uc3m | Universidad Carlos III de Madrid

Grado en Ingeniería Informática

2017-2018

Trabajo Fin de Grado

Diseño y construcción de un Bot de entretenimiento

David Morcuende Cantador

Tutor

Araceli Sanchis de Miguel

Leganés, 2018

RESUMEN

Este Trabajo de Fin de Grado (TFG) versa sobre la construcción de un ChatBot que permite a los usuarios pedir tanto por texto como por voz (en lenguaje castellano) recomendaciones por género y por similitud, información sobre películas y series.

Para desarrollar este sistema ha sido necesario construir diversos módulos: un sistema de reconocedores de lenguaje natural castellano, una aplicación Android, un Bot y un sistema reconocimiento. La función principal del sistema de reconocedor de lenguaje natural es procesar la entrada de texto del usuario y extraer información de las intenciones del usuario y las palabras claves en el texto, para ello se hacen competir modelos estadísticos, redes de neuronas y gramáticas. La aplicación Android es la interfaz de entrada y salida de información, es el primer punto de acceso del usuario con el sistema., y es la interfaz que muestra la información del Bot al usuario. El sistema recomendador se encarga de dar recomendaciones a los usuarios usando modelos preentrenados, dichos modelos se basan en la aplicación de la similitud coseno y la función Kernel polinomial.

Cabe destacar que la infraestructura de todo el sistema ha sido diseñada para ser ejecutada en la nube con scripts de despliegue automático usando la tecnología de contenedores Docker para hacer que los sistemas sean eficientes, eficaces y fáciles de mantener y monitorizar.

Para desarrollar toda la tecnología mencionada se han llevado a cabo análisis sobre las tecnologías actuales y las que más beneficiarían al sistema en su conjunto para que este sistema transcienda más allá de este proyecto, y que sea mantenible y actualizable como un producto de mercado, beneficiando y ayudando a cualquier usuario que pretenda usar la aplicación.

Palabras clave

ChatBot; Procesamiento de Lenguaje Natural; Sistemas recomendadores; modelos estadísticos; redes de neuronas; gramáticas

ÍNDICE DE CONTENIDOS

Índice de figuras

Índice de tablas

1. INTRODUCCIÓN

En este apartado se describe el contexto y la motivación de este Trabajo de fin de grado, TFG, que explica la temática estudiada e implementada y su por qué. Además de instanciar los objetivos marcados, que serán ampliamente explicados en los siguientes apartados tanto su investigación, diseño y posterior implementación. En esta sección también se aclara toda la tecnología usada en la realización del TFG, tanto Hardware como Software. Por último, se describe la estructura de esta memoria indicando los apartados que la componen y un breve resumen de los mismos.

1.1 Contexto y motivación

A día de hoy y con el auge de la automatización, los asistentes virtuales, la domótica… las personas buscan maneras cada vez más fáciles y cómodas para solucionar sus problemas. Así, mucha gente ya se apoya en asistentes virtuales véase Siri o Google Assistant para que resuelvan pequeñas tareas mundanas de forma rápida, como puede ser, poner una alarma a una determinada hora, o preguntarle por el tiempo que va a hacer hoy… estos asistentes se encuentran embebidos en los *smartphones* que hoy en día la gran mayoría de personas lleva en el bolsillo, Como se puede ver en la Figura 1 el número total de usuarios de smartphone en el mundo crece cada año en varios cientos de miles y se estima que para 2019 los usuarios asciendan a más de 2 millones y medio.

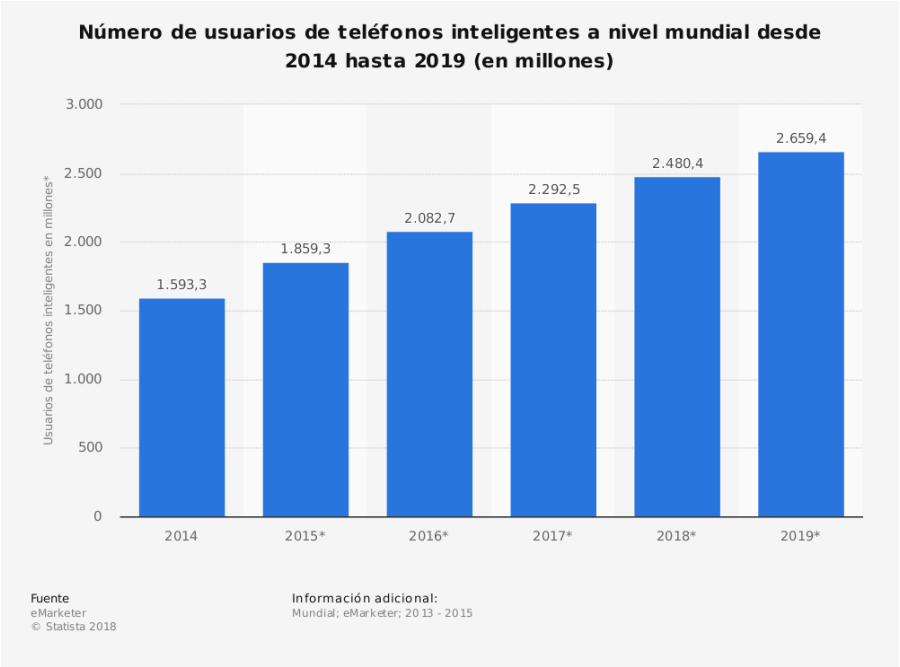


Figura 1 Usuarios de teléfonos Inteligentes [1]

Por otro lado, cada vez se consume más video bajo demanda ya sea porque las personas cada vez tienen un tiempo más limitado o porque la serie que se quiere ver no se emite en un horario conveniente Esto ha dado lugar a plataformas que, bajo el pago de una suscripción, ya sea anual o mensual, permiten ver contenido *a la carta* mediante video en *Streaming*. Como se observa en la Figura 2 el porcentaje de subscripciones en Europa está creciendo exponencialmente.

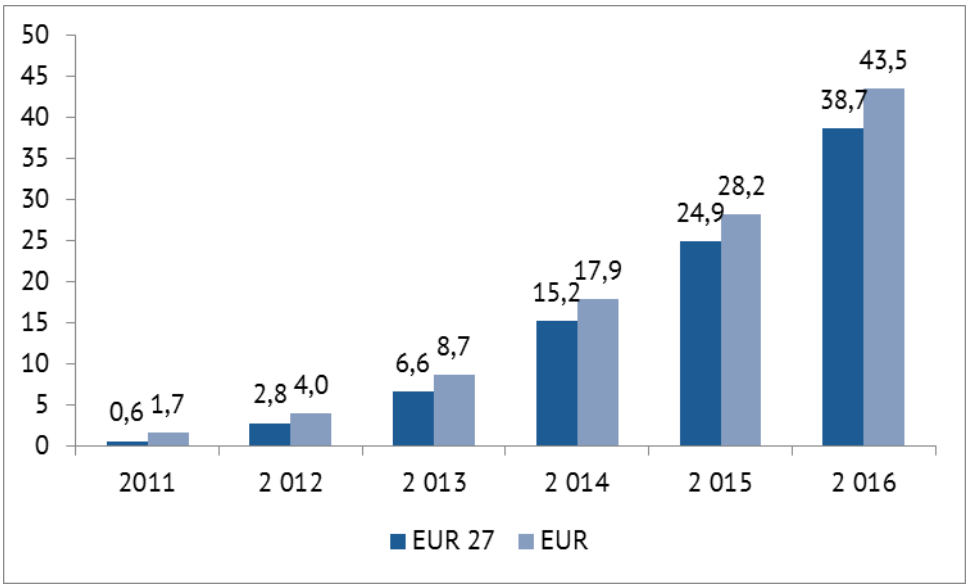


Figura 2: Porcentaje de subscripciones plataformas de streaming en Europa [2]

Aunando estos dos datos, se llegó a la conclusión de que cada vez resulta más necesario un asistente, que esté embebido en un smartphone, que permita de manera rápida preguntar por un contenido que se desee visionar, o si no se tiene claro que es lo que se quiere ver, que se pueda preguntar por una recomendación en basada, por ejemplo, en otra película o a los gustos del usuario.

La creación de un sistema como este permite al desarrollador situarse en un campo que aún se está desarrollando y así poder contemplar desde dentro como van evolucionando este tipo de sistemas.

Habiendo analizado datos y conclusiones ya se tiene una idea dibujada del problema y de cómo ofrecer una solución, pero para poder dar una **buena solución** hay que establecer unos objetivos.

1.2 Objetivos

El objetivo de este proyecto es desarrollar un ChatBot que permita una interfaz textual tanto escrita como hablada en lenguaje castellano que dé a los usuarios la posibilidad de hacer preguntas sobre películas y series, y pedir recomendaciones basadas en otras películas y/o géneros. Además de preparar la infraestructura para su despliegue automático en servidores o entornos Cloud.

Para ello es necesario diseñar una aplicación Android que haga de intermediario bidireccional entre el usuario y el Bot que, frente a una entrada de texto devolverá una o varias respuestas.

A parte del diseño de la aplicación Android será necesario diseñar el Bot que será el encargado de procesar el texto y dar una respuesta coherente con la finalidad que busca el usuario. Esto requiere la generación de un **módulo cognitivo** que se encargue del procesado textual y de un **módulo recomendador** en el caso de que el usuario pida recomendaciones.

Con toda esta información, el proyecto deberá centrarse en:

- Diseñar y construir un ChatBot que sea capaz de dar respuestas acertadas y concretas frente a una petición de usuario, resolviendo así el problema de tener que navegar entre cientos de películas para leer su sinopsis y quizá encontrar algo que le guste habiendo perdido mucho tiempo.
- Desarrollar una aplicación móvil que permita al usuario interactuar con el Bot. Se elige el móvil como interfaz ya que en España el uso de smartphones se ha duplicado en los últimos 5 años [3] y además España es el quinto país del mundo en el que más tiempo pasa con el teléfono, con una media de 2 horas y 11 minutos diarios por usuario [4].
- Dicha aplicación deberá tener un diseño sencillo y minimalista, siguiendo las buenas prácticas dictadas por *material design* [5] y las reglas de la Heurística de Nielsen [6]. Dicho diseño está pensado para que la interfaz sea fácil de usar, aunque cuando se trata de un ChatBot con una canal conversacional también hay que tener en cuenta su diseño para que el Bot genere frases adecuadas que guíen al usuario a formular preguntas que generen respuestas correctas y el usuario no se frustre si el Bot en algún punto de la conversación no entiende a la perfección su petición.

- Será necesario construir un sistema recomendador que sea capaz de dar recomendaciones correctas frente a la petición de género, o de encontrar similitudes entre películas.
- Uno de los componentes principales será el de Procesamiento del lenguaje natural ya que es el que dictaminará que intención tenía el usuario al hacer la petición que hizo, si el sistema consigue interpretarlo correctamente, será capaz de dar una respuesta acertada, si no, el sistema no sabrá que contestar o responderá algo distinto a lo esperado.

  Todos los módulos se basarán en la tecnología de contenedores Docker para que el control de cada sistema sea exhaustivo y pueda ser mantenido con facilidad. [7]
- Una vez establecidos los objetivos, se pasa a describir el hardware y software necesario.

1.3 Hardware y software utilizado

Para el desarrollo del proyecto ha sido necesario contar con los siguientes recursos tanto hardware como software.

1.3.1 Hardware

- Ordenador: Asus R510V
- Smartphone: Samsung Galaxy S7 Edge

1.3.2 Software

- Entorno de desarrollo Visual Studio Code.
- Entorno de desarrollo PyCharm.
- Node.js: Entorno de ejecución para JavaScript, asíncrono con arquitectura de entrada y salida basada en eventos.
- Python: Lenguaje de programación interpretado
- Git: Software de control de versiones
- Gitkraken: Interfaz gráfica de para Git.
- Elasticsearch: motor de búsqueda y análisis RESTful de código abierto
- Kibana: Asistente grafico para el análisis en tiempo real de los datos de Elasticsearch.
- Docker: Software de código abierto que favorece y automatiza el despliegue de aplicaciones, proporcionando una capa de virtualización.
- Portainer.io: interfaz gráfica para el mantenimiento de los contenedores Docker.

- Mongodb: base de datos NoSQL orientada a documentos.
- Microsoft Bot Framework: Framework de desarrollo de Bots proporcionado por Microsoft
- ANTLR: software para generación y procesamiento de gramáticas.
- Trello: software de administración de proyectos con un tablero tipo Kanban.

1.4 Estructura de la memoria

En este apartado se explica la estructura de la memoria, que en este caso está dividido en 9 epígrafes, cada epígrafe se explica por separado a continuación:

1. **Introducción**: epígrafe en el que se describe el contexto en el que se enmarca el proyecto, la motivación que ha llevado a elegirlo y a realizarlo, los objetivos marcados para lograr el proyecto, los medios tanto software como hardware necesarios para la realización del proyecto y por último una descripción de la estructura de la memoria.
2. **Estado del arte**: capítulo en el que se explica la tecnología más innovadora que se está usando en el campo en el que se desarrolla el proyecto, también se hace un análisis del contexto de la tecnología y como esta afecta al día a día y se relaciona con las tecnologías que se han usado en este proyecto.
3. **Bot**: apartado en el que se profundiza sobre la construcción del proyecto, desde la fase más inicial de la toma de requisitos, pasando por el diseño hasta, finalmente la implementación. Tras hacer un análisis general de todo, se especifica para cada módulo los detalles pertinentes.
4. **Evaluación**: apartado en el que se realiza una comparativa entre los distintos módulos de reconocimiento del lenguaje natural a partir de técnicas de análisis de resultados: validación cruzada y pares de subconjuntos de entrenamiento-test.
5. **Gestión del proyecto**: Capítulo en el que se recoge la planificación del proyecto, tanto temporal (diagrama Gantt) como monetaria (presupuesto y costes), en este apartado se incluyen el entorno socioeconómico y el marco regulador.
6. **Conclusiones**: apartado en el que se realiza una retrospectiva del proyecto indicando aspectos clave del mismo.
7. **Trabajo futuro**: capítulo en el que se describen las tareas a realizar si se quisiera seguir con el proyecto y mejorarlo.
8. **Glosario**: capítulo en el que se recogen términos técnicos usados en este documento junto con su definición para facilitar la comprensión del lector.

9. **Bibliografía**: apartado donde se encuentran la lista de referencias bibliográficas de las fuentes de información usadas en este documento. La ordenación de las mismas sigue un sistema de citación numérico, las referencias se ordenarán por orden de aparición en el texto.

2. ESTADO DEL ARTE

En este apartado se desarrollará de una manera más detallada las posibilidades que ofrece un sistema como el que se ha desarrollado, en el marco de la sociedad actual, definiendo la tecnología usada, el motivo por el que se ha elegido y las posibilidades que ofrece.

Primero se hará una reflexión sobre el estado actual del mundo móvil y se hará hincapié en el sistema operativo Android, el sistema operativo en el que se ha desarrollado la aplicación móvil.

Después se hablará sobre el concepto de los Bots, qué son y cómo derivan en los ChatBots.

Finalmente se explicará con detalle la arquitectura de un ChatBot, los módulos que lo compone y la tecnología usada para su implementación.

2.1 Tecnología Móvil

Como se ha expuesto anteriormente los dispositivos móviles están sufriendo un auge muy notable, cada vez más gente da el salto de disponer de un simple móvil para llamar o mandar mensajes únicamente, a un teléfono inteligente (*smartphone*), de acuerdo con los datos recopilados por la compañía Zenith un 66,5% de los adultos poseen un teléfono inteligente, que en España, corresponde a que un 88,9% de los adultos españoles poseen uno de estos teléfonos [8].

Este auge masivo viene dado porque los smartphones cada vez son más accesibles a las personas porque sus precios han bajado y hay una alta competitividad en el mercado, lo que permite a los usuarios tener un gran abanico de opciones a elegir.

Pero además, según un estudio de la agencia de publicidad IAB, un 65% de los usuarios de smartphones lo usan cada 30 minutos, y un 22% lo mira cada 5 [9].

Esto es debido a que es un dispositivo altamente portátil y con una alta capacidad computacional. El teléfono inteligente gracias a su alta gama de aplicaciones aporta a los usuarios facilitadores en el día a día, como comunicación en tiempo real, calendarios, redes sociales, entretenimiento, etc. De media un usuario tiene 17,8 aplicaciones instaladas [10].

El uso del móvil esta tan extendido en el día a día que muchos sectores se ven beneficiados porque aportan ventajas competitivas que favorecen la relación con sus usuarios. Es por eso por lo que se ha optado desarrollar la aplicación para dispositivos móviles.

2.1.1 Android

En cuanto al sistema operativo a elegir para el desarrollo de la aplicación hay que pensar en que es lo que se busca. Si lo que se busca es exclusividad o imagen de marca habrá que hacer un estudio sobre las posibilidades de éxito con esa marca/sistema operativo, si por el contrario lo que se pretende es llegar al mayor número de personas, sin lugar a duda la opción a elegir es Android. En la siguiente gráfica se muestra la cuota de mercado según el sistema operativo en España. En la Figura 3 se puede observar como en el primer trimestre de 2018 la cuota de mercado de Android es del 86,1% frente a IOS su siguiente competidor que tan solo cuenta con un 13.6%

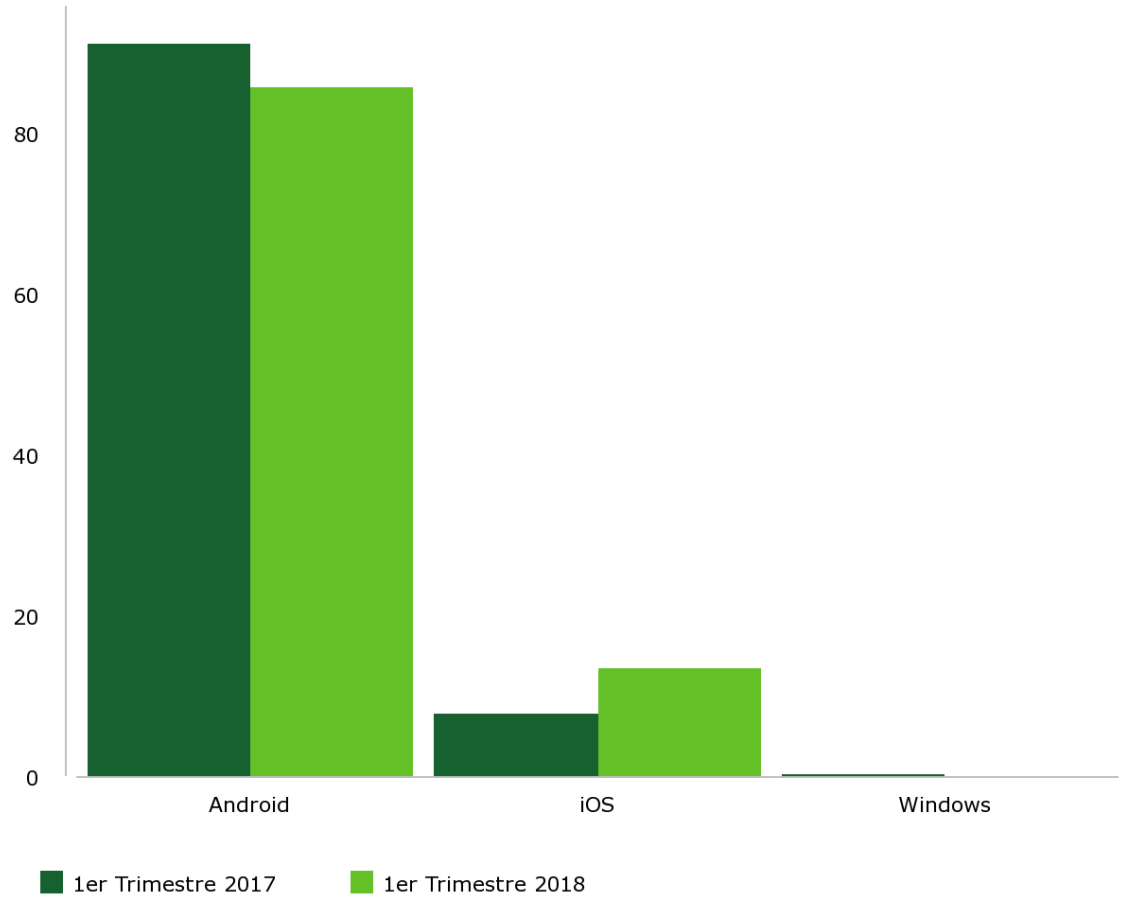


Figura 3: Cuota de mercado de sistemas operativos en España (%) [11]

Por lo que el principal motivo por elegir desarrollar la aplicación en el sistema operativo Android se debe a que es el que lidera la cuota del mercado.

