, ya sea porque publican antes de pasar por sus protocolos de revisión o simplemente que el predictor que usan da falsos negativos. Se ha buscado documentación sobre el proceso que siguen y no se ha encontrado documentación al respecto.

Análisis Imágenes

Las imágenes pornográficas encontradas, en su mayoría eran publicidad o promoción para que los usuarios de Instagram visitaran una página web. También se encontró mucha pornografía que contenía códigos de referencia a perfiles de la red social Snapchat. Todas estas imágenes solían seguir un mismo patrón de colores o una temática parecida, llegando a tener difuminados logos o etiquetas sobre la imagen. En principio, el hecho de llevar imágenes fundidas o difuminadas es lo que otorga dificultad de detección, pero aun así creo posible la detección de estos patrones sin cometer demasiados falsos positivos.

Solución Propuesta

Se proponen varias medidas a implantar a fin de detectar este tipo de imágenes:

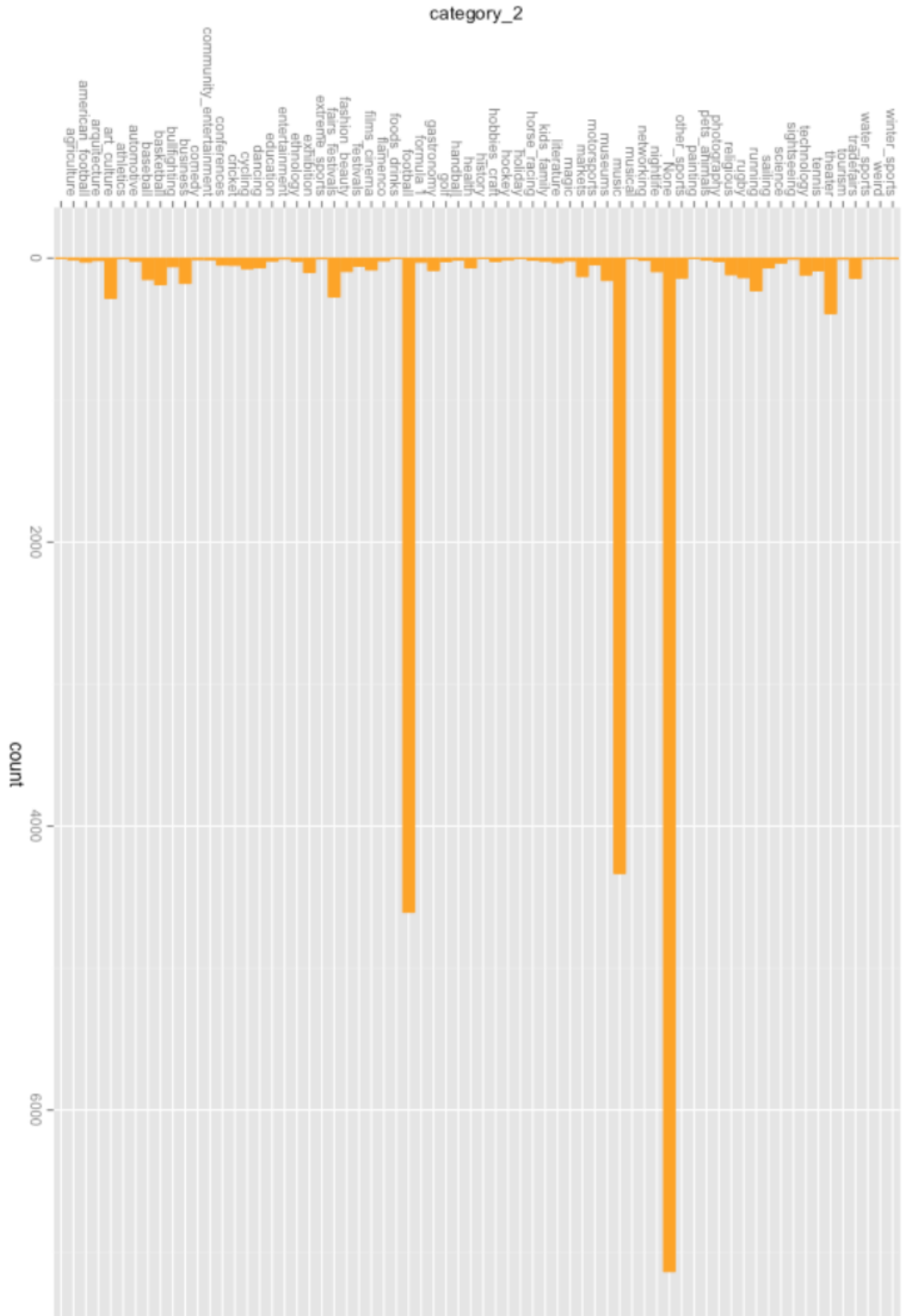
- **Detección de texto:** El análisis de texto de las descripciones en busca de palabras de peso en el mundo de la pornografía, podría limpiar bastantes resultados o dar indicio de posible imagen pornográfica.
- **Detección de texto en la imagen:** Usando técnicas de reconocimiento de texto en las imágenes es posible extraer palabras ilustradas en las imágenes, que en el caso de las imágenes pornográficas suelen ir acorde a la temática. Sería necesario tener al igual que en el caso anterior una lista de palabras y pesos de palabras relacionadas con el mundo de la pornografía.
- **Detección de colores:** Muchas de las imágenes llevan fundidos logos o mantienen patrones de plantillas, cambiando un mínimo porcentaje de la imagen. Analizando los colores de las imágenes se podrían definir patrones de búsqueda de exclusión.

Ninguna de las detecciones es relevante 100% en la detección de pornografía por lo que habría que hacer uso de un cómputo de las tres y si una persona pudiera darle validez a la revisión sería perfecto. En el caso de fallo en la revisión, se anotarán las imágenes en cuestión para reentrenar el modelo en un futuro.

Resultados

La implementación realizada de alguno de los métodos propuestos produjo resultados buenos e interesantes, pero sobre un conjunto pequeño de imágenes pornográficas y no pornográficas. Habría que implementar el resto de medidas propuestas y realizar pruebas con un set más grande.

5. Distribución de eventos en categorías

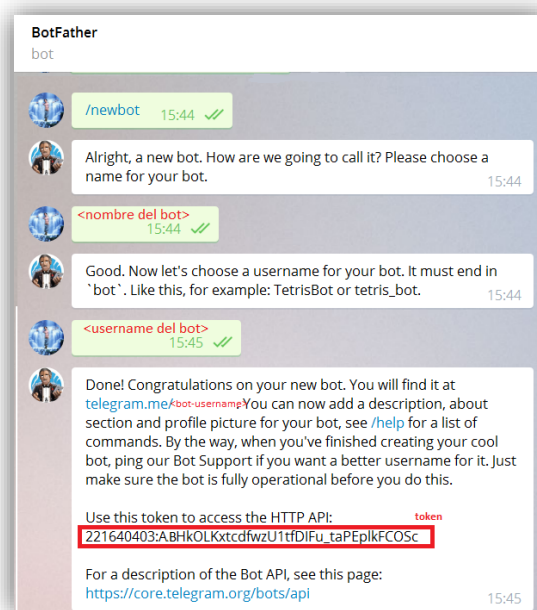


6. Creación de un bot de Telegram

A continuación, se va a mostrar los pasos a seguir para crear y configurar un bot dentro de la aplicación con soporte para redes sociales Telegram.

1. Registro del bot

El primer paso a realizar sería registrar el bot dentro del sistema. Este proceso está automatizado en varios bots de Telegram que hacen el registro del bot en cuestión sencillo. El bot recomendado por excelencia para realizar el registro es **@BotFather**. Se le abriría conversación al bot y se seleccionaría el comando **/newbot**. A continuación, se pedirá la introducción del nombre del bot y una vez enviado el nombre de usuario del bot (este nombre debe terminar con las letras “bot”). Una vez terminado y habiendo sido aceptados los nombres introducidos como válidos, mostrará un mensaje informando de que el bot fue creado satisfactoriamente, incluyéndose en este el **Token** de acceso a la Api para este bot. Este token es necesario para autenticar nuestras peticiones en la Api.



2. Acceso a la Api

El endpoint de acceso al Api permite tanto peticiones GET como POST. La ruta sería la siguiente, <https://api.telegram.org/bot<TOKEN>>. Todos los métodos permitidos por el api se pueden consultar en <https://core.telegram.org/bots/api>.

3. WebHook

En el caso de querer gestionar personalmente los mensajes y redirigirlos a un endpoint propio, se podrá definir un webhook. Para ello se deberá seguir los siguientes pasos:

- Crear un certificado de servidor (puede ser autofirmado).
- Habilitar el protocolo HTTPS mediante el certificado en el endpoint propio.
- Registrar el endpoint y el certificado en Telegram.