2.2 Bot

Un Bot es un programa de ordenador que realiza tareas normalmente repetitivas y fáciles de definir en pasos concretos.

Se estima que un 60% de trabajos tienen al menos un 30% de tareas automatizables [12]. Es por eso que el mundo de los Bots está en auge porque una vez automatizada la tarea solo es necesario ejecutarla rutinariamente.

Pero el problema llega cuando se desea tener un interfaz con ese automatismo y que además sea capaz de realizar más de una tarea, ahí entran en juego los ChatBots.

2.3 ChatBot

Un ChatBot es un programa capaz de interactuar con el usuario usando el lenguaje natural [13], dicho usuario puede ser una persona u otro Bot.

Su funcionamiento se basa principalmente en 3 partes:

1. El usuario hace una petición usando lenguaje natural
2. El agente procesa esa solicitud
3. El agente da una, o varias respuestas en un tiempo corto, como si de una conversación se tratase.

Una interacción de un humano con un Bot podría representarse mediante el esquema de la Figura 4:

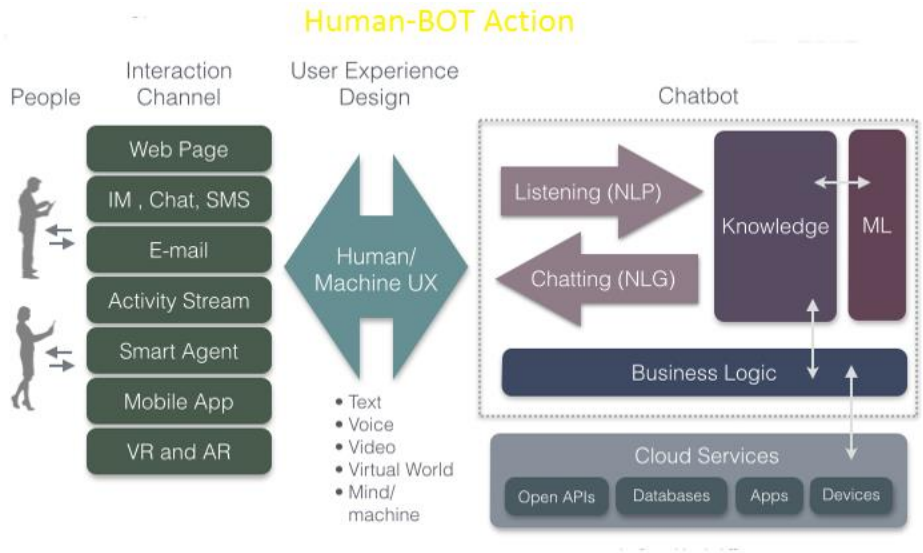


Figura 4: Interacción humano-Bot [14]

2.3.1 Procesamiento del lenguaje natural

La primera parte, y principal, del ChatBot es la entrada, que será mediante lenguaje natural (en este caso concreto en castellano), y para convertir ese texto en algo con algún tipo de significado conexo es necesario procesarlo. El procesamiento de lenguaje natural es un campo de la computación y la lingüística cuyo propósito es el de estudiar el significado de una frase en su contexto, para ello es necesario realizar un análisis morfológico, sintáctico, semántico y pragmático.

La tecnología para implementar los módulos de procesamiento de lenguaje natural no está consensuada, ya que es un ámbito que se está desarrollando y aún no hay ningún método que destaque claramente sobre los demás [7415]. A continuación, se describirá la tecnología usada en este proyecto, así como los tipos de modelos que se pueden construir con cada una de ellas:

Métodos estadísticos

Para esta implementación se ha usado el software LUIS (Language Understanding) [16] provisto por Microsoft.

Se trata de un método de aprendizaje no supervisado en el que para una frase dada hay que definir intenciones y entidades:

- Intención: Se identifica con el verbo de la oración y expresa el objetivo que quiere conseguir el usuario con esa oración.
- Entidad: palabra o palabras clave que pueden ser agrupadas en una familia de palabras.

Por ejemplo, en la oración: *busca la película Thor*.

- La intención es **buscar**, ya que la finalidad de la oración es obtener información sobre algo.
- Las entidades son:
    - *Película*: entidad de **tipo de contenido**, ya que entre los tipos de contenido pueden estar las películas, las series, los cortos, etc.
    - *Thor*: entidad de título, ya que es el título de una película.

Como LUIS es un software comercial no se dispone de mucha información sobre sus algoritmos, pero se sabe que utiliza métodos estadísticos y heurísticas para hacer la clasificación. Entre sus estadísticas, se tiene en cuenta:

- La longitud de la oración
- Cuantas entidades tiene la oración
- Qué entidades tiene la oración

Redes de neuronas (Word2vec)

Las Redes de Neuronas Artificiales surgen en los años 40 con el objetivo de crear sistemas artificiales que modelen el funcionamiento del cerebro [17]. Se usan de forma extensa en diferentes problemas reales. Además, existen investigaciones recientes que han dado lugar a un nuevo paradigma, las redes de neuronas convolucionales o Deep Learning, modelo propuesto por Yann LeCun [18] y que en 2012, fueron refinadas por Dan Ciresan et al [19].

Para la implementación de las redes de neuronas en este trabajo se ha usado el software libre Rasa NLU [20]. Como LUIS, también es un método de aprendizaje no supervisado, y los modelos de entrenamiento se generan de la misma forma, para una oración dada hay que catalogar su intención y etiquetar sus entidades.

Sin embargo, este algoritmo utiliza Word2vec + heurísticos para generar los modelos.

Word2vec viene del inglés *Word to vector* convertir una palabra en un vector y su principal funcionalidad es: dado un vocabulario generado con las palabras de las frases que generan el corpus, el objetivo es entrenar una red de neuronas para que, dada una palabra, obtenga la probabilidad de que cada palabra sea vecina de la primera.

Para generar los modelos de entrenamiento es necesario agrupar las palabras en pares, y habrá que agrupar tantas palabras para una dada, como el tamaño que se defina, veamos un ejemplo con la frase en ingles *The quick Brown fox jumps over the lazy dog* y un tamaño de ventana de 2, como se muestra en la Figura 5:

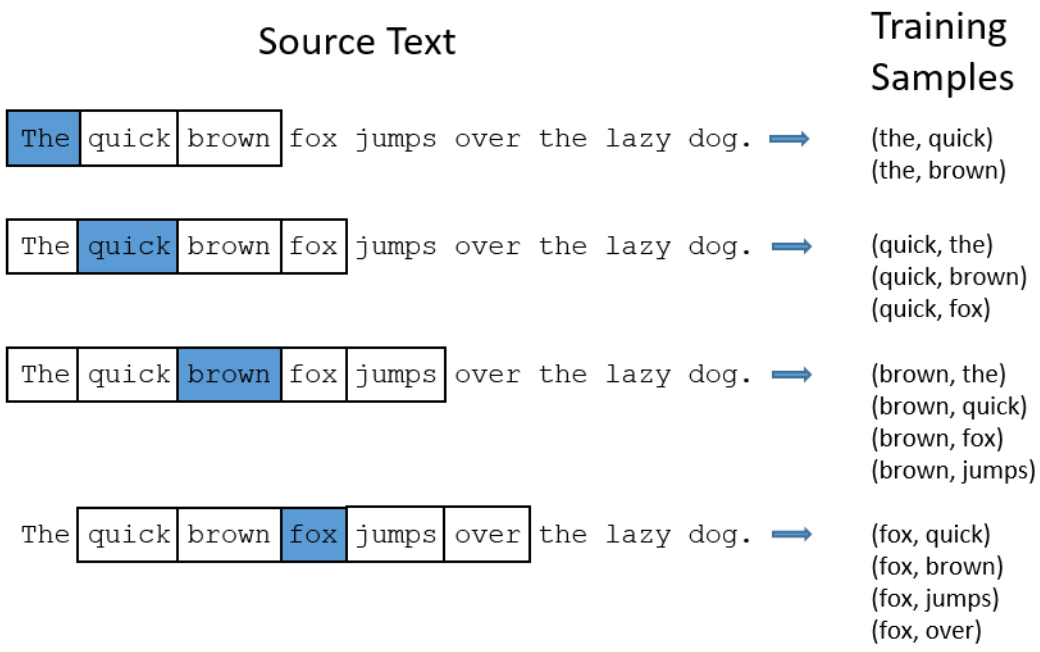


Figura 5: Ejemplo Word2vec [21]

Para usar el modelo una vez entrenado, se deberá usar un *one-hot-vector* (Vector que tiene solo un 1 en una posición y en las demás un 0, por ejemplo [0,0,0,1,0]) cuyo tamaño será el del total de palabras que maneje el corpus y que cada posición hará referencia a una palabra en el corpus. Veamos un ejemplo con un corpus de 10.000 palabras, como en la Figura 6.

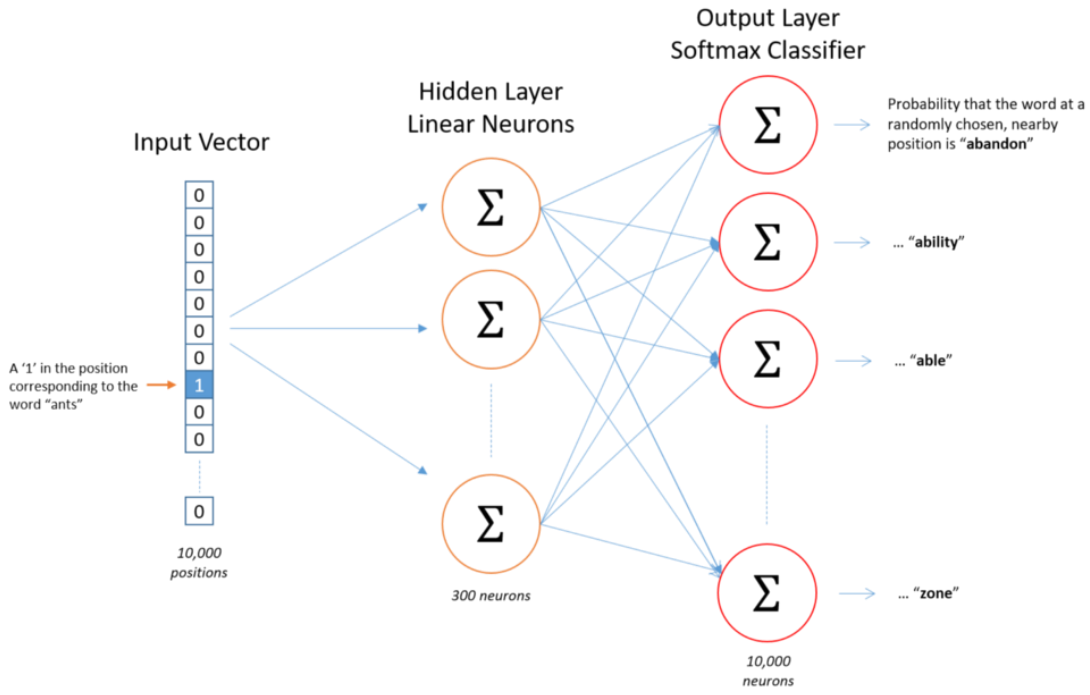


Figura 6: Ejemplo de uso de modelo de word2vec [21]

En el ejemplo anterior se usan 300 neuronas en la capa oculta, esto quiere decir que se están tomando 300 características para usar el modelo, este número es representativo porque es el que Google usó en el dataset que usó con Google News [22], aunque este hiperparámetro puede ser modificado a voluntad para buscar mejoras y/u optimizaciones.

Un ejemplo matemático, Figura 7, usando:

- Corpus de 5 palabras
- 3 neuronas en la capa oculta (3 características)

$$[0 \quad 0 \quad 0 \quad 1 \quad 0] \times \begin{bmatrix} 17 & 24 & 1 \\ 23 & 5 & 7 \\ 4 & 6 & 13 \\ 10 & 12 & 19 \\ 11 & 18 & 25 \end{bmatrix} = [10 \quad 12 \quad 19]$$

Figura 7: Ejemplo calculo word2vec [22]

A la salida de la operación anterior habrá que calcular el vector correspondiente a la salida de la capa oculta para obtener la capa de salida.

A la capa de salida (los resultados) deberá aplicarse la **función exponencial normalizada** (softmax) para que la salida sea un vector normalizado [0-1] que identifique cada palabra con la probabilidad de que sea la palabra correcta [21], Figura 8.

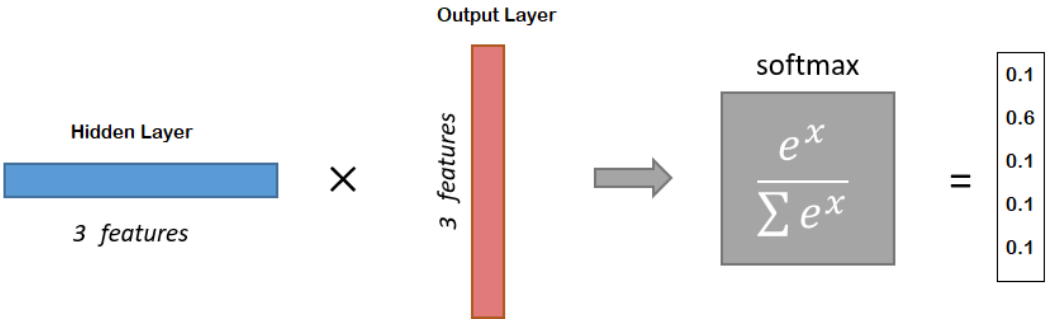


Figura 8: Ejemplo calculo final word2vec[2121]

Gramáticas

Una gramática puede definirse como el grupo de principios, reglas y preceptivos que rigen el empleo de un lenguaje particular. Lo que quiere decir que en la gramática se explica la forma en que los elementos de la lengua se enlazan para formar frases y estas, textos [23].

En concreto se han utilizado gramáticas libres de contexto: una gramática libre de contexto es una gramática formal en la que cada regla tiene la forma $V \rightarrow w$, siendo V un símbolo No terminal y w una cadena de terminales y/o no terminales.

A estas gramáticas se las conoce como libre de contexto ya que el No terminal V siempre puede ser sustituido por w sin tener en cuenta el contexto en el que ocurra [24].

Para implementar las gramáticas se cuenta con el software ANTLR (ANother Tool for Language Recognition), software que proporciona un marco para construir parsers, intérpretes y compiladores. De todas las funcionalidades que ofrece ANTLR, en este proyecto solo se ha focalizado en la parte del parsing, para determinar si una sentencia o palabra pertenece al lenguaje conocido por la gramática, para ello se basa en los algoritmos LL(*), este algoritmo es muy popular ya que aporta funciones de optimización y mejoras frente a otros algoritmos como puede ser LR ya que sólo necesita ver el siguiente token para hacer el análisis de sus decisiones [25].

A la hora de hacer el parsing, ANTLR utiliza arboles sintácticos y teoría de grafos para realizar los análisis, lo que le convierte en uno de los mejores softwares para realizar la tarea, a continuación, se muestran unos ejemplos Figuras 9, 10 y 11.