7. Instrucciones de uso para el despliegue

El despliegue del sistema automatizado se realizará usando Vagrant y Ansible. Todo el despliegue se llevará a cabo mediante el uso del proyecto de despliegue. El sistema viene configurado por defecto para su despliegue en local, excepto la máquina del bot de Telegram que requerirá de una Ip estática. En el caso de querer hacer uso de despliegues en entornos Cloud, simplemente se subirán los roles desarrollados y las variables correspondientes, en el caso de querer hacer un despliegue gestionado manualmente por un usuario se seguirán los siguientes pasos

1. Descarga del proyecto de despliegue

Lo primero que se debe hacer es descargar el proyecto de despliegue del repositorio de código. Para ello únicamente se requiere tener instalado git en nuestro sistema. Para descargar el repositorio de código se ejecutará el siguiente comando:

Protocolo	Comando
ssh	git clone git@bitbucket.org:EPS_UAM_TFM_MMP/tfm_bender.git
https	git clone https://lobomick@bitbucket.org/eps_uam_tfm_mmp/tfm_bender.git

2. Modificar la configuración

Dentro del proyecto se podrán modificar varios elementos referentes al despliegue:

- **Ip de despliegue de las máquinas**
Se deberá entrar en los ficheros *Vagrantfile* y *devel.hosts* y modificar la ip de las máquinas en cuestión.
- **Port Forwarding**
Se deberá acceder al *Vagrantfile* y añadir el port forwarding a la lista de puertos en la máquina en cuestión.
- **Aumentar memoria de la máquina**
Se deberá acceder al *Vagrantfile* y modificar el tamaño de la memoria reservada para la máquina en cuestión.

3. Despliegue de máquinas

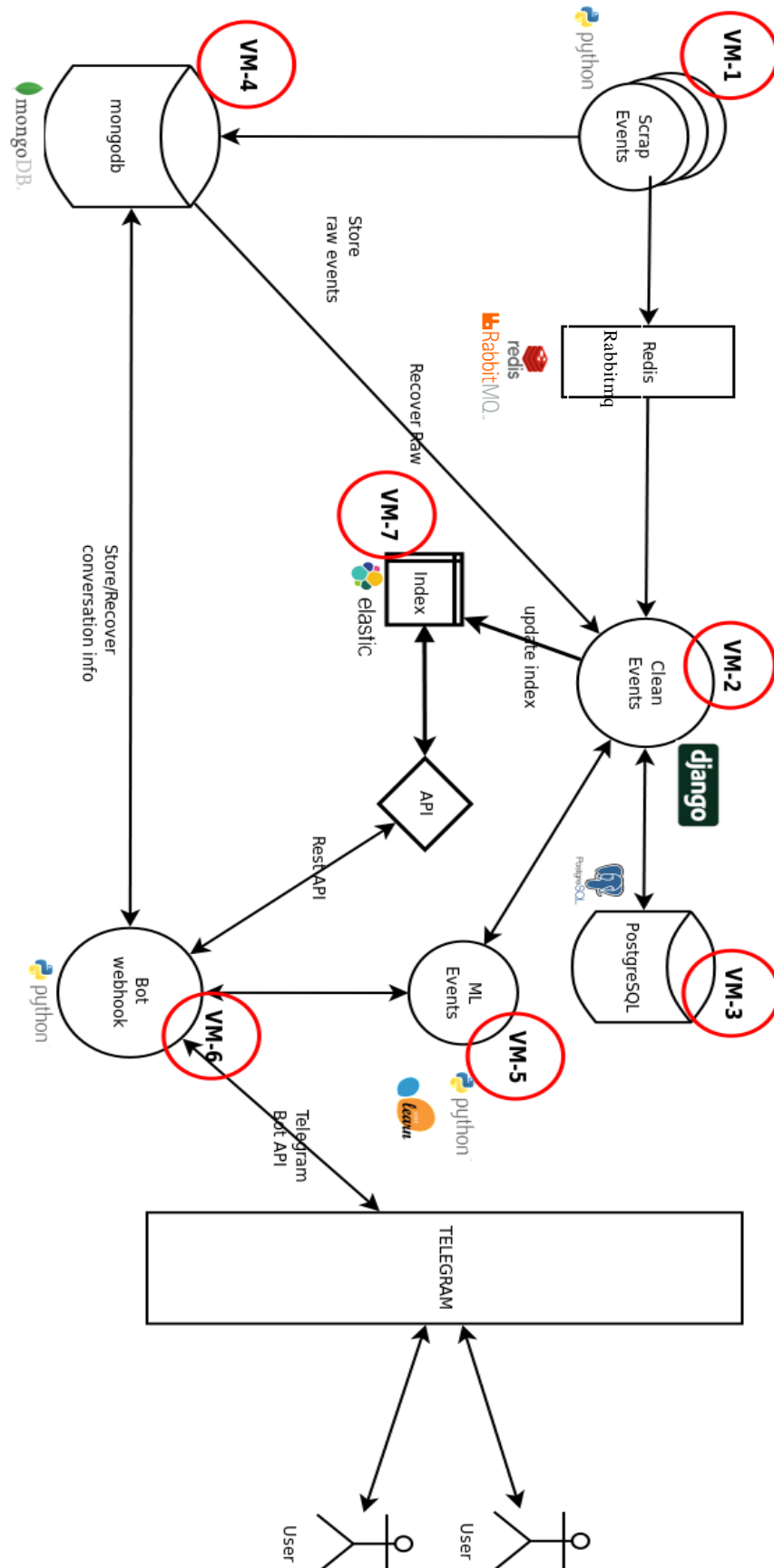
Para el despliegue y gestión de las máquinas se requiere de la instalación de Python, Vagrant y Ansible.