Diagrama de Conway

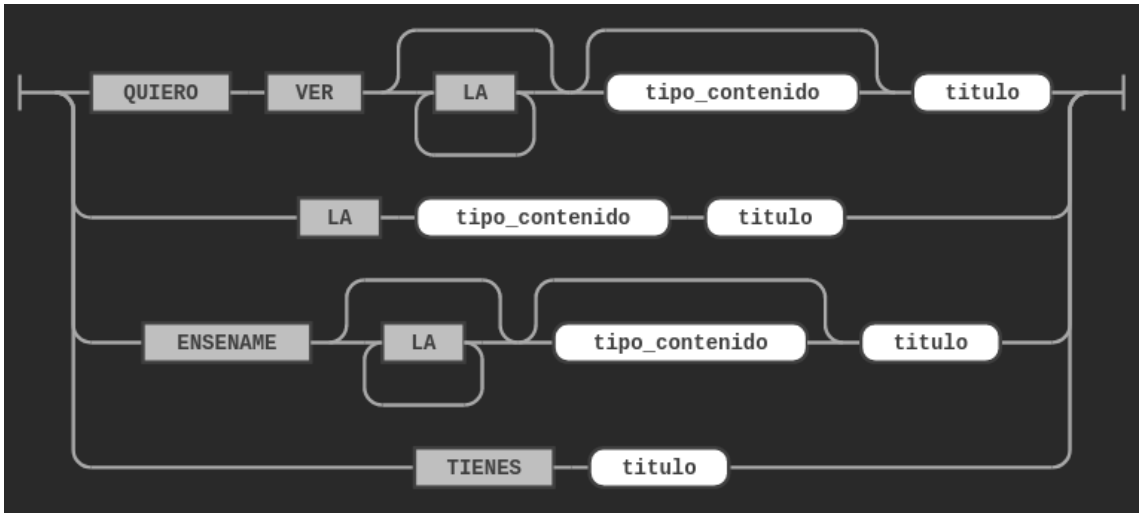


Figura 9: Diagrama de Conway

Red de transición aumentada

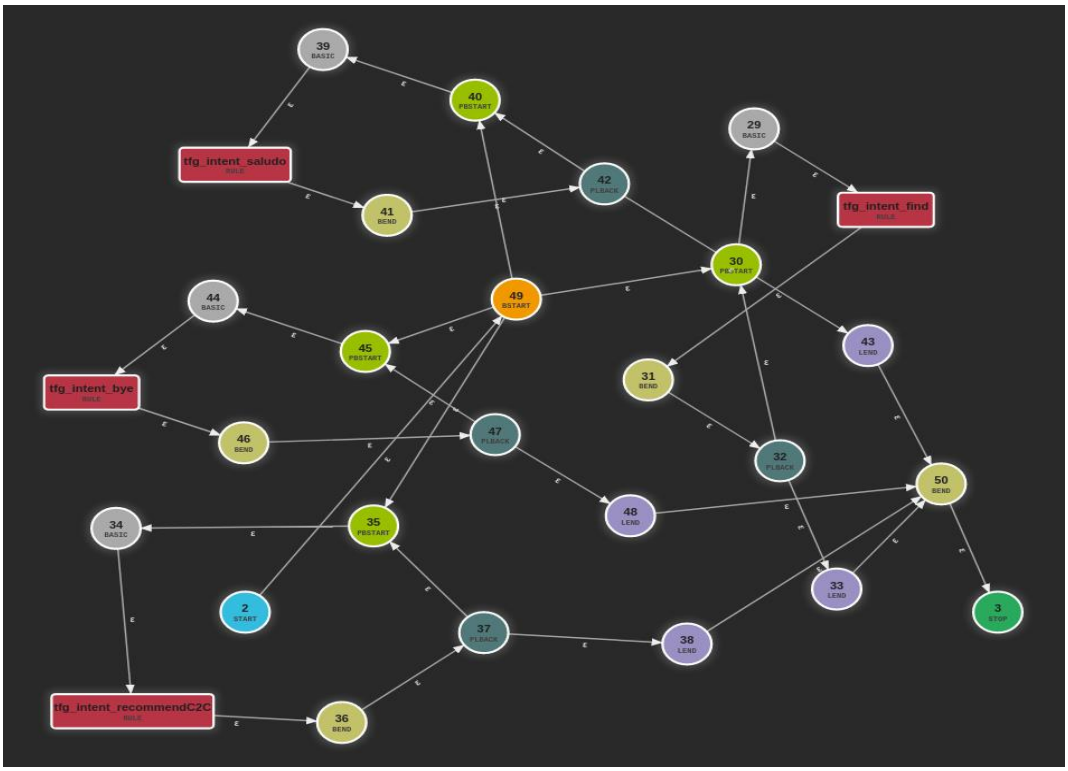


Figura 10: Red de transición aumentada

Dendograma

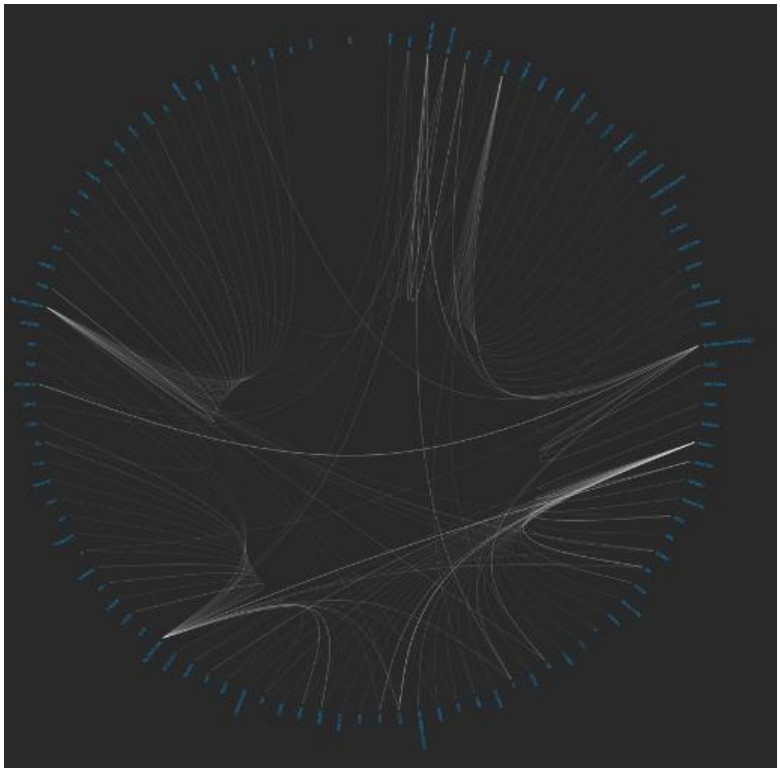


Figura 11: Dendograma

El potencial de los dendogramas es que recogen de manera muy eficiente las relaciones jerárquicas.

2.4 Recomendador

Antes de diseñar el recomendador se deben extraer los datos sobre los que se va a trabajar y revisarlos. Para este proyecto se ha usado un dataset de MovieLens [26] Dicho dataset consta de 26.000.000 ratings y 750.000 tags aplicados a 45.000 películas por 270.000 usuarios e información adicional sobre las películas como: los géneros que la definen, un resumen, la fecha de estreno, el lenguaje original, actores, directores y lo que recaudó.

Dicho dataset se encuentra en inglés, y aunque los algoritmos aplicados sean independientes del idioma, es conveniente eliminar las palabras vacías, ya que estas palabras sí que son dependientes del idioma y pueden mejorar mucho los modelos.

A la hora de diseñar el recomendador se tuvieron en cuenta varios aspectos:

- Es necesario hacer un recomendador basado en géneros y otro basado en otras películas
- El recomendador basado en otras películas se basará en:
    - Resumen de la película
    - Basado en palabras clave, actores y directores
    - Ordenar las recomendaciones en base a las puntuaciones que obtuvo cada película.

2.4.1 Recomendador de Géneros

Para implementar el recomendador basado en género, primero es necesario aplicar una función de peso a las películas para tener una ratio de cuan buena es una película, para ello a todas las películas se le aplicará una variación de la fórmula de la media ponderada Ecuación 1, en la que se tiene en cuenta el número de votos de la película ponderado a la valoración media y el mínimo de votos, ponderados por la media de votos general. Se aplican estas ponderaciones para que no exista un desbalanceo entre una película que cuente con un número elevado de votos y una que cuente con un número reducido de votos.

$$ratio\ puntuación = \left(\frac{v}{v+m} * R\right) + \left(\frac{m}{v+m} * C\right)$$

Ecuación 1: Función de peso ponderado a las valoraciones de películas

Donde:

- v: es el número de votos de la película
- m: es el mínimo de votos requerido
    - por ejemplo, si esta variable toma el valor 0.8, la película tiene que tener al menos más del 80% de votos que las demás.
- R: es la valoración media
- C: es la media de votos general

Tras haber aplicado la Ecuación 1 y para un género dado, se escogerán películas cuyo genero principal sea el género por el que se desea recomendar y que tengan mayor ratio de puntuación.

2.4.2 Recomendador Basado en el resumen de las películas

Como el resumen de las películas está formado por un conjunto de oraciones hay que procesar esas palabras para convertirlas en datos numéricos significativos para hacer cálculos. Para ello hay que hallar la representación en vectores del texto, que se puede expresar aplicando la formula TF-IDF (Term Frequency - Inverse Document Frequency), formula que hace una relación entre la frecuencia de aparición de un término en el texto con la frecuencia de aparición del término en todo el conjunto de textos, como se muestra en la Ecuación 2. [27]

$$TFIDF = TF(t,d) * IDF(t,D)$$

$$TF(t,d) = \frac{t}{d} \qquad TF(t,d) = \log \frac{D}{1 + \{d \in D : t \in d\}}$$

Ecuación 2: Frecuencia de término – frecuencia inversa de documento

Siendo:

- t: el termino
- d: el texto
- D: número total de textos
- $\{d \in D : t \in d\}$: numero de documentos donde aparece el termino t

Una vez aplicada la formula y teniendo vectorizados los resúmenes se aplica la similitud coseno entre los vectores, para calcular cuantitativamente cual es la similitud entre películas. La fórmula de la similitud coseno se recoge en la Ecuación 3:

$$similitud = \cos(\theta) = \frac{A * B}{||A|| \cdot ||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A^2{}_i}\ \sqrt{\sum_{i=1}^{n} B^2{}_i}}$$

Ecuación 3: Similitud coseno

Habiendo calculado la similitud coseno de una película con las demás, ya se tienen todos los datos para dar un resultado, las películas con menor valor de similitud serán las más afines a la película sobre la que se quiere hacer la recomendación ya que se encontrarán más cerca en el espacio vectorial.

2.4.3 Recomendador basado en palabras clave actores y directores

Se utiliza la misma aproximación que en el apartado anterior, pero usando los datos de: palabras clave, géneros, actores y directores.

2.4.4 Orden basado en las puntuaciones

Para ordenar los resultados según las puntuaciones se utilizan las mismas formulas planteadas en el apartado 2.4.1.

2.5 Microsoft Bot Framework

Microsoft Bot Framework es el Framework que se ha elegido para la construcción del Bot, ya que es nativo en Cloud. Microsoft provee un servicio en la nube de Azure, llamado **Bot Service** que es una implementación de este Framework.

Este servicio está completamente integrado con las principales plataformas de mensajería hoy día, como pueden ser el chat de Skype o Facebook Messenger, además de poder integrarse con asistentes virtuales como Cortana.

Se ha elegido este Framework en concreto porque ofrece muchas facilidades a la hora de desarrollar un ChatBot ya que cuenta con funciones preparadas para conectarse a reconocedores como los vistos en el apartado 2.3.

2.6 Servicios REST

Para la comunicación entre los distintos módulos se han implementado APIS REST para el intercambio de información. De entre todas las operaciones que ofrece REST solo se han usado **GET** y **POST.**

El servicio REST (Representational State Transfer) es una arquitectura para proporcionar estándares de comunicación entre sistemas informáticos, que separa el Cliente del Servidor haciendo que, mientras la especificación no cambie, el código del lado del cliente y del servidor puedan ser modificados indistintamente. La información se

transmite en diferentes formatos como pueden ser XML o JSON, en este caso particular se usa JSON como formato de intercambio de mensajes.

Este tipo de arquitectura se ha escogido porque ofrece un alto grado de eficacia y seguridad a la hora de intercambiar mensajes.

Para definir la especificación de cada servicio se ha usado Swagger, un framework opensource que permite construir y documentar servicios REST.

2.7 Servicios Docker

La idea detrás de Docker es crear contenedores ligeros y portables para las aplicaciones software que puedan ejecutarse en cualquier máquina con Docker instalado, independientemente del sistema operativo que la máquina tenga por debajo, facilitando así también los despliegues. Es autocontenido y se inspira en la idea de máquina virtual.

La generación de estos contenedores se hace mediante la especificación de un "Dockerfile", un archivo en formato yaml en el que se especifica de que imagen base se quiere partir y que comandos se deben ejecutar en esta máquina. Al ser contenedores estancos, se ofrece una capa de aislamiento extra que previene de errores, ya que si el contenedor está bien definido dispondrá de todo el software y las librerías necesarias para su ejecución y que aporta seguridad extra ya que las conexiones entre contenedores no se pueden hacer directamente a no ser que se especifique lo contrario.

Además, permite definir ficheros "Docker-compose" que también están en formato yaml que permiten generar una pila de despliegue, que es capaz de desplegar todos los contenedores, las interconexiones y los volúmenes de datos automáticamente.

Cada módulo que ha sido necesario implementar en este proyecto se ha hecho bajo la filosofía de contenedores de Docker y todos ellos están incluidos en una pila de despliegue automático de "Docker-compose".

Además, usando el software portainer.io, se puede tener una monitorización en tiempo real, del uso de los recursos físicos del servidor y de los recursos de cada contenedor por separado en dashboards fácilmente accesibles.

3. BOT

El objetivo de este apartado es describir de una manera extensa el desarrollo del Bot, haciendo hincapié en los distintos módulos necesarios para su correcto funcionamiento, para ello se pasará por las fases de: especificación de requisitos, en donde se expondrán todos ellos; la arquitectura del sistema junto con los casos de uso y diagramas de secuencia y de flujo.

3.1 Especificación de requisitos

A continuación, se describirán los diferentes requisitos que deberá cumplir el sistema. Los requisitos quedarán diferenciados en 2 clases: funcionales y no funcionales.

Los requisitos funcionales son aquellos que definen las funcionalidades que prestará el sistema. Mientras que los no funcionales son aquellos que no se refieren directamente a las funciones específicas suministradas por el sistema. [28]

3.1.1 Requisitos funcionales

Para mayor claridad a la hora de exponer los requisitos se hará uso de tablas con una plantilla que constará de:

- Identificador: Identificador univoco de cada requisito
- Nombre: Nombre del requisito que debe ser autoexplicativo
- Descripción: Descripción extensa del requisito
- Relación: Si el requisito tiene relación con algún otro, deberá estar indicado con cual en este apartad

Identificador	RF-01
Nombre	Caja de texto
Descripción	En la pantalla principal del ChatBot deberá haber una caja de texto editable que permita al usuario introducir el texto para ser enviado al Bot.
Relación	-

Tabla 1: RF-01

Identificador	RF-02
Nombre	Botón de Enviar
Descripción	En la pantalla principal del ChatBot deberá haber un botón para enviar el texto escrito en la caja de texto.
Relación	RF-01

Tabla 2: RF-02

Identificador	RF-03
Nombre	Botón de introducir texto por voz
Descripción	En la pantalla principal del ChatBot deberá haber un botón que permita activar el micrófono para que el usuario pueda usar su voz para introducir el texto.
Relación	RF-01

Tabla 3: RF-03

Identificador	RF-04
Nombre	Texto enviado por el usuario
Descripción	En la pantalla de conversación con el ChatBot deberán aparecer los mensajes enviados por el usuario en una caja de texto no editable.
Relación	-

Tabla 4: RF-04

Identificador	RF-05
Nombre	Texto enviado por el Bot
Descripción	En la pantalla de conversación con el ChatBot deberán aparecer los mensajes enviados por el Bot en una caja de texto no editable.
Relación	-

Tabla 5: RF-05

Identificador	RF-06
Nombre	Texto enviado por el Bot en tarjetas
Descripción	En la pantalla de conversación con el ChatBot deberán aparecer las tarjetas enviadas por el Bot en un cuadro de dialogo no editable.
Relación	-

Tabla 6: RF-06

Identificador	RF-07
Nombre	Plantilla de tarjeta
Descripción	Si el Bot debe construir una tarjeta lo hara siguiendo la siguiente plantilla: • Imagen • Titulo • Botón de más información
Relación	RF-06

Tabla 7: RF-07

Identificador	RF-08
Nombre	Retorno a pantalla principal desde más información
Descripción	Desde la pantalla de más información se podrá volver a la pantalla principal usando tanto el botón “atrás” del móvil o el botón de la misma pantalla.
Relación	RF-09

Tabla 8: RF-08

Identificador	RF-09
Nombre	Botón de atrás en la pantalla de más información
Descripción	En la pantalla de más información deberá haber un botón para volver a la pantalla principal.
Relación	RF-08

Tabla 9: RF-09

3.1.2 Requisitos No funcionales

Este apartado contiene la información de los requisitos no funcionales, es decir, aquellos relacionados con la compatibilidad y la seguridad.

Para mostrar la información se seguirá la misma plantilla que la de los requisitos funcionales a excepción del campo *relación*.

Identificador	RNF-01
Nombre	Sistema Operativo
Descripción	La aplicación deberá funcionar en cualquier dispositivo Android con una versión de Android superior a la 4.

Tabla 10: RNF-01

Identificador	RNF-02
Nombre	Conexión a internet
Descripción	El dispositivo que ejecute la aplicación deberá contar con una conexión a internet.

Tabla 11: RNF-02

Identificador	RNF-03
Nombre	Usabilidad y estilo
Descripción	La aplicación deberá seguir las buenas prácticas de la guía de Material Design.

Tabla 12: RNF-03

Identificador	RNF-04
Nombre	Rendimiento Aplicación
Descripción	La aplicación deberá responder a los mensajes en menos de 4 segundos.

Tabla 13: RNF-04

Identificador	RNF-05
Nombre	Seguridad
Descripción	La aplicación deberá usar los protocolos seguros HTTPS en la transferencia de mensajes.

Tabla 14: RNF-05

3.2 Diagramas y casos de uso

En este proyecto se tienen en cuenta un total de 4 casos de uso: Saludo, despedida, búsqueda de un contenido y recomendación.

Si el Bot detecte que la intencionalidad del usuario es saludar o despedirse el Bot guardara esa información de contexto y responderá con una frase acorde al contexto indicado, sin embargo, en los casos de uso de búsqueda de información o recomendación el Bot deberá procesar más información y el flujo será más complejo, por eso se adjuntan 2 diagramas de Flujo que se pueden encontrar a continuación.

En cuanto a la búsqueda de contenido el usuario realizará la petición mediante la aplicación y el Bot utilizará los módulos de comprensión de texto para conocer la intención del usuario y posteriormente hará uso del módulo cognitivo para conocer con precisión sobre que elemento se está realizando la petición, para finalmente buscar la información en la base de datos y mostrarlo de nuevo en la aplicación justo debajo del mensaje del usuario, el flujo quedaría entonces así:

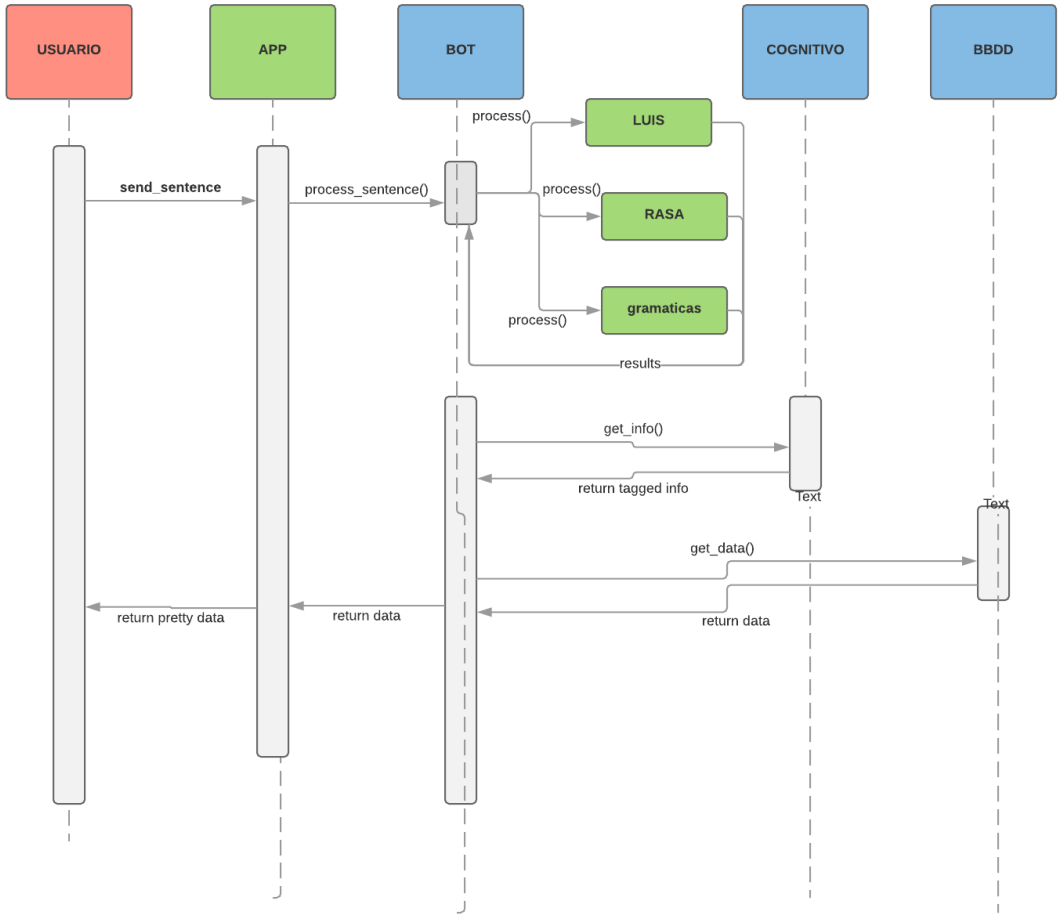


Figura 12: Diagrama de secuencia de búsqueda de contenido

El caso de uso de recomendación es similar, la única diferencia es que tras conocer el elemento por el que se realiza la petición debe consultar al módulo de recomendación que datos debe mostrarle al usuario, se puede apreciar el flujo en el siguiente diagrama:

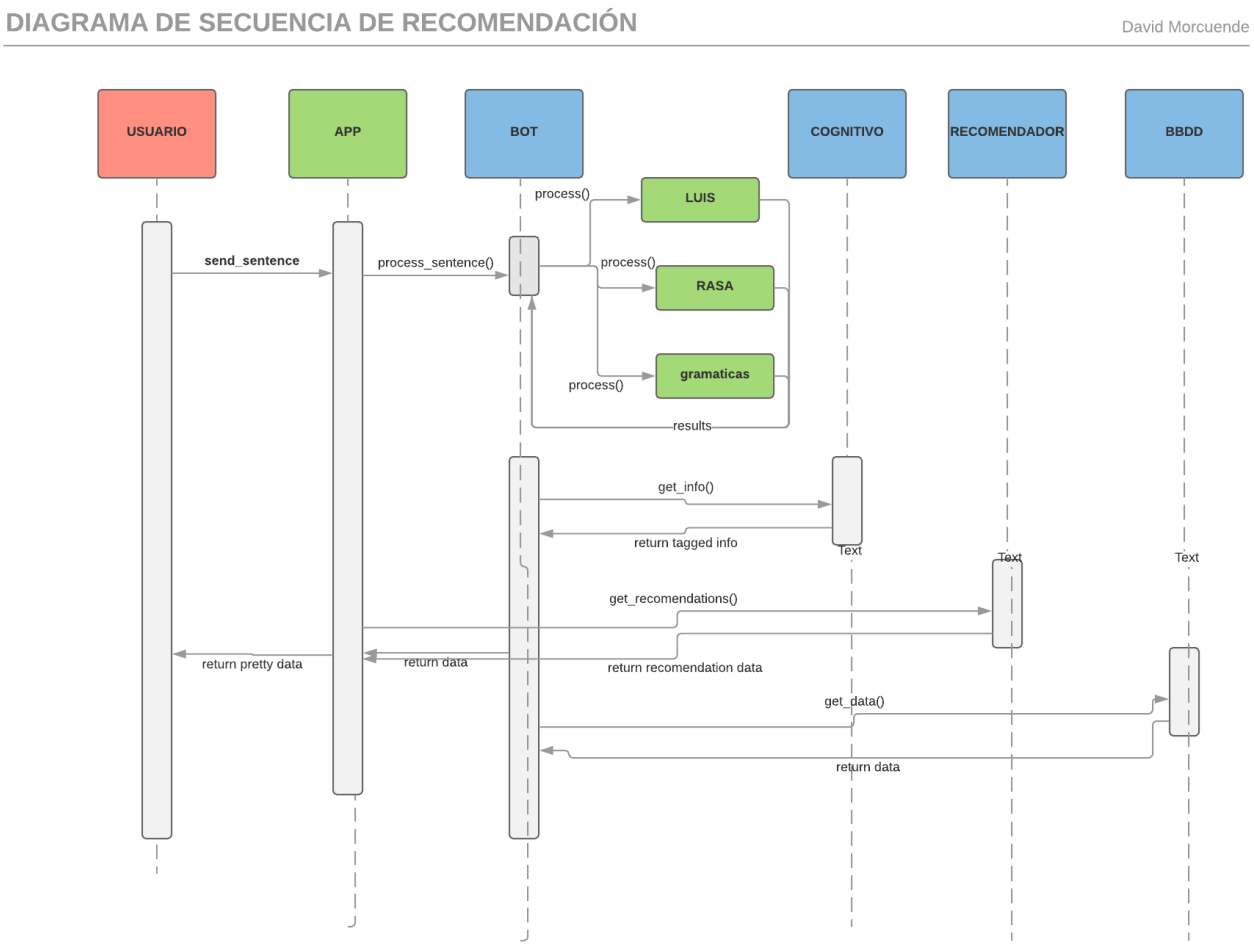


Figura 13: Diagrama de secuencia de recomendación

3.2.1 Diagrama de Flujo

Habiéndose definido los casos de uso y los diagramas de secuencia, en la Figura 14 se expone un diagrama de flujo de la aplicación, mostrando en cada caso, las diferentes vistas que genera la aplicación para cada caso de uso en particular, viéndose las variaciones, como, por ejemplo, para el caso de uso de buscar un contenido, que es posible encontrar el contenido exactamente (se muestra ese contenido) o la búsqueda arroja más de un contenido (se muestra un carrusel de contenidos).

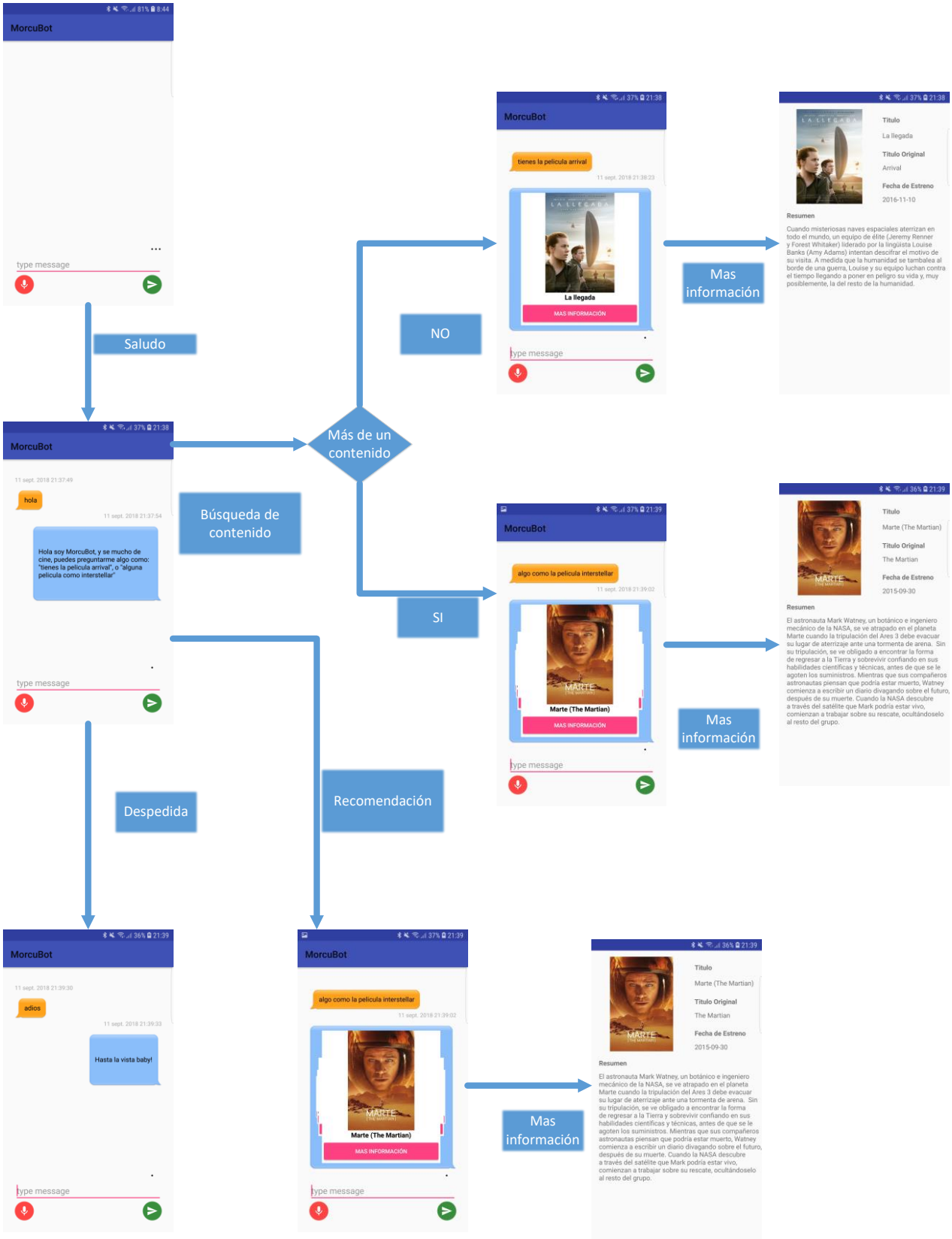


Figura 14: Diagrama de Flujo

3.3 Arquitectura del sistema

Este apartado desarrollará la arquitectura del sistema de forma general, indicando las dependencias entre los módulos y una explicación de cada módulo por separado, indicando cuál es su función y como ha sido desarrollado.

En la siguiente figura se muestra la arquitectura de todo el sistema, indicando con flechas las relaciones de cada módulo, si la flecha es de un solo sentido, significa que la relación es de uso, sin embargo, si es bidireccional significa que hay una dependencia porque las 2 se usan mutuamente.

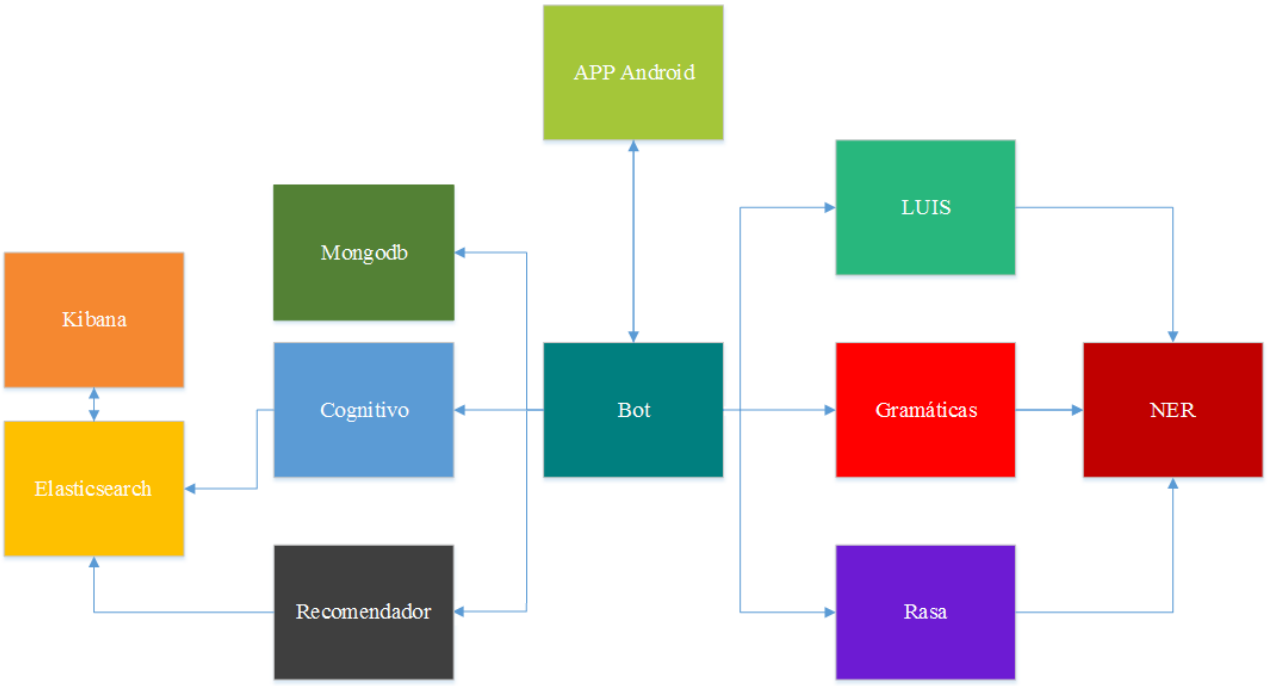


Figura 15: Arquitectura del sistema

Como ya se ha explicado anteriormente, el proyecto versa sobre un ChatBot, cuya entrada vendrá dada por la interacción con la aplicación Android. Tras recibir la entrada del usuario el Bot procesará la entrada usando LUIS, Rasa y gramáticas. Cuando la entrada ha sido procesada el sistema cognitivo extrae la información relevante y la devuelve al Bot que si es necesario hará uso del módulo de recomendación y sino usará la base de datos para proporcionar la información de vuelta a la aplicación Android para que la muestre al usuario procesando las vistas como sea oportuno para mostrarle la información de forma amigable al usuario.

A continuación, se explicará cada módulo por separado.

3.3.1 App Android

La aplicación Android como ya se ha mencionado con anterioridad es el punto de comunicación del usuario con el Bot. Este módulo cuenta con una interfaz gráfica que el usuario usará para comunicarse.

La primera vista que el usuario verá es una pantalla como la de la Figura 17 en la que se muestra un chat vacío y en la parte de abajo una caja de texto editable junto con dos botones, uno de enviar y otro de activación del micrófono.

Tanto si el usuario escribe algo y lo envía como si usa la voz para generar un mensaje se desencadenará el flujo definido en *3.2.1 Diagrama de Flujo*.

A la hora de registrar los eventos disparados por el Bot se hace uso de un Bus de eventos, en vez del uso de los interfaces de Listeners propios de Android. Se elige este modo ya que permite evitar dependencias cruzadas, por lo que la aplicación está más desacoplada, evitar los errores en cuanto al sincronismo de las llamadas en tiempo de ejecución y favorecer los test unitarios [29].

Un bus de eventos es un servicio en el que se registran eventos y los suscriptores de ese evento reciben información frente a cambios. La figura 16 describe como un evento es registrado en el bus y la información les llega a todos aquellos suscriptores que requieren la información de ese evento concreto. Un suscriptor puede estar suscrito a varios eventos e incluso puede registrar eventos y estar suscritos a otros.

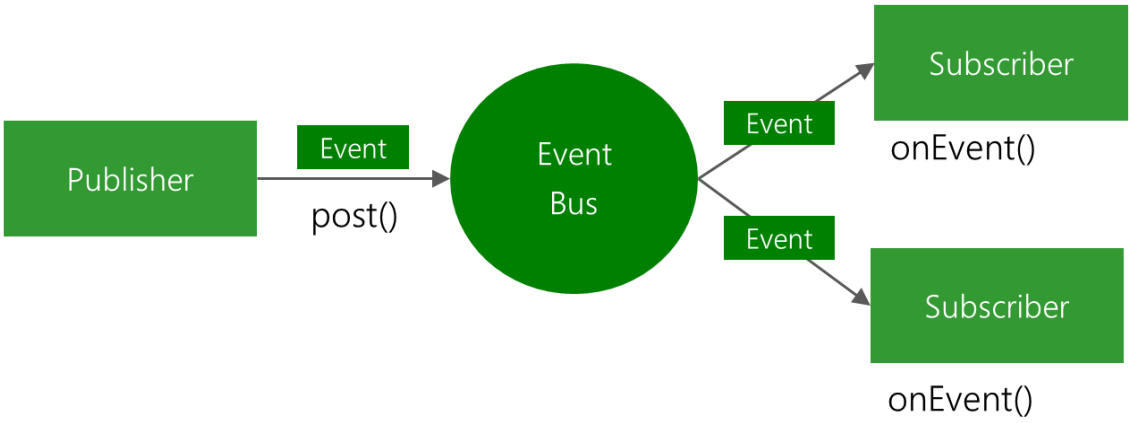


Figura 16: Bus de eventos [30]

Tras procesar todos los eventos la comunicación con el Bot se realiza mediante un canal de Direct Line, canal proporcionado por el Framework de Microsoft.

3.3.1.1 Vistas

A continuación, se definen y muestran las diferentes vistas construidas para mostrar la información procesada al usuario, siempre siguiendo las heurísticas de Nielsen y las buenas prácticas de material Design.

- Inicio: pantalla inicial de comunicación con el usuario donde se listarán las tarjetas Figura 17, tanto de información de entrada, como de salida.

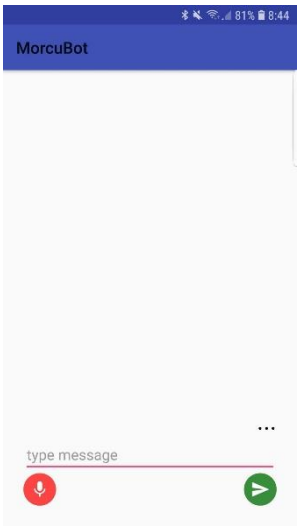


Figura 17: Pantalla inicial

- Mensaje de texto: si solo aparece como respuesta un mensaje de texto, la tarjeta aparecerá como la de la Figura 18.

Figura 18: Pantalla con texto

- Un único contenido: si la respuesta es una tarjeta con un único contenido se mostrará como en la Figura 19.

Figura 19: Pantalla con un único contenido

- Más de un contenido: si la respuesta consta de más de un contenido se mostrará un carrusel como el de la figura 20.

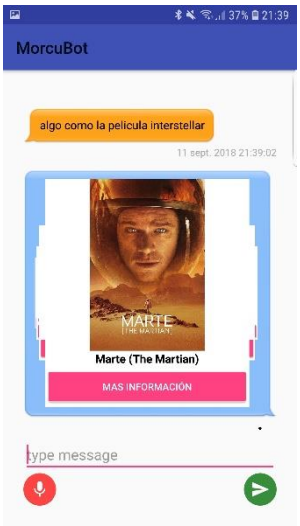


Figura 20: Pantalla con varios contenidos

- Si se pulsa sobre el botón de más información sobre un contenido la pantalla será similar a la de la Figura 21.

Figura 21: Pantalla de más detalles

3.3.2 Bot

El Bot es el módulo central que se encarga del procesamiento de todos los datos haciendo uso de los demás módulos. Su entrada será la entrada del usuario procesada por la aplicación Android, tras procesarla hará uso de los módulos necesarios para generar una salida correcta. En este apartado se verá como es este proceso, en la Figura 22 se muestra la arquitectura del Bot.