Elemento a instalar	Instrucciones
Vagrant	Vagrant se descargará desde https://www.vagrantup.com/downloads.html para posteriormente instalarlo.
Ansible	Ansible requiere de la instalación previa de Python y pip, ya que se encuentra como librería dentro de este lenguaje. Se instalará mediante el siguiente comando: <i>pip install ansible</i>

Se van a hacer uso de los siguientes comandos de Vagrant para gestionar las máquinas virtuales del proyecto:

Comando	Descripción
Vagrant up <nombre de la máquina>	Crear y configura la máquina virtual.
Vagrant provision <nombre de la máquina>	Reaprovisiona la configuración de la máquina.
Vagrant destroy <nombre de la máquina>	Destruye la máquina
Vagrant status	Muestra el estado de las máquinas del <i>Vagrantfile</i> .

8. Diseño de arquitectura



9. Prueba de uso guiado bot Telegram

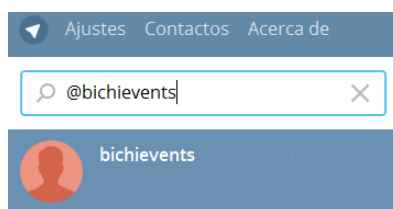
A continuación, se le van a mostrar una serie de actividades a realizar a fin de conocer toda la funcionalidad del Bot Bichibot de Telegram. En el caso de tener alguna duda con alguna de las actividades propuestas que componen la prueba, puede preguntar o mandar un email a miguel.moraleda@estudiante.uam.es

1. Instalar Telegram

Dependiendo de la plataforma que esté usando deberá dirigirse a la página web <https://telegram.org/> con su navegador y elegir su dispositivo para instalar la aplicación. En caso de usar un dispositivo móvil o Tablet podrá usar su market preferido para instalar la aplicación. Una vez instalado se registrará un usuario siguiendo los pasos indicados o se iniciará sesión con un usuario ya existente.

2. Iniciar Bot

En Telegram existen entidades que ofrecen servicios llamados bot. Para iniciar un bot se deberá buscarlo como si fuera un usuario, pero añadiendo '@' delante del nombre. En nuestro caso el bot que se quiere buscar es 'bichievents', por lo tanto, se introduciría la cadena de búsqueda '@bichievents'.



Una vez encontrado se seleccionará pulsando en él, abriéndose una ventana de chat con un botón en la parte inferior que pone "Start". Se pulsará ese botón y con ello se habrá iniciado el bot para ese usuario, pudiendo establecer una conversación con él.

3. Añadir Bot a conversación

Además de tener una conversación individual, los bot se pueden añadir a conversaciones, normalmente para permitir que se nutran de los mensajes y la información vertida en la misma por parte de los usuarios. En el caso de tener una conversación grupal, el añadir un bot a la misma se realiza de la misma forma que añadir una persona más al grupo, se pulsará sobre la imagen del chat abriéndose una ventana con la información del mismo, y se pulsará en añadir miembro. En el buscador se seleccionará el bot en cuestión y este será añadido a la conversación.





En el caso del bot Bichievents el usarlo en conversaciones grupales le permite extraer información tanto de las personas individuales del mismo como del grupo.