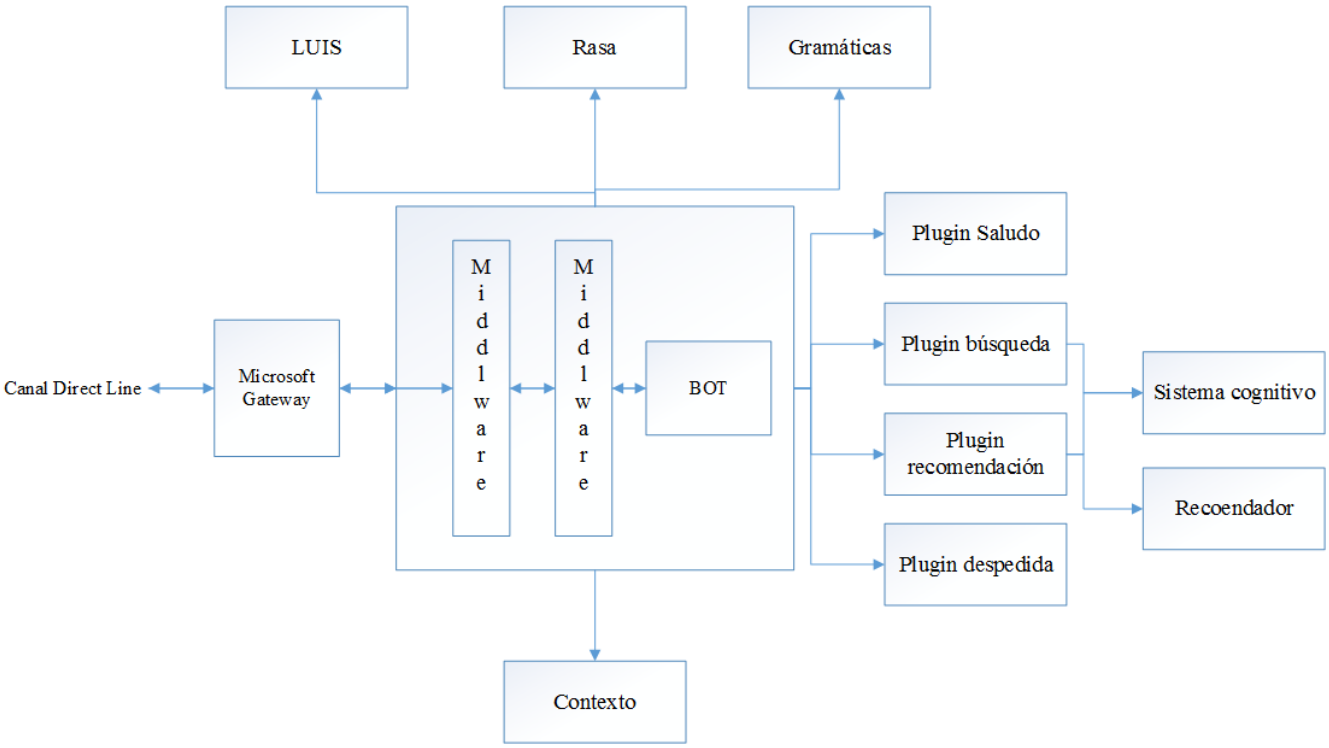


Figura 22: Arquitectura del Bot

La entrada del Bot viene dada, como ya se ha dicho, por la aplicación Android y esto se hace mediante un canal de DirectLine, un canal proporcionado por el Framework de Microsoft que permite la autenticación de las peticiones mediante una clave secreta univoca [31].

La entrada se verifica por razones de seguridad contra la puerta de entrada de Microsoft para comprobar que los mensajes provienen de una fuente confiable.

Si la comunicación es verificada como segura el mensaje tendrá permiso para proseguir con el procesamiento, que en este caso será efectuada por los Middlewares que

preprocesará la entrada, en caso de ser necesario, para que el formato cumpla con los requisitos del Bot.

Cuando la entrada llega al Bot, lo primero que hace es enviar el mensaje a los módulos de Procesamiento de lenguaje natural: LUIS, Rasa y gramáticas. Estos módulos devolverán 3 argumentos principales: intención, entidades y puntuación.

- Intención: intencionalidad del usuario al formular la oración. Normalmente la intención está estrechamente unida al verbo de la oración ya que indica la acción esperada. En la Tabla 15 se pueden observar todas las intenciones admitidas.
- Entidades: palabras clave que permiten la comprensión del contexto de la intención. En la Tabla 16 se pueden observar todas las entidades admitidas.
- Puntuación: Calificación con la que el modelo expresa cuanta confianza tiene en la respuesta que ha proporcionado.
    - La puntuación es una métrica que se añade ya que al haber varios modelos ejecutándose en paralelo es necesario contar con métricas para asegurarse que se toma como referencia el modelo que mejor confianza en sí mismo tenga.

Intención	Descripción
Saludo	La intencionalidad del usuario es saludar al Bot, o iniciar la conversación con el mismo.
Despedida	La intencionalidad del usuario es despedirse del Bot, o finalizar la conversación con el mismo.
Búsqueda	La intencionalidad del usuario es realizar una búsqueda de contenido.
Recomendación	La intencionalidad del usuario es recibir una recomendación.

Tabla 15: Intenciones

Entidad	Descripción
Tipo de contenido	Hace referencia al tipo de contenido que el usuario está esperando: películas, series, cortos, etc.
Título	Hace referencia al título de un contenido.
Género	Hace referencia al conjunto de géneros que el usuario está buscando.

Tabla 16: Entidades

Según la intención determinada por los módulos de procesamiento del lenguaje natural el Bot redireccionará la información a un plugin u otro. Cada plugin tiene un conjunto de funciones para generar la información correspondiente para mostrársela al usuario. Los plugins de saludo y despedida solo guardan la información y le devuelven un mensaje de saludo o despedida respectivamente, sin embargo, para los de Búsqueda y recomendación se debe recuperar información y procesarla, a continuación, se explican estos plugins más detalladamente.

Para los dos plugins mencionados, es necesario que el sistema cognitivo procese la frase, para determinar a qué contenido ha pretendido hacer referencia el usuario cuando ha expresado un título o un género (el módulo cognitivo será descrito en el apartado 3.3.9), cuando el módulo de respuesta se sabrá con certeza a que contenido, o grupo de contenidos se está haciendo referencia.

Cuando ya se tiene la información necesaria se hará una petición a la base de datos para recuperar la información de: título, titulo original, imagen de portada, fecha de estreno y resumen.

Si la intención del usuario ha sido de recomendación con la información recuperada se hace una petición a módulo de recomendación para que resuelva los contenidos y posteriormente se recupera la información en la base de datos.

Cuando ya se la información para darle respuesta al usuario, los middlewares generan una estructura correcta y se la envían de vuelta a la aplicación Android para que la muestren de manera atractiva al usuario.

3.3.3 NER

Reconocedor de entidades nombradas (Named Entity Recognizer) es una tarea de extracción de información, en la que se etiquetan palabras claves (entidades) en un texto.

En este caso particular se usa la aproximación del grupo de procesamiento del lenguaje natural de la Universidad de Stanford, que propone un clasificador basado en modelos de secuencia de campo aleatorio condicional de cadena lineal [32].

La ventaja que ofrecen estos clasificadores es que el entrenamiento es un entrenamiento supervisado cuyos modelos se forman simplemente etiquetando las entidades a ser clasificadas en los propios datos, generando modelos de secuencia para cualquier tipo de clasificador.

En esta implementación los modelos se forman con ficheros *tsv* (ficheros separados por tabulaciones) en los que se identifica la entidad con una etiqueta y las palabras que no son entidades con una O mayúscula, así un ejemplo podría ser.

Palabra1	O
Palabra2	O
Entidad1	ENTIDAD
Palabra3	O

Los modelos formados se entrenan con el clasificador mencionado anteriormente y se usan como apoyo a los módulos de procesamiento del lenguaje natural para que el reconocimiento de entidades tenga una mejor confianza.

3.3.4 LUIS

LUIS es un servicio de comprensión del lenguaje de Micosoft, basado en estadísticas, pero no deja de ser un servicio de aprendizaje automático supervisado, lo que quiere decir que ha de ser entrenado con modelos previamente etiquetados.

Como se ha descrito en el apartado anterior el Bot cuenta con cuatro intenciones y tres tipos de entidades, por lo que al definir el modelo es necesario indicar para cada frase de entrenamiento a que intención hace referencia y que entidades componen la frase, especificando en que posiciones se encuentran dichas entidades y qué entidad es.

La construcción de estos modelos puede realizarse de dos maneras: mediante interfaz gráfica, o mediante la composición de un fichero en formato JSON.

En la Figura 23 se puede apreciar que se está añadiendo una nueva frase al corpus de la intención de búsqueda de un contenido en la que se han etiquetado una entidad de tipo contenido y otra de título, respectivamente.

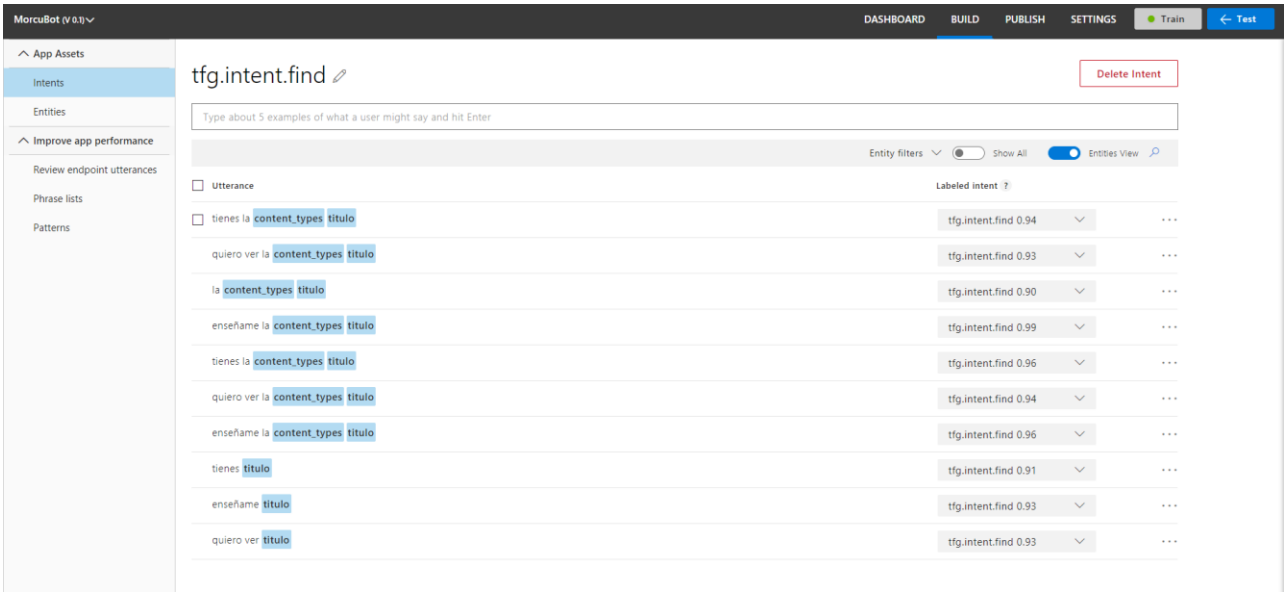


Figura 23: Añadir frase a LUIS

Sin embargo, la misma frase añadida en el modelo en formato JSON tendría el aspecto de la Figura 24:

```
{
      "text": "tienes la pelicula arrival",
      "intent": "tfg.intent.find",
      "entities": [
        {
          "entity": "content_types",
          "startPos": 10,
          "endPos": 17
        },
        {
          "entity": "titulo",
          "startPos": 19,
          "endPos": 25
        }
      ]
    },
```

Figura 24: Frase en modelo de LUIS en formato JSON

A la hora de generar los modelos se han utilizado dos aproximaciones, entrenando el sistema con cadenas de texto inalteradas (a excepción de una limpieza de caracteres como los símbolos de puntación), y el mismo modelo, pero sustituyendo las entidades por entidades marcadas después de haber sido procesadas por NER. Se evaluarán ambos modelos en el Punto 4.

Tras haber entrenado los modelos, el algoritmo entrenado se invoca mediante el uso de servicios REST, dando como resultado una estructura de datos en formato JSON que indica que intención es la más probable junto con las entidades que ha detectado, esta respuesta es la que utiliza el Bot para tomar decisiones con respecto a que decisiones tiene que tomar.

3.3.5 Rasa

Rasa al igual que LUIS es un servicio de comprensión del lenguaje natural, pero, a diferencia de LUIS es de código abierto y funciona en base a redes de neuronas.

La generación de modelos es idéntica a la generación de modelos JSON de LUIS, Rasa, sin embargo, no cuenta con una interfaz gráfica nativa, de ser necesaria habría que contar con software de terceros.

Las llamadas a las funciones de Rasa también se realizan usando REST que devuelven los resultados en estructuras en formato JSON.

3.3.6 Gramáticas

Una aproximación de comprensión del lenguaje distinta a las propuestas por LUIS y Rasa son los analizadores sintácticos o *parsers*. Un *parser* analiza una cadena de símbolos, en este caso, frases dichas por el usuario, para comprobar si se adecuan a las reglas de una gramática. En caso de adecuarse, se dice que la cadena de texto pertenece a la gramática y por lo tanto que tienen una estructura con algún significado.

Para extraer el significado de la cadena de texto que ha sido reconocida por la gramática se hace uso de un método para recorrer arboles sintácticos conocido como visitador.

Este método se basa en recorrer el árbol sintáctico desde la parte más izquierda de la cadena de texto, comprobando como la cadena puede encajar con los nodos del árbol sintáctico. Este proceso se realiza hasta que se encuentra, si existe, un nodo terminal que encaja perfectamente. Si, además, el terminal es un conjunto de caracteres que aportan

información como pueden ser las palabras claves o entidades (géneros, tipos de contenido y/o títulos) el valor debe ser procesado y guardado, así, la cadena que cumpla las normas sintácticas se definirá como una intención con sus entidades.

El método se conoce como visitador ya que se va “visitando” cada terminal que se procesa para comprobar si encaja con las reglas sintácticas y, si es un no terminal que hace referencia a una entidad, se procesa y se guarda.

Con este método se puede definir una cadena de texto como intención junto con sus entidades como los demás métodos, así todos ellos pueden competir en paralelo y el que proporcione un resultado con mejor puntuación será presumiblemente el que mejor defina la cadena de caracteres.

3.3.7 Elasticsearch

Herramienta que permite indexar un gran volumen de datos para hacer consultas sobre ellos. Las ventajas de esta herramienta residen en que la búsqueda es extremadamente eficiente en grandes cantidades de datos, además de proveer de funciones de búsqueda aproximada.

La función de este módulo es obtener, a partir de la frase del usuario, las entidades de: tipo de contenido, género o título. Para ello en la indexación en esta herramienta se tienen en cuenta varios índices:

- Títulos de películas
- Títulos de series
- Títulos de cortos/otros
- Géneros de películas
- Géneros de series
- Personas

Se indexa de esta forma para poder recuperar la información necesaria de forma eficiente.

Una de las principales ventajas de este sistema es la búsqueda aproximada, ya que las búsquedas se realizan sobre la entrada del usuario, que ha podido cometer algún error tipográfico haciendo que el título, por ejemplo, tenga una o varias letras mal escritas.

3.3.8 Kibana

Módulo de visualización de datos de Elasticsearch. Con este módulo se realiza el mantenimiento de Elasticsearch, como comprobar el estado del servicio, u obtener métricas de los datos.

Por ejemplo, en la Figura 25 se muestra el número de películas en rangos de tiempo de 10 años empezando en 1950.

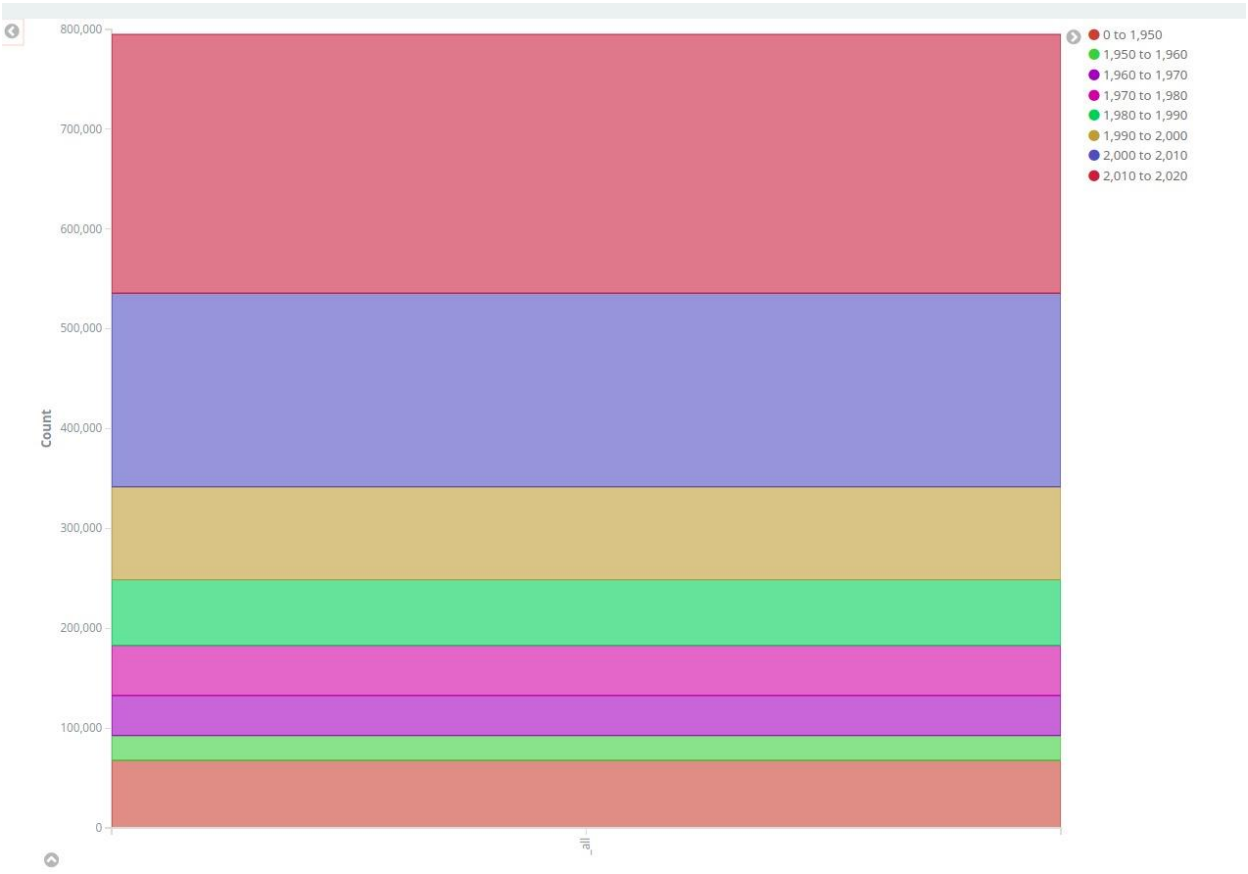


Figura 25: Número de películas entre 1950 y 2020

3.3.9 Cognitivo

El sistema cognitivo es el encargado de procesar la información proveniente del Bot para hacer uso de Elasticsearch para ser capaz de etiquetar correctamente dentro de una frase dicha por el usuario que entidades son tipos de contenido, que entidades son genérenos y que entidades son títulos. Indicando para cada uno en que posiciones de la frase se encuentra.

Este sistema provee de métricas al Bot de la confianza que tiene de esos resultados, también aporta información sobre si el usuario hizo algún error tipográfico y lo corrige para que el error no transcienda a los demás módulos.

3.3.10 Recomendador

Este módulo se compone de un sistema recomendador cuya entrada es el id de una película o un género y genera como salida el id de las películas basadas en la recomendación que se elija. En este caso los tipos de recomendación se clasifican en 2: por género y por películas.

El recomendador de genero elije un conjunto de películas basadas en el género pedido por el usuario y les aplica una función de peso ponderado a la valoración de la crítica.

En cuanto al recomendador basado en las películas, se pueden obtener recomendaciones basadas en el resumen de las películas, en palabras clave actores y directores aplicando la función de ponderación.

Para realizar las consultas por las recomendaciones se utiliza el protocolo HTTP. Como este módulo es una implementación propia, ha sido necesario desarrollar un servidor que provea al cliente de los datos del recomendador, para ello se ha definido un estándar de comunicación usando el Framework de código abierto Swagger. cuya representación gráfica puede verse en la Figura 26.

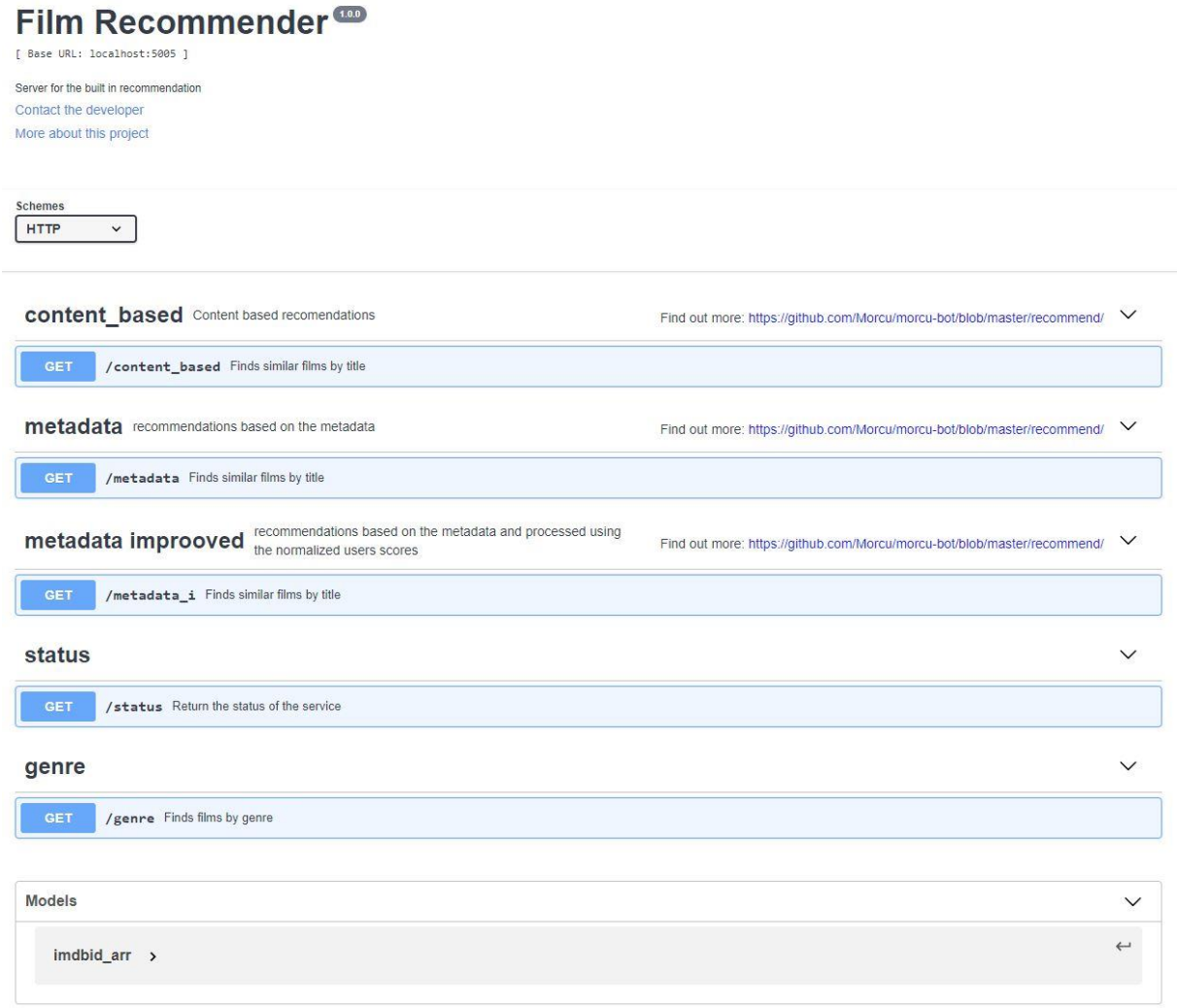


Figura 26: Swagger del sistema recomendador

Se elige el recomendador usando el argumento ya que es una buena forma de relacionar películas ya que el transcurso de los sucesos será similar o al menos de la misma temática.

Intentando mejorar el modelo, se realiza una aproximación que minimice el ruido, para ello, en vez de usar los argumentos de las películas, se intenta relacionar las películas por palabras claves y actores ya que la correlación parece más clara de este modo.

Tras analizar los resultados, no resulta claro que recomendador es mejor que otro ya que las recomendaciones que pueden aportar ambos son coherentes, sin embargo, las recomendaciones son muy subjetivas y dependerán de cada persona en particular.

Sin embargo, realizar un orden ponderado a las puntuaciones de los usuarios hace que de entre todas las recomendaciones coherentes solo se muestren las que más han gustado a los usuarios en primer lugar.

3.3.11 Mongodb

Base de datos NoSQL orientada a documentos. Los datos almacenados en esta base de datos son semi-estructurados, esto quiere decir que no es necesario definir un esquema, aunque si es recomendable.

MongoDB destaca sobre todo en su gran escalabidad horizontal y en la velocidad de resolver consultas [33], además de permitir el uso de funciones de Map Reduce de computación paralela sobre grandes colecciones de datos.

El formato en el que se almacenan los datos es BSON una versión modificada del formato JSON, pero en Binario. Aunque los esquemas y la información puedan ser definida usando JSON.

En la Figura 27 se muestra la composición de la base de datos:

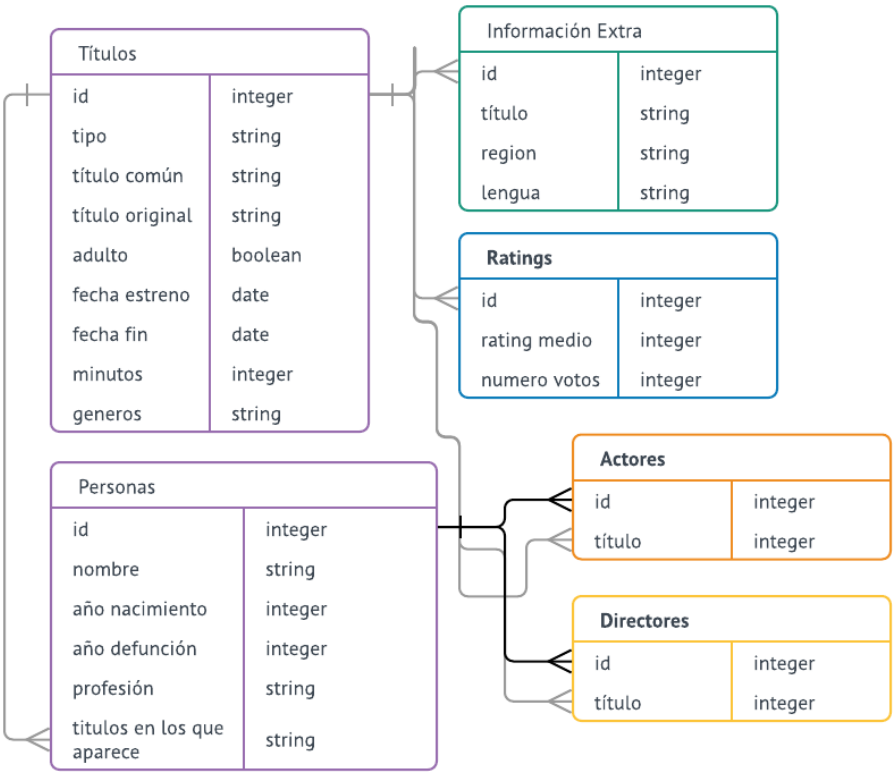


Figura 27: Composición de la BBDD

El uso de la base de datos es recuperar la información por parte del Bot, tras ser procesada por los sistemas cognitivos, para ser mostrada al usuario mediante la aplicación móvil.

3.3.12 Portainer

Para monitorizar todos los sistemas y módulos anteriormente mencionados se hace uso del software libre Portainer. Portainer cuenta con una interfaz de usuario que permite administrar fácilmente y de manera independiente las imágenes virtuales que ejecutan los distintos módulos.

En la Figura 28 se aprecia el Dashboard inicial, donde se puede comprobar el estado del servicio, tanto a nivel de sistema como a nivel de recursos como: memoria usada, volúmenes de datos, volumen de red y volumen de CPU usada.

A parte de únicamente mostrar métricas, también puede interactuar con cada máquina virtual, por separado, haciendo así que el mantenimiento de las mismas sea muy eficiente.

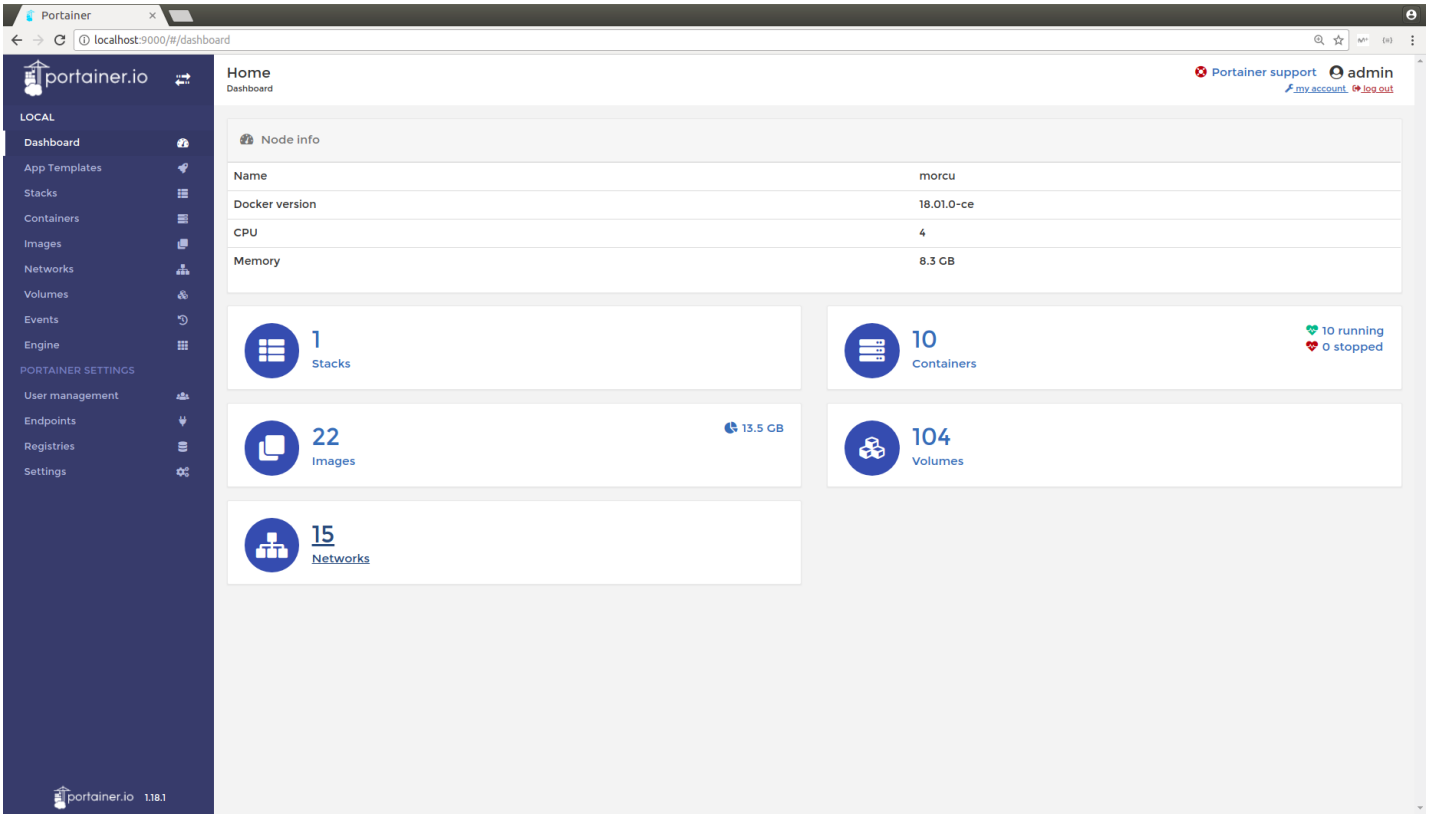


Figura 28: Dashboard principal de portaner

4. EVALUACIÓN

Tras desarrollar e implementar el sistema se pasa ahora a una etapa de evaluación, en la que se comprueba que el desarrollo ha sido correcto, evaluando principalmente los sistemas de procesamiento de lenguaje natural, ya que es una de las partes más críticas del sistema, es la que dictamina la intencionalidad del usuario y la que desencadena las funcionalidades del Bot.

En este apartado se va a realizar un estudio sobre los sistemas de procesamiento de lenguaje natural implementados, primero se evaluará el etiquetador de entidades NER y posteriormente se evaluarán los clasificadores de intenciones y entidades LUIS, Rasa y gramáticas.

Finalmente se presentará una conclusión en base a los resultados obtenidos.

4.1 NER

Para evaluar este módulo se han separado los datos del modelo de entrenamiento en 2 subconjuntos, un subconjunto de entrenamiento y un subconjunto de test. Para que los conjuntos no queden desbalanceados por intención, se separan los datos eligiendo un 20% de entradas de cada intención para generar el conjunto de test, así se puede asegurar que en el conjunto de test se van a evaluar todas las intenciones en la misma proporción.

En la Tabla 17 se muestran los resultados del test.

Entidad	P	R	F1	PR	FP	FN
CONTT	1	1	1	18	0	0
GENRE	1	1	1	9	0	0
TITLE	1	1	1	17	0	0
Total	1	1	1	44	0	0

Tabla 17: Resultados evaluación NER

Donde:

- P es el p-valor: “nivel de significación más pequeño posible que puede escogerse, para el cual todavía se aceptaría la hipótesis alternativa con las observaciones actuales”
- R es el coeficiente de correlación
- F1 es el valor-f: medida de precisión que tiene un test
- PR son los positivos reales
- FP son los falsos positivos
- FN son los falsos negativos

Tras evaluar la tabla 17 se aprecia que el modelo se ajusta perfectamente al contexto siendo capaz de etiquetar perfectamente el 100% de las entradas de test, esto quiere decir que siempre que se ajuste al modelo será capaz de encontrar las entidades correctas.

4.2 LUIS

Para evaluar los modelos de LUIS se utiliza la propia herramienta de Microsoft que permite realizar test en modo batch (procesamiento por lotes), en el anexo se adjuntan las Figuras correspondientes a los resultados de los test, en las Tablas 18 y 19 se muestra un resumen de los resultados de las evaluaciones.

Evaluación de modelo de LUIS **sin** NER

Intención	PR	FP	puntuación
despedida	15	0	1
búsqueda	10	0	1
recomendación	16	0	1
saludo	22	0	1

Tabla 18: Resultados evaluación LUIS sin NER

Evaluación de modelo de LUIS **con** NER

Intención	PR	FP	puntuación
despedida	15	0	1
búsqueda	10	0	1
recomendación	17	0	1
saludo	21	0	1

Tabla 19: Resultados evaluación LUIS con NER

Donde:

- PR son los positivos reales
- FP son los falsos positivos
- Puntuación $= \frac{positivos\ reales}{positivos\ reales + falsos\ positivos}$

Comprobando las puntuaciones los modelos de LUIS se comportan perfectamente incluso con pequeñas variaciones en las frases usando entidades con las que no se ha entrenado y cambiando algunas palabras de género y/o número.

4.3 Rasa

Rasa permite evaluar los modelos de dos formas distintas mediante un conjunto de test etiquetado con la salida esperada para cada entrada o mediante validación cruzada.

4.3.1 Evaluación

Para la evaluación con un conjunto de test, se seguirá la misma forma de evaluar que para LUIS usando incluso los mismos sets de entrenamiento y test. En las Tablas 20 y 21 se muestran los resultados.

Evaluación de modelo de Rasa **sin** NER

Intención	PR	FP	puntuación
despedida	15	0	1
búsqueda	10	0	1
recomendación	16	0	1
saludo	22	0	1

Tabla 20: Resultados evaluación Rasa sin NER

Evaluación de modelo de Rasa **con** NER

Intención	PR	FP	puntuación
despedida	15	0	1
búsqueda	10	0	1
recomendación	17	0	1
saludo	21	0	1

Tabla 21: Resultados evaluación Rasa con NER

Donde:

- PR son los positivos reales
- FP son los falsos positivos
- $\text{Puntuación} = \frac{positivos\ reales}{positivos\ reales + falsos\ positivos}$

4.3.2 Validación cruzada

La validación cruzada es una técnica usada para evaluar los datos garantizando la independencia entre los conjuntos de entrenamiento y test.

La técnica consiste en dividir K veces el mismo conjunto de datos en conjuntos de entrenamiento y test y calcular la media de los resultados.

Por ejemplo, si se utiliza un valor de K igual a tres el conjunto de datos se dividirá en tres, A, B y C, y se realizarán tres iteraciones.

1. Entrenar con conjunto A y B y evaluar con C
2. Entrenar con conjunto A y C y evaluar con B
3. Entrenar con conjunto B y C y evaluar con A

Finalmente se calcula la media de los resultados, dando una estimación muy aproximada del porcentaje de aciertos.

Aplicando validación cruzada a los modelos usando una K igual a diez se obtienen los resultados de las tablas 22 y 23

Evaluación de modelo de Rasa **sin** NER

Evaluación	Exactitud	F1	Precisión
Intención	0.908	0.892	0.907
Entidad	0.867	0.821	0.808

Tabla 22: Resultados validación cruzada Rasa sin NER

Evaluación de modelo de Rasa **con** NER

Evaluación	Exactitud	F1	Precisión
Intención	0.968	0.959	0.958
Entidad	1	1	1

Tabla 23: Resultados validación cruzada Rasa con NER

Donde:

- Evaluación: es el parámetro a evaluar
- Exactitud: distancia a la clasificación correcta
- F1: valor-F, medida de precisión de un test.
- Precisión: distancia entre las entradas de test

Como se puede apreciar en la validación cruzada los resultados de test han arrojado valores algo inferiores, eso es debido a que los conjuntos de entrenamiento y test se escogen de manera aleatoria y es posible que algún conjunto de test tenga valores que no aparezcan en el entrenamiento y el modelo no pueda llegar a generalizarlas.

Aun así, se puede comprobar que los modelos que utilizan el reconocedor de entidades NER generalizan algo mejor que los que no.

4.4 Gramáticas

En cuanto a la evaluación de gramáticas, no es necesario usar un conjunto de test ya que el objetivo de las gramáticas es comprobar si la frase de entrada concuerda con sus reglas, detectando la intención y las entidades. Por lo que no hay métricas de puntuación que indiquen la confianza en el resultado.

El resultado de la gramática es binario, o la frase concuerda con sus reglas y la detecta con intenciones y entidades o la rechaza.

4.5 Análisis comparativo

Tras haber analizado todos los modelos por separado se puede concluir que los modelos de LUIS (estadístico) y Rasa (word2vec) clasifican correctamente en un porcentaje muy alto las intenciones y las entidades, siendo los modelos que cuentan con un reconocedor de entidades nombradas (NER) los que mejor funcionan.

Cabe destacar que los modelos tienen un alto grado de acierto porque el dominio es muy acotado. Tampoco se encuentran diferencias muy notables entre los sistemas utilizados, incluso si los dominios son más grandes y con datos más dispersos, la evaluación de los modelos proporcionados por LUIS y por Rasa en los Chabots obtienen puntuaciones similares como se puede apreciar en *Evaluating Natural Language Understanding Services for Conversational Question Answering Systems* [34]

Por otro lado, las gramáticas siempre obtendrán mejor resultado si la entrada cumple con las reglas de la misma, pero al contrario que los otros sistemas tiene una capacidad casi nula de generalización.