4. Comandos de bot

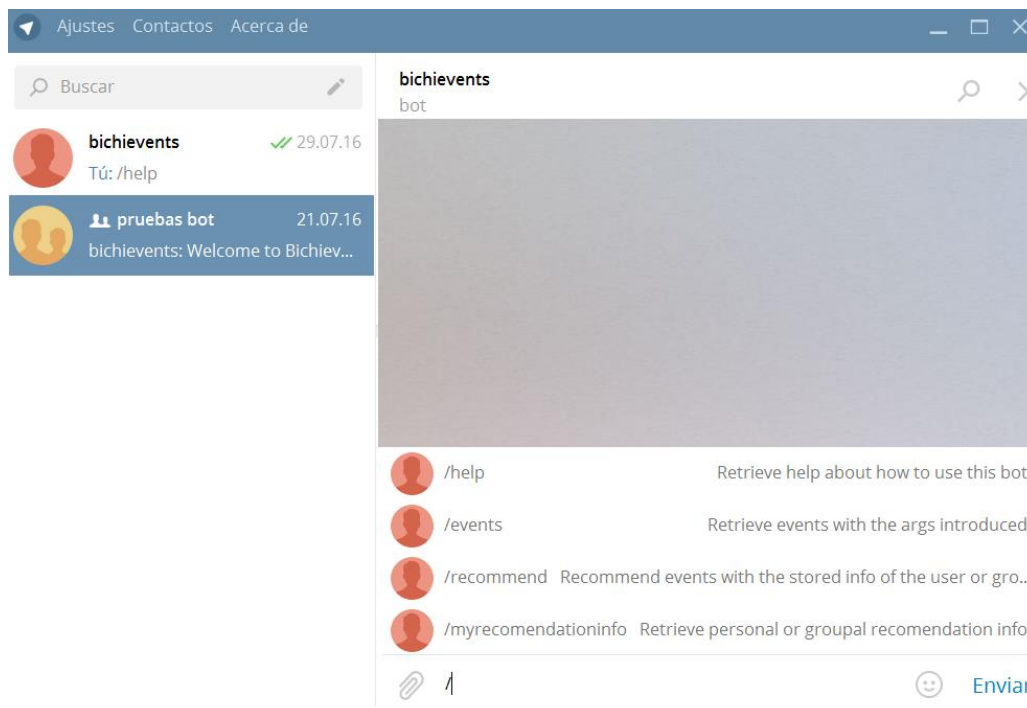
Los bots de Telegram tienen una serie de funcionalidades llamadas comandos que se usan para ejecutar los servicios que proporciona el bot. Estos comandos se escriben en la zona de escritura o se seleccionan del desplegable del bot. Su sentencia consiste en una frase que comienza por '/' seguido del nombre del comando y de argumentos que definen mejor la funcionalidad que quiere el usuario.

A continuación, desplegará la lista de comandos disponibles en bichievents, para ello se pulsará en el botón de despliegue de comandos como indica la imagen inferior.

 |Escribe un mensaje...

Desplegar comandos del bot    

Y a continuación se muestra toda la lista de comandos disponibles y que se irá descubriendo su funcionamiento a lo largo de esta prueba.



5. Bchievents comando help

En el paso anterior se vio como desplegar los comandos y a continuación se va a usar uno de ellos. El comando elegido será el de **/help** que mostrará información sobre las distintas opciones que ofrece el bot. Para ello lo se seleccionará en la lista de comandos el comando **/help** y se le dará a enviar. Una vez enviado devolverá en forma de chat de texto, la información de ayuda de cómo usar los distintos comandos del bot.

Instrucciones
Escribir /help en el cuadro de introducir texto en el chat o seleccionar el comando del menú desplegable de comandos del bot.

6. Bchievents comando events

El siguiente comando que se verá será **/events**, que sirve para obtener información de eventos que ocurren en un lugar y un tiempo determinados. Este comando si es llamado sin añadirle nada más devolverá una lista de eventos. En el caso de querer filtrar más la búsqueda se le puede añadir un nombre de una región y hasta dos fechas, entendiendo la menor como límite inferior de la búsqueda y la mayor como límite superior. En cuanto al orden no hay que preocuparse, ya que el bot se encargará de entenderlos.

Instrucciones
Escribir /events en el cuadro de introducir texto en el chat o seleccionar el comando del menú desplegable de comandos del bot.
Escribir /events madrid en el cuadro de introducir texto en el chat o seleccionar el comando del menú desplegable de comandos del bot y añadirle madrid como argumento.
Escribir /events madrid 2016-07-01 2016-09-01 en el cuadro de introducir texto en el chat o seleccionar el comando del menú desplegable de comandos del bot y añadirle madrid 2016-07-01 2016-09-01 como argumento.

7. Bichievents botones

Una vez completado el apartado anterior, se ha podido observar que han aparecido al final de la lista de eventos una serie de botones, 'More events', 'Evaluate recomendation', 'Previous events recomendation'.



Estos botones permiten realizar interacción sobre la lista de eventos proporcionada. El botón 'More events' devolverá más eventos, el botón 'previous events' mostrará eventos anteriores ya vistos tras haber pulsado el botón 'More events' y el de 'Evaluate recomendation' permitirá al usuario evaluar qué le parecen los eventos proporcionados. La manera de interactuar con estos botones, será simplemente pulsándolos.

8. Bichievents botones more/less

Como ya se dijo antes, los botones more/less nos permitirá mostrar más eventos o volver a los anteriores respectivamente. En esta actividad se va a avanzar 2 veces en la lista de eventos y a retroceder una pulsando los botones.

Instrucciones
Lanzar un comando /events o aprovechar un resultado de la instrucción anterior.
Pulsar el botón More events .
Pulsar de nuevo el botón More events .
Pulsar el botón Previous events .