5. GESTIÓN DEL PROYECTO

En este apartado se detalla la planificación del proyecto describiendo los diferentes estadios por los que se ha iterado indicando en cada caso, el flujo de trabajo, cómo se ha trabajado el día a día de las tareas, el control de código y el objetivo de cada fase.

Adicionalmente se expone la planificación temporal, usando como apoyo un diagrama de Gantt para tener una visual gráfica del tiempo de dedicación previsto.

5.1 Planificación

El proyecto ha tenido 4 fases principales: Planificación, Diseño, Implementación y Análisis y Entrega. A continuación, se muestra un diagrama detallado de las fases y tras el diagrama se explicarán minuciosamente cada fase y tarea asociada.

Figura 29: Fases del proyecto

1. **Planificación:** Fase en la que se realiza un estudio de la tecnología y la arquitectura del sistema para tomar las decisiones de diseño e implementación.

1.1 Estudio de las tecnologías: tarea en la que se realizó un estudio sobre las posibles tecnologías que eran susceptibles de ser usadas para la implementación del

sistema, haciendo énfasis en las tecnologías más usadas, más valoradas y más innovadoras.

1.2 Estudio de la arquitectura: Estudio sobre la arquitectura del sistema, donde se definen los módulos necesarios y las dependencias que puedan tener.

1.3 Decisión de las tecnologías: Con el estudio de la arquitectura del sistema y el estudio previo de las tecnologías existentes para la implementación del sistema se hace la decisión final de cuáles son las tecnologías más beneficiosas para el proyecto.

1.4 Asignación del proyecto: Se presenta una idea inicial del sistema, presentando una arquitectura inicial y la tecnología para implementarla, para la validación de la directora del TFG.

1.5 Diseño básico de la arquitectura: Se hace un diseño inicial de la arquitectura definiendo de manera precisa los módulos de los que se compone el sistema, de qué se componen dichos módulos y que relaciones pueden tener intra e inter modulo

1.6 Estudio de la temática: Con la versión inicial de la arquitectura se elige la temática en la que se va a centrar la aplicación, haciendo un estudio de posibles problemas a los que se puede dar solución.

1.7 Análisis de aplicaciones similares: Con el tema central ya definido se realiza un estudio de posibles aplicaciones similares o aplicaciones con la misma temática que este proyecto.

2. **Diseño:** Tras haber tomado todas las decisiones con respecto a la temática, la arquitectura, el diseño y las aplicaciones similares que han podido aportar alguna idea extra, se realiza el diseño de cada módulo de forma precisa.

2.1 Bot: Se realiza el diseño del Bot. La posible entrada y salida y la interconexión de módulos.

2.2 Módulo Cognitivo: Se realiza el diseño del módulo cognitivo y como se va a relacionar con los datos.

2.3 Recomendador. Diseño del recomendador junto con sus posibles variantes y definición en swagger del API-REST

2.4 Aplicación Android. Diseño de la aplicación Android y de su conexión con el Bot.

3. **Implementación**: Habiéndose definido todos los módulos con sus dependencias, flujos de datos y flujos de entrada salida se procede a la implementación de los mismos.

3.3 Procesado y volcado de la información: La etapa inicial es el procesado y volcado la información, para que los sistemas puedan hacer uso de la información, esta, tiene que estar bien procesada, limpia (sin errores de formato o datos vacíos) y almacenada en bases de datos.

3.4 Reconocedores: Se implementa la entrada del Bot. La primera etapa del Bot es la comprensión del lenguaje natural, para saber a qué se refiere el usuario y como ha de actuar.

3.5 Bot: Se implementa el Bot y se conecta con todos los demás módulos ya que es el elemento Core.

3.6 Módulo Cognitivo: Se implementa el módulo que les da sentido a los datos y provee al Bot de información sobre los datos.

3.7 Recomendador: Se implementa el recomendador con todas las posibles variantes.

3.8 Aplicación Android: Se implementa la funcionalidad de entrada y salida de mensajes, así como, la parte visual de la aplicación para que los usuarios interactúen con la misma.

3.9 Pruebas: Se realizan pruebas E2E (End to End) del sistema, estas pruebas consisten en hacer pruebas del sistema completo, comprobando que, dada una entrada, todos los módulos que están implicados la procesan correctamente y generan una respuesta correcta. No se realizan pruebas de cada módulo por separado en esta fase, ya que en las fases de implementación se añaden pruebas simples para demostrar que el módulo funciona correctamente y está bien implementado.

4. **Análisis:** Tras la implementación del sistema y las pruebas, se realizan análisis para comprobar lo bien que funciona el sistema en conjunto.
5. **Preparación de la Entrega:** Tras haber completado satisfactoriamente el sistema y haber realizado análisis de su comportamiento se empieza a escribir la memoria en la que se explica tanto el proceso de estudio, díselo, implementación y análisis.

5.1 Memoria: Se escribe el documento final del proyecto que explica todas las fases por las que pasa el proyecto, además de la tecnología usada, la planificación del proyecto, los costes estimados, el marco regulador, el entorno socio-económico y las conclusiones finales.

5.2 Presentación: Después de escribir la memoria, se realizan unas slides para presentar todo el proyecto (TFG) al tribunal.

Tras definir las fases y las tareas del proyecto, se puede establecer una planificación temporal del mismo, para ello se hará uso de un diagrama de Gantt, en el que se puede distinguir la duración del proyecto en general y en particular de cada fase y subtareas indicando la fecha de inicio y fin.

A continuación, se muestra la planificación temporal de las tareas y después de la planificación se encuentra el diagrama de Gantt en el que se aprecia detalladamente los rangos de fecha de cada tarea y presenta un modo gráfico de ver si alguna tarea se solapa con otra.

Nombre	Fecha de inicio	Fecha de fin
Proyecto1-TFG	2/10/17	16/09/18
Inicio del proyecto	2/10/17	21/01/18
Planificación	22/01/18	21/02/18
Estudio de las tecnologías	22/01/18	23/01/18
Estudio de la arquitectura	24/01/18	26/01/18
Decision de las tecnologías	27/01/18	29/01/18
Asignación del proyecto	13/02/18	13/02/18
Diseño básico de la arquitectura	14/02/18	15/02/18
Estudio de la tematica	16/02/18	17/02/18
Análisis de aplicaciones similares	18/02/18	20/02/18
Análisis	21/02/18	21/02/18
Diseño	22/02/18	30/03/18
Bot	22/02/18	23/02/18
Módulo Cognitivo	24/03/18	26/03/18
recomendador	27/03/18	28/03/18
Aplicación Android	29/03/18	30/03/18
Implementación	31/03/18	7/08/18
Procesado y volcado de la información	31/03/18	29/04/18
Reconocedores	30/04/18	19/05/18
Bot	20/05/18	8/06/18
Módulo Cognitivo	9/06/18	22/06/18
Recomendador	22/06/18	6/07/18
Aplicación Android	7/07/18	1/08/18
Pruebas	1/08/18	7/08/18
Preparación Entrega	8/08/18	16/09/18
Memoria	8/08/18	1/09/18
Presentación	2/09/18	16/09/18

Figura 30: Planificación temporal

Figura 29: Diagrama de Gantt

Como se puede observar, hay tareas que se solapan como pueden ser las tareas de creación del Bot junto a los módulos cognitivos y de recomendación. Estas tareas se solapan porque hay una dependencia de la una en la otra y se requiere una integración continua entre ellas por eso el desarrollo se realiza en paralelo.

Finalmente, y usando los datos del diagrama podemos hacer una estimación de la duración del proyecto que ha sido de 238 días (350 días – 112 días ya que transcurrieron 112 días desde que se asignó el TFG hasta que se empezó a trabajar en él).

5.2 Flujo de trabajo

En el trabajo del día a día se ha utilizado una metodología similar a Scrum, sin aplicar todas las reglas que Scrum propone.

Para controlar las tareas se ha utilizado el software Trello, que permite administrar proyectos por tareas, con una interfaz de tipo Kanban en la que hay diferentes columnas que representan las diferentes fases de las tareas, en este caso particular se han utilizado 3 columnas:

- TODO: Columna dedicada a las tareas que aún están por hacer
- In Progress: Columna en la que se encuentran las tareas que se están haciendo ahora mismo.
- Finished: donde se encuentran las tareas acabadas.

A continuación, se muestra una imagen extraída del tablero que se ha utilizado en este proyecto.

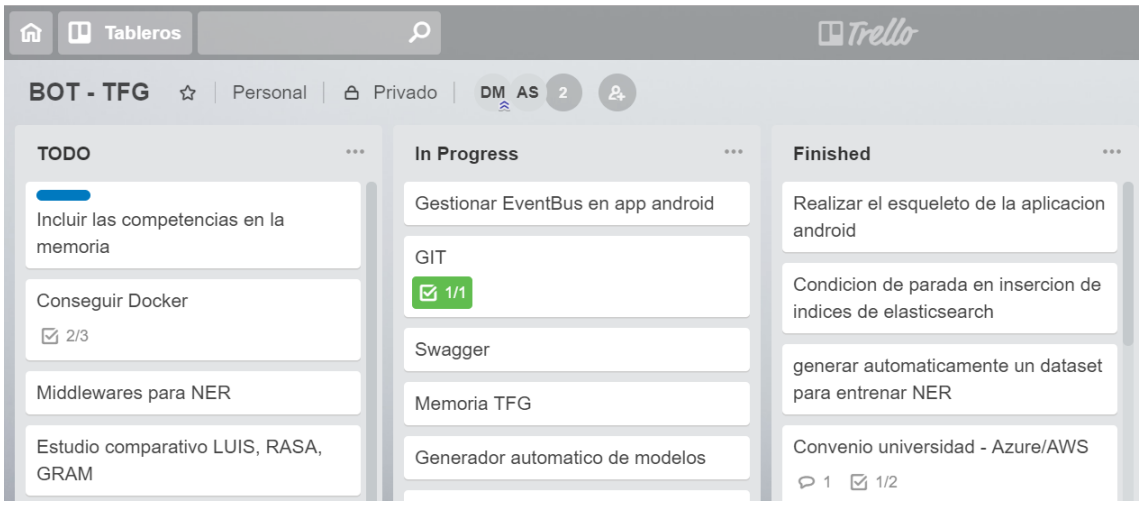


Figura 30: Dashboard principal de Trello

La aproximación a la metodología scrum está relacionada a los desarrollos ágiles, en los que se establecen *sprints* (ciclos de trabajo) que duran entre 2 y 4 semanas, según el proyecto y en el que se definen las tareas que se van a realizar en ese ciclo de trabajo, teniendo en cuenta que tareas se han realizado en el anterior, cuáles son las más indicadas para hacer en el siguiente y que posibles impedimentos se van a encontrar. [35]

Con esta metodología también se permite el trabajo en paralelo en más de una funcionalidad ya que la aproximación no es realizar una funcionalidad tras otra de manera secuencial, sino que, al dividir una funcionalidad en varias tareas, facilita que las funcionalidades se vayan realizando en paralelo ya que se completan pequeñas tareas, facilitando así la construcción de módulos que van a interconectarse ya que están siendo desarrollados a la vez.

En este proyecto solo ha contribuido una persona, por lo que la confección de los ciclos de trabajo para organizar las tareas ha sido sencilla ya que la misma persona tenía respuesta a todas las preguntas de organización y sabía que posibles impedimentos va a encontrar. Tampoco es necesario que se reúna consigo mismo para ver cómo avanzan las tareas, solo es necesario hacer una introspección de cómo avanza el proyecto en cada ciclo de trabajo para organizar las tareas que se deben realzar y en qué orden.

5.3 Control de versiones

Para realizar el control de versiones se ha utilizado el software Git alojado en la plataforma de *Hosting* GitHub.

Estos dos softwares en conjunto permiten tener una copia remota del código que se está desarrollando y que es accesible para todos los miembros del proyecto en el que se está trabajando.

Además, permite tener un registro de cambios y un control de versiones del código que se está construyendo para la implementación del proyecto. Ya que cada vez que se complete una tarea que requiera cambio de código este será copiado desde el ordenador local del desarrollador al repositorio remoto donde se encuentra la versión del código más actualizada.

Para realizar la tarea del versionado de código y para tener una visual de cómo evoluciona el proyecto se ha utilizado GitKraken una interfaz gráfica para el uso de Git.

Esta interfaz gráfica te permite visualizar el historial de cambios realizados en el código con la fecha en la que se registró el cambio, quien realizó el cambio y un comentario sobre qué es lo que ha cambiado y si afecta a algún modulo en concreto.

A continuación, se muestra una captura de pantalla de Gitkraken usando este proyecto como ejemplo Figura 33. Después de la figura se explicará cada parte de la imagen detalladamente (La Figura se encuentra en el anexo A en un tamaño mayor para facilitar su visionado).

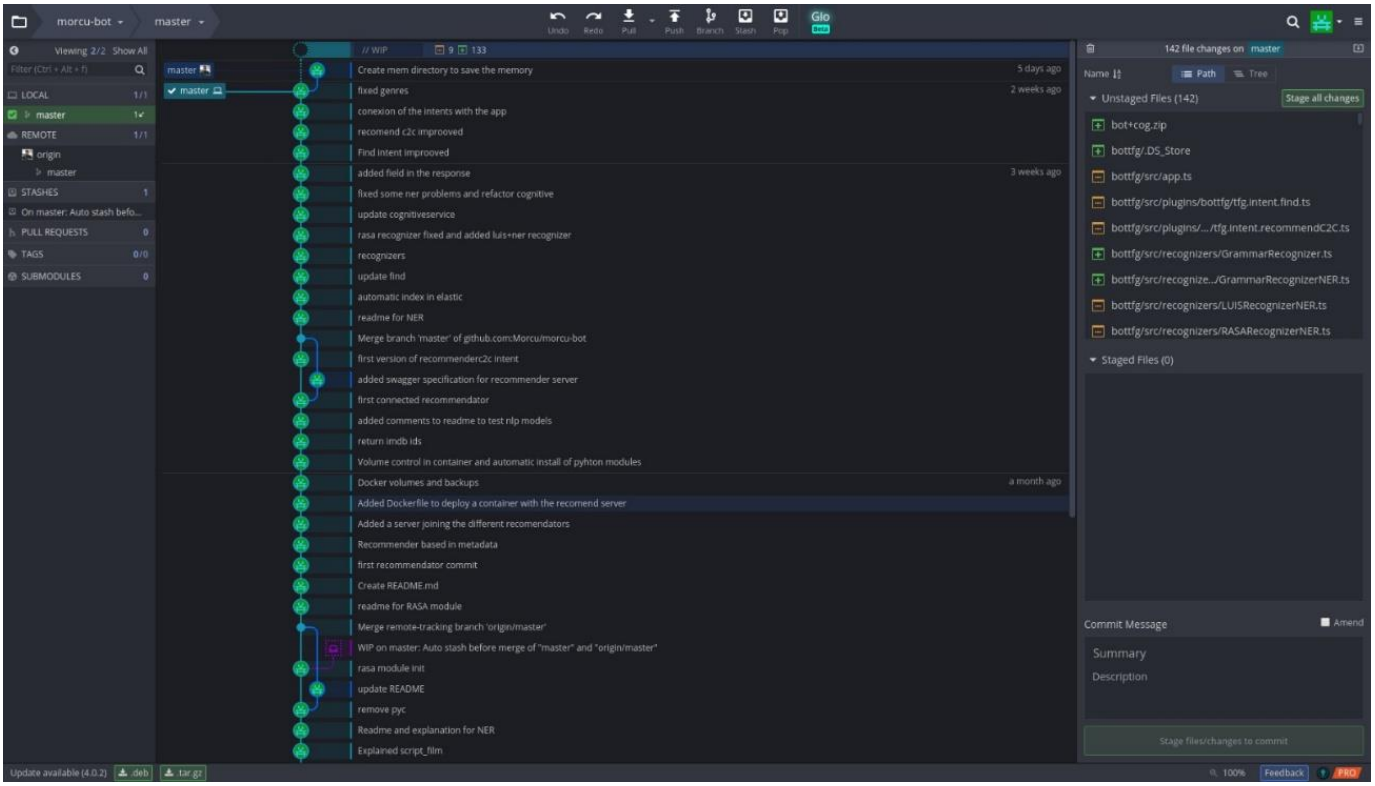


Figura 33: Panel principal de Gitkraken

En la parte central de la imagen anterior se puede apreciar el historial de cambios mencionado anteriormente, en el que se distingue: quien ha realizado el cambio, en qué fecha y que comentario ha hecho.

En la parte superior derecha en el cuadro *Unstaged changes* aparecen los ficheros locales que han sido modificados con respecto a los ficheros que se encuentran en el repositorio remoto.

Justo debajo de este cuadro, se encuentra el cuadro de *staged changes* en este cuadro se muestran los archivos que como los anteriores han sido modificados y que no están en el repositorio remoto pero que quieren ser subidos al repositorio remoto.

Debajo de este cuadro se encuentra el cuadro de *Commit Message,* cuadro en el que se pueden escribir los comentarios sobre las modificaciones que se han realizado al código. Dentro de este cuadro hay otros 2, el primero, *Sumary*, sirve para escribir un comentario corto que se mostrará en el historial del centro, sin embargo, en el segundo, *Description*¸ sirve para escribir comentarios más largos y con mayor complejidad que expliquen por ejemplo el algoritmo implementado o explicaciones para otro desarrollador en caso de que haya que modificar ese fragmento de código, o avisos que especifiquen que el cambio de este fichero afectará a otro modulo también.

En este proyecto se siguen unas buenas prácticas a la hora de escribir los comentarios, extraídas del libro The Git, the Bad and the Ugly [36].

- *Los mensajes de commit tienes dos partes principales: un asunto y un mensaje (como los correos electrónicos). Si el contenido del commit se puede explicar en el asunto, no es necesario incluir un mensaje. Ambas partes deben ir separadas por una línea en blanco.*
- *La línea de asunto no debería extenderse más de 50 caracteres, y el cuerpo del mensaje debería tener una extensión máxima (por línea) de 72 caracteres. Esto ayuda a su visualización en distintas plataformas y dispositivos.*
- *La línea de asunto debe comenzar con letra mayúscula y terminar sin punto. Piensa, por ejemplo, en el asunto de un correo electrónico.*
- *Usa el imperativo en el mensaje de commit. Internamente Git usa el imperativo en los mensajes que genera. Por ejemplo, Merge branch 'feature-refactor' tras fusionar la rama feature-refactor. Para mantener la coherencia de todos los mensajes de commit, adoptaremos la convención de Git de usar el imperativo.*

5.5 Entorno Socioeconómico

En este apartado se expondrá por un lado el presupuesto del proyecto desglosando tanto los costes directos, los de personal y los indirectos. Por otro lado, se hará un análisis exhaustivo del impacto socioeconómico, definiendo un plan de explotación del proyecto y utilizando como apoyo al análisis una matriz DAFO (Debilidades, Amenazas, Fortalezas y Oportunidades).

5.5.1 Presupuesto

En este apartado se van a explicar los costes asociados al proyecto, tanto directos como indirectos.

Los costes directos son aquellos atribuibles al objetivo del coste, en este caso concreto el coste de los equipos, el personal y las herramientas. Los indirectos son aquellos que están compartidos o que no es posible establecer de forma directa como el coste de la electricidad.