9. Bichievents botón valoración

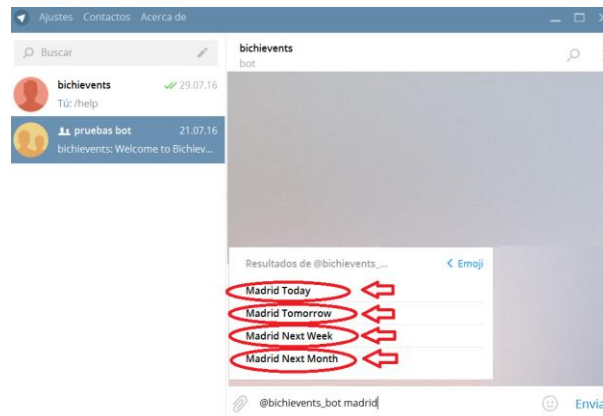
El botón de valoración sirve para que el usuario de su opinión sobre los eventos mostrados y su afinidad a ellos. Una vez que el usuario pulsa el botón de evaluación, le saldrá un teclado con números de 1 al 5, siendo 1 que los eventos son poco afines y 5 muy del agrado del usuario.

Instrucciones
Lanzar un comando /events o aprovechar un resultado de la instrucción anterior.
Pulsar el botón Evaluate recomendation .
Elegir una valoración según tus gustos.

10. Bichievents inline query

Además de los comandos, que para usarlos se debe tener el bot dentro de la conversación, es posible usar el bot usando lo que en Telegram se llama inline query. Para ello no es necesario que el bot esté incluido en la conversación, aunque con ello se perderá la capacidad de que el bot sea consciente de lo que en esta ocurre. La forma de usar los inline query en Telegram es introduciendo la sentencia formada por '@', el nombre del bot y algún texto más, en la caja de introducción de texto en el chat. A continuación, la interfaz reacciona y te muestra opciones de con un selector emergente, y al elegir una tiene su efecto. En el caso del bichievents funciona con los mismos parámetros que el comando **/events** y al introducir el inline query aparecerán distintas

opciones para elegir en cuanto a rangos de fechas. Una vez elegido una se mostrará como resultados la lista de eventos correspondiente y sus botones.



Instrucciones
Escribir '@bichievents_bot' en el cuadro de introducir texto en el chat.
Observar las opciones y seleccionar una.
Escribir '@bichievents_bot madrid' en el cuadro de introducir texto en el chat.
Observar las opciones y seleccionar una.
Confirmar que han salido los mismos resultados que usando el comando /events anteriormente.

11. Bichievents comando recomend

El comando **/recomend** funciona exactamente igual que el comando **/events** en cuanto a argumentos que se le pueden añadir, pero le da al sistema libertad para que, a partir de lo aprendido del usuario, le recomiende eventos de forma personalizada. El bot de forma transparente al usuario, aprende tanto de las valoraciones que este hace como de los mensajes enviados en el chat en el que se encuentra, y todo esto le permite saber los gustos del usuario. Es posible que en esta prueba no acaben de salir resultados de recomendación muy afines al usuario debido a su corto tiempo de uso.

Instrucciones
Escribir /recomend en el cuadro de introducir texto en el chat o elegir el comando en el desplegable de comandos del bot.
Observar si los eventos recomendados son del agrado del usuario.
Escribir /recomend madrid en el cuadro de introducir texto en el chat.
Observar que los eventos recomendados son ahora solo de Madrid y de temáticas aparentemente afines al usuario.
Escribir /myrecomendinfo en el cuadro de introducir texto en el chat o elegir el comando en el desplegable de comandos del bot.
Observar la lista de categorías que aparecieron, comprobar que se está de acuerdo con ella y el orden de la misma.

Hasta aquí llega la prueba para conocer bichibot, se ruega a los participantes que antes de realizar la encuesta anexa a esta prueba, den un tiempo de uso al bot a fin de incrementar la experiencia de usuario.

Gracias por su participación

10. Encuesta pruebas bot Telegram

El formulario de encuesta se encuentra disponible en <https://goo.gl/forms/c5sY0oRbFBVekPKF2>

Por favor, indique su grado de conformidad o disconformidad con cada una de las siguientes afirmaciones donde 1 = Muy en Desacuerdo / 2 = En Desacuerdo / 3 = Ni de Acuerdo ni en Desacuerdo / 4 = De Acuerdo / 5 = Muy de Acuerdo

	1	2	3	4	5
He usado antes un bot de Telegram.					
Me resultó interesante la aplicación.					
Me resulto fácil de aprender a utilizarlo.					
Me ha resultado útil la aplicación.					
Me resulto más sencillo que buscar en el ordenador.					
Me resultaron repetitivas las recomendaciones.					
Siento que tengo el control sobre la herramienta.					
Es fácil de usar.					
La interacción me pareció cómoda.					
Usarla requiere poco esfuerzo.					
Entiendo perfectamente los textos de la herramienta.					
La información presentada era fácil de entender					
Las recomendaciones se asemejaban a mis gustos.					
He sentido invadida mi intimidad.					
El bot supo extraer mis gustos correctamente.					
Recuerdo fácilmente como utilizarla.					
No hizo falta ayuda para su uso					
Me gustan este tipo de iniciativas.					
Me pareció una experiencia positiva.					
Me ha parecido compleja la prueba.					
Entendí las instrucciones de la prueba.					

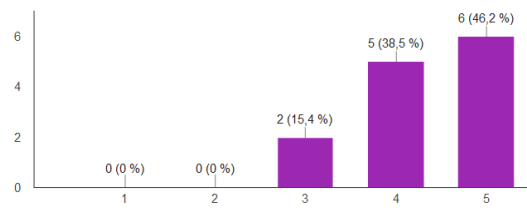
Preguntas Abiertas

Señale los aspectos que más le hayan gustado de la herramienta:

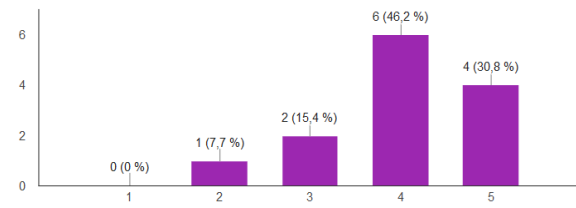
Señale los aspectos que menos le hayan gustado de la herramienta:

11. Resultados de la encuesta de pruebas bot Telegram

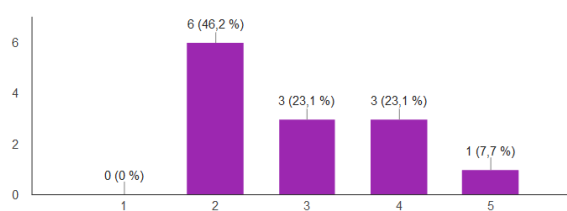
Me resultó interesante la temática de la aplicación.



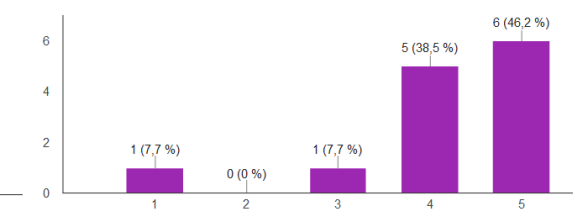
Me ha resultado útil la aplicación.



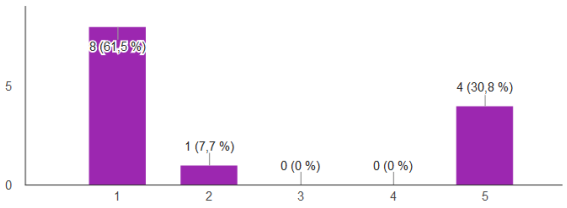
Me ha parecido compleja la prueba.



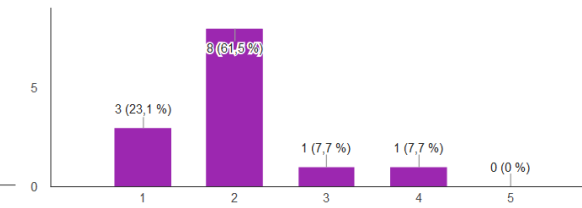
Entendí las instrucciones de la prueba.



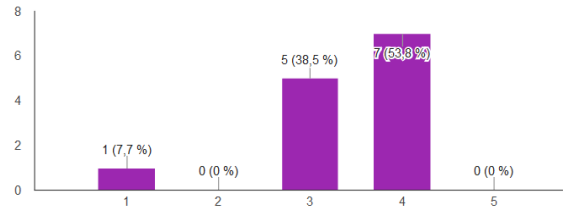
He usado antes un bot de Telegram.



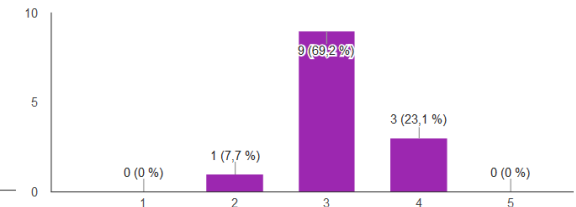
Me resultaron repetitivas las recomendaciones.



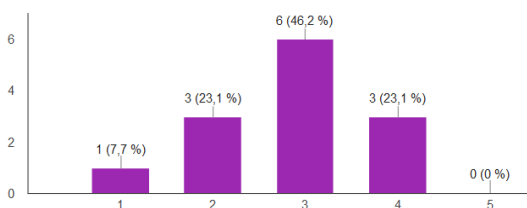
Las recomendaciones se asemejaban a mis gustos.



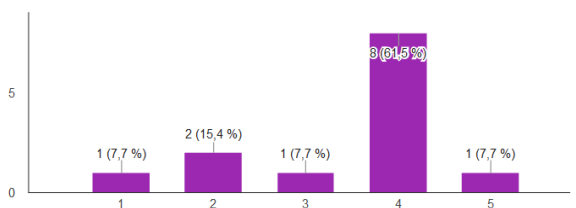
El bot supo extraer mis gustos correctamente.



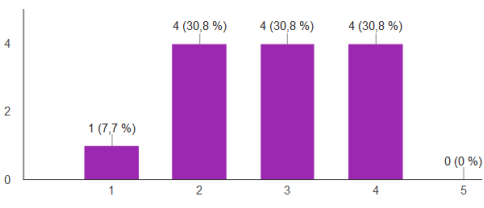
La interacción me pareció cómoda.



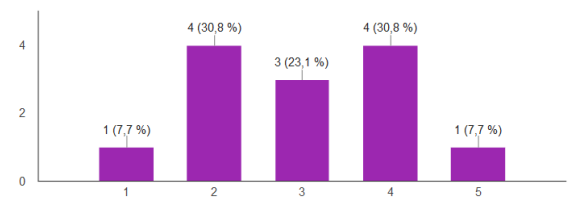
Usar el bot requiere poco esfuerzo.



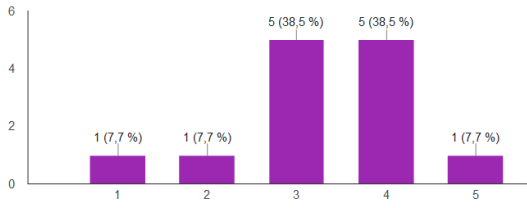
Me resultó más sencillo que buscar en el ordenador.



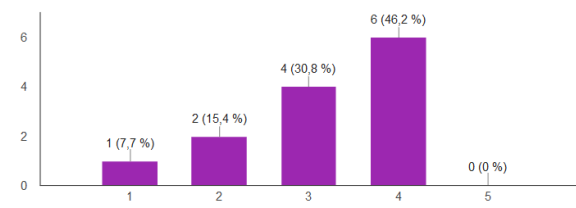
Me resultó fácil de aprender a utilizarlo.



Siento que tengo el control sobre la herramienta.

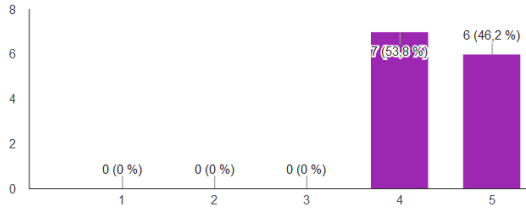


Es fácil de usar.

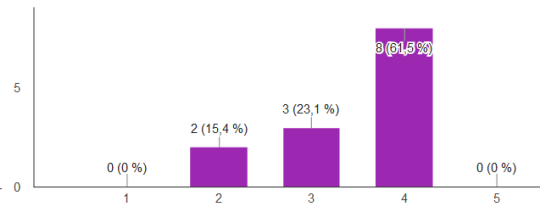


Recomendación de eventos sociales por texto y colores

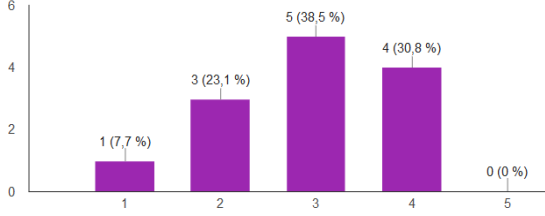
Entiendo perfectamente los textos de la herramienta.



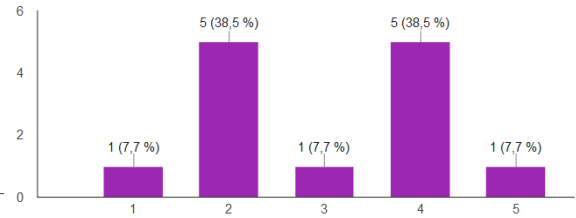
La información presentada era fácil de entender



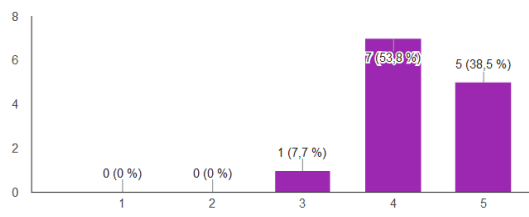
Recuerdo fácilmente como utilizarla.



No hizo falta ayuda para su uso.



Me gustan este tipo de iniciativas.



Me pareció una experiencia positiva.