A continuación, se desglosan los costes en costes de personal y costes de herramientas.

5.5.1.1 Coste de personal

Para obtener el coste del personal se utilizará la siguiente fórmula:

$$Coste = Duracion\ Proyecto\ * Horas\ dedicadas * Num\ Trabajadores * \frac{Coste\ hombre/\ mes}{Numero\ días\ /\ mes\ \ * 8\ h/dia}$$

Ecuación 4: Coste de personal [37]

Donde:

- *La duración del proyecto se computa en días*
- *Las horas dedicadas son por hombre al día*

Aplicando la fórmula con los valores computados en el proyecto obtenemos:

$$Coste = 238\ * 4 * 1 * \frac{2571{,}42}{30 * 8} = \ 81.599{,}72 = 10.199{,}96$$

Ecuación 5: Coste de personal con valores numéricos

5.5.1.2 Coste de Herramientas

Haciendo referencia al apartado 1.3 donde se definía el Hardware y software utilizado, se desglosarán aquí los precios concretos de dichas herramientas.

5.5.1.2.1 Hardware

Herramienta	Precio	total
Ordenador: Asus R510V	616,19€	616,19€
Smartphone: Samsung Galaxy S7 Edge	616,19€	616,19€

Tabla 24: Coste del Hardware

5.5.1.2.2 Software

Herramienta	Precio	total
Entorno de desarrollo Visual Studio Code	0 €	0 €
Entorno de desarrollo PyCharm	0€ (licencia universidad)	0 €
Node.js	0 €	0 €
Python	0 €	0 €
Git	0€ (licencia universidad)	0 €
Gitkraken	0€ (licencia universidad)	0 €
Elasticsearch	0 €	0 €
Kibana	0 €	0 €
Docker	0 €	0 €
Portainer.io	0 €	0 €
Mongodb	0 €	0 €
Microsoft Bot Framework (Azure)	53€/mes (8 meses)	424 €
ANTLR	0 €	0 €
RASA	0 €	0 €
LUIS	0€ (límite 10.000 peticiones)	0 €
Trello	0€	0€
TOTAL	**-**	**424€**

Tabla 25: Coste del Software

Nota: en los precios mostrados no está incluido I.V.A

Habiéndose definido los precios se procede a calcular el coste total. Para calcular el coste de los dispositivos hay que hacer uso de la siguiente formula de coste imputable, que tiene en cuenta la amortización de los dispositivos para que el coste reflejado sea estrictamente el coste imputable a este proyecto concreto.

$$coste\ imputable\ = \frac{meses\ de\ utilización}{periodo\ deprecacion} * coste\ del\ equipo * porcentaje\ de\ uso$$

Ecuación 6: Coste imputable [38]

En la Tabla 26 se utiliza la fórmula definida para calcular el coste imputable.

Equipo	Coste (€)	Porcentaje de uso dedicado al proyecto	Meses de utilización	Periodo de deprecación (meses)	Coste imputable (€)
Asus R510V	616,19	100%	7,93	60	81,43
Samsung Galaxy S7 Edge	636,35	22.22%	4,06	24	23,91
TOTAL	-	-	-	-	105,34

Tabla 26: Coste Imputable

5.5.1.3 Coste total

Tras haber especificado el coste de cada parte, se procede a sumar los costes de personal, los costes de las herramientas, los costes de los dispositivos y los costes indirectos, cuyo valor será el 20% de los costes anteriormente definidos.

Procedencia del coste	Coste Total
Personal	10.199,96€
Herramientas	424€
Dispositivos	105,34€
Indirectos	2.145,86
Total, sin IVA	12.875,11
Total, con IVA	15.578,88

Tabla 27: Coste total

5.5.2 Análisis socioeconómico

En este apartado primero se realizará un análisis económico explicando el plan de explotación, después se explicará el impacto social del proyecto y por último se compondrá una matriz DAFO analizando Debilidades, Amenazas, Fortalezas y Oportunidades.

Los estudios sobre el impacto económico miden la repercusión y los beneficios de las inversiones, en este caso se realizará el estudio sobre el proyecto en su conjunto, para ello se analizará el impacto directo, indirecto e inducido:

- Impacto directo: se corresponde con el valor añadido que aporta el producto.
  - El sistema transforma los metadatos de las películas, las series, los actores y directores para que el acceso y la recuperación de la información ayude en todo momento al usuario a encontrar lo que busca, o la aproximación más acertada.

    - El valor añadido es por tanto el procesamiento de una información ya existente proveyendo a los usuarios de un sistema inteligente que hace el trabajo por ellos con tan solo pedírselo.
- Impacto indirecto: se corresponde con los sectores que se benefician de que el sistema esté en funcionamiento.
    - Los sectores que se ven beneficiados son los proveedores de servidores en Cloud o en su defecto los proveedores de servidores físicos donde el sistema está siendo ejecutado ya que normalmente estos sistemas funcionan con subscripciones de pago por uso.
- Impacto inducido: se corresponde con los beneficios que pueden conseguir otros sectores por la funcionalidad y el valor añadido que aporta el sistema.
    - Ya que este proyecto se centra en la construcción de un sistema que aporta información relacionada con las películas y las series, el sector del video bajo demanda y de los servicios de retransmisión por streaming pueden verse beneficiados, ya que en sistema en esta fase solo proporciona información, no contenido.
    - Es posible que si el sistema funciona correctamente y tiene una buena acogida entre el público se podría hacer una integración con alguno de estos sectores para que ambas partes se vean beneficiadas ya que un sistema proporcionará información y el otro contenido consiguiendo así un sistema con mucho valor.

Habiendo definido el impacto económico se procede a analizar el impacto social.

El impacto social puede entenderse como la influencia, la huella o el efecto que se genera sobre la sociedad en la que se integra el sistema [39], por lo tanto, el posible impacto de este sistema es que se dejen de mirar listas de películas/series en los principales proveedores de contenidos para escoger algo que ver cuando no se sabe muy bien que ver, o tener que buscar la información del contenido en el que se está interesado entre 2 o 3 páginas saltando de una en otra pulsando botones.

Este sistema da la facilidad de resolver los problemas que se acaban de citar y además con una simple orden de texto, incluso se puede realizar esta orden con la voz si el dispositivo cuenta con un micrófono.

5.5.2.1 DAFO

- Debilidades: Existencia de muchos metadatos procesados, pero falta el contenido en sí mismo.
- Amenazas: Empresas que ya hagan uso de los contenidos y los metadatos, los procesen y hagan sistemas similares, o muestren su información de una manera más amistosa para el usuario.
- Fortalezas: El sistema cuenta con tecnología muy innovadora, sobre todo en el ámbito del procesamiento del lenguaje natural.
- Oportunidades: Existen muy pocos sistemas de este tipo en el mercado actual, ya que es una tecnología que está en pleno desarrollo.

5.6 Marco regulador

Este apartado tiene como objetivo analizar el conjunto de normas y leyes aplicables al sistema. En este caso particular la legislación aplicable es la relacionada a la protección de datos personales, ya que el Bot recopila datos personales de nivel básico del usuario, que permiten identificarlo, durante su funcionamiento, puesto que en la interacción Humano-Bot se recopilan datos sobre la identidad del usuario además de datos como gustos que son de carácter privado.

Por lo tanto, es necesario el cumplimiento de la Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal [40]. Adicionalmente, como el desarrollo del proyecto va más allá de la fecha 25 de mayo de 2018, cuando entra en vigor el Reglamento General de Protección de Datos (RGPD), también se debe actuar conforme a esta nueva normativa [41].

Es importante cumplir con el *habeas data*, recurso legal a disposición de todo individuo que permite acceder a un banco de información o registro de datos que incluye referencias informativas sobre sí mismo [42], que en el ordenamiento jurídico español está conformado por [43]:

- Derecho de acceso de la información por parte del usuario.
- Derecho de rectificación.
- Derecho a la actualización de sus datos.
- Derecho de eliminación o supresión.
- Derecho de oposición.
- Derecho al olvido.

- Derecho de exclusión.

Para asegurar el cumplimiento de estas normativas, se llevan a cabo las siguientes acciones:

- En los avisos legales de la aplicación se informa al usuario de que sus datos son recopilados y guardados por la aplicación para ser tratados con la finalidad de mejorar la experiencia de usuario.
    - Para cumplir con la RGPD se le pide consentimiento explícito de que acepta dichos términos.
- Los datos personales son custodiados por el autor de este TFG mediante el mantenimiento constante y actualizado de la seguridad de los sistemas en los que se almacena dicha información. Se dispone control de acceso y autenticación con un historial de los mismos.
- Los datos almacenados sobre los usuarios serán eliminados cuando dejen de ser pertinentes para la finalidad para la que se recopilaron.
- En todo momento se puede ejercer cualquiera de los derechos antes mencionados.

6 CONCLUSIONES

Este proyecto ha sido un proyecto muy ambicioso en el que se definía desde cero un sistema completo partiendo desde las fases de diseño, fases a veces infravaloradas en pos de la implementación directa, tras haber realizado el proyecto he comprendido la importancia que tiene una visión clara de todos los módulos y sus interrelaciones.

Antes de la implementación per se, era necesario contar con la infraestructura básica del sistema, fue necesario aprender a usar la arquitectura de virtualización Docker y su implementación en entornos Cloud, hasta ahora no tenía constancia de cómo se podían instanciar estas herramientas.

En cuanto al desarrollo del CatBot fue necesario entender desde las primeras fases la interacción del ChatBot con el usuario ya que las ideas de automatización y procesamiento de la información del Bot eran muy potentes pero el añadido de la interfaz hombre-máquina fue un verdadero reto. Sobre todo, en la evaluación de las intenciones de los usuarios.

Los sistemas de procesamiento del lenguaje natural han requerido un estudio y una evaluación previa usando los conocimientos adquiridos en la universidad sobre como evaluar sistemas y como entrenar y probar los modelos, como por ejemplo separar los datos en conjuntos de entrenamiento y test, o realizar una validación cruzad para obtener datos de si los modelos están generalizando.

Finalmente, la revisión de los servicios REST que conectan todos los módulos ha permitido entender las conexiones de red entre distintos entornos y como securizar los mismos.

Este proyecto es una recopilación de los conocimientos adquiridos en el grado y de la ampliación de los mismos ya sea con nueva información sobre los temas que ya conocía o con nuevos temas sobre los que no tenía nada de información, desarrollando un sistema completo, autogestionado y que se comunica con los usuarios usando una aplicación que pueden llevar a todos lados con su móvil.

7 TRABAJOS FUTUROS

En la realización del proyecto al completo se han tomado algunas decisiones de compromiso para poder realizar un proyecto en su totalidad con el tiempo que se contaba. Algunas ideas mostradas en esta memoria tanto en la fase introductoria como en el estado del arte no han sido explotadas en su totalidad, en este apartado se van a comentar aquellas ideas o funcionalidades que permitirían mejorar la aplicación tanto a nivel de funcionalidad como a nivel de usuario:

7.1 Integración con servicios de video bajo demanda en streaming

Uno de los puntos que aportaría gran valor a la aplicación sería que esta, no solo mostrara información, sino que también pudiese mostrar el contenido.

Utilizando gestores de contenido con suscripción a los que los posibles usuarios ya están suscritos, simplemente dando la opción de que, si tienes una cuenta activa vinculada con la aplicación, está permita mediante un comando de voz o un botón que el contenido buscado o recomendado sea mostrado en el dispositivo.

7.2 Mejoras en la interfaz gráfica

Seguir mejorando en la interfaz de comunicación principal con el usuario, para hacerla más atractiva. Una buena interfaz aporta valor al usuario, ya que, por un lado, estará más predispuesto a usarla y por el otro le será más sencilla de usar.

7.3 Seguir entrenando los modelos de comprensión del lenguaje natural

Los modelos de compresión del lenguaje natural deben aprender de los usuarios, es el Bot el que debe aprender como hablan los usuarios y no al revés.

Para realizar esta tarea se pueden analizar los registros de la aplicación para ver que estilo de frases utilizan los usuarios y así poder realizar un flujo de aprendizaje, en el que el Bot este continuamente aprendiendo.

Otro método puede ser la realización de encuestas a los potenciales usuarios para obtener información sobre cómo estos le hablarían al Bot.

7.4 Mecanismos de automatización de actualización de información

Actualmente no se cuenta con ningún mecanicismo automático que permita la carga de datos actualizados a las bases de conocimiento, tanto en la base de datos como en los sistemas cognitivos.

8 GLOSARIO

- Azure: servicio de cloud ofrecido por Microsoft.
- CPU: unidad central de procesamiento.
- Dashboard: tablero con la representación de datos y/o métricas.
- Dataset: conjunto de datos.
- Docker: software de código abierto de virtualización.
- Framework: conjunto buenas prácticas y procesos estandarizados que aportan un marco sobre el que trabajar sobre un determinado tema o proyecto.
- NoSQL: también conocido como “no solo SQL” es un sistema de base de datos que no usan SQL como lenguaje principal de consultas.
- Plugin: complemento, software informático que aporta nueva funcionalidad muy específica a otro software.
- Smartpone: teléfono inteligente.
- Streaming: distribución digital de contenido multimedia.
- Swagger: framwork para diseñar, construir y documentar servicios RESTfull.
- Yaml: formato de serialización de datos fácilmente legible por humanos.
- JSON: formato de texto ligero para el intercambio de datos.
- Scrum: marco de buenas prácticas enfocadas al trabajo colaborativo.

9 BIBLIOGRAFIA

1 "Smartphones: número de usuarios mundiales 2014-2019 | Estadística", Statista, 2018. [Online]. Disponible en: https://es.statista.com/estadisticas/636569/usuarios-de-telefonos-inteligentes-a-nivel-mundial--2019/. [Accedido: 29- Ago- 2018].

2 A. Hindhaugh, "Subscription VOD service revenue in Europe grew by 128% annually between 2011 and 2016", European Audiovisual Observatory, 2018. [Online]. Disponible en: https://www.obs.coe.int/en/web/observatoire/2018-press-releases/-/asset_publisher/qCvKtWM6Klji/content/subscription-vod-service-revenue-in-europe-grew-by-128-annually-since-2011?inheritRedirect=false. [Accedido: 29- Ago- 2018].

3 D. Justo, "El uso de 'smartphones' en España se duplica en los últimos cinco años", Cadena SER, 2018. [Online]. Disponible en: http://cadenaser.com/ser/2017/02/28/ciencia/1488281552_888684.html. [Accedido: 29- Ago- 2018].

4 R. Rodríguez, "Enganchados al móvil: España, 5º país del mundo que más tiempo pasa con el teléfono. Noticias de Tecnología", El Confidencial, 2018. [Online]. Disponible en: https://www.elconfidencial.com/tecnologia/2017-05-26/movil-uso-exceso-espana-salud-enganchados-smartphone_1389117/. [Accedido: 29- Ago- 2018].

5 "Guidelines", Material Design, 2018. [Online]. Disponible en: https://material.io/design/guidelines-overview/#addition. [Accedido: 29- Ago- 2018].

6 B. Martinez, "La lógica de la usabilidad: las 10 heurísticas de Jakob Nielsen", BEEVA | Soluciones de tecnología e innovación para empresas, 2018. [Online]. Disponible en: https://www.beeva.com/beeva-view/diseno-y-ux/la-logica-de-la-usabilidad/. [Accedido: 29- Ago- 2018].

7 "Docker", Docker, 2018. [Online]. Disponible en: https://www.docker.com/. [Accedido: 29- Ago- 2018].

8 R. Molla, "Two-thirds of adults worldwide will own smartphones next year", Recode, 2018. [Online]. Disponible en: https://www.recode.net/2017/10/16/16482168/two-thirds-of-adults-worldwide-will-own-smartphones-next-year. [Accedido: 29- Ago- 2018].

9 L. Goldberg, "1 in 5 Smartphone Owners Worldwide Use Their Device Every 5 Minutes & Nearly Half of All Users Are Motivated to React to Ads after Seeing Them

on Mobile", IAB, 2018. [Online]. Disponible en: https://www.iab.com/news/globalmobilerelease/. [Accedido: 29- Ago - 2018].

10 Informe ditrendia: Mobile en España y el Mundo 2017, 1st ed. Madrid: Ditrendia, 2018.

11 D. Sunnebo, "Kantar - Ventas de smartphones: las marcas chinas toman Europa", Es.kantar.com, 2018. [Online]. Disponible en: https://es.kantar.com/tech/m%C3%B3vil/2018/mayo-2018-cuota-de-mercado-de-smartphones-en-espa%C3%B1a-primer-trimestre-2018/. [Accedido: 29- Ago- 2018].

12 A FUTURE THAT WORKS: AUTOMATION, EMPLOYMENT, AND PRODUCTIVITY, 1st ed. Nueva York: McKinsey & Company, 2017.

13 B. Abu Shawar and E. Atwell, Chatbots: Are they Really Useful?, 1st ed. Alemania: ? LDV-Forum: Zeitschrift für Computerlinguistik und Sprachtechnologie, 2007, pp. 29-49.

14 D. Cerdas Mendez, "Evolución de los Chatbots – Planeta Chatbot : todo sobre los Chatbots y la Inteligencia Artificial", Planeta Chatbot, 2017. [Online]. Disponible en: https://planetachatbot.com/evoluci%C3%B3n-de-los-chatbots-48ff7d670201. [Accedido: 29- Ago - 2018].

15 D. Collaguazo, "¿Qué es el Procesamiento de Lenguaje Natural y cómo ponerlo en práctica con recursos abiertos?", iadb, 2018. [Online]. Disponible en: https://blogs.iadb.org/abierto-al-publico/2017/06/20/que-es-el-procesamiento-de-lenguaje-natural-y-como-ponerlo-en-practica-con-recursos-abiertos/. [Accedido: 04- Sep- 2018].

16 "LUIS: Language Understanding Intelligent Service", Luis.ai, 2018. [Online]. Disponible en: https://www.luis.ai/. [Accedido: 29- Ago - 2018].

17 P. Isasi Viñuela and I. Galván León, Redes de neuronas artificiales. Madrid, [etc.]: Pearson-Prentice Hall, 2008.

18 LeCun Y., Haffner P., Bottou L., Bengio Y. (1999) Object Recognition with Gradient-Based Learning. In: Shape, Contour and Grouping in Computer Vision. Lecture Notes in Computer Science, vol 1681. Springer, Berlin, Heidelberg

19 D. Ciresan, U. Meier, L. M. Gambardella, J. Schmidhuber - Deep, Big, Simple Neural Nets for Handwritten Digit Recognition, December 2010

20 "Rasa NLU: Language Understanding for chatbots and AI assistants", Rasa.com, 2018. [Online]. Disponible en: https://rasa.com/docs/nlu/. [Accedido: 29- Ago - 2018].

21 C. McCormick, "Word2Vec Tutorial - The Skip-Gram Model", Mccormickml.com, 2016. [Online]. Disponible en: http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/. [Accedido: 29- Ago- 2018].

22 "word2vec", Code.google.com, 2018. [Online]. Disponible en: https://code.google.com/archive/p/word2vec/. [Accedido: 29- Ago - 2018].

23 "Gramática", Rae.es, 2018. [Online]. Disponible en: http://www.rae.es/obras-academicas/gramatica. [Accedido: 13- Sep- 2018].

24 Hausser, R. (2001). Foundations of computational linguistics. 2nd ed. Berlin: Springer.

25 "Analizador sintáctico LL - EcuRed", Ecured.cu, 2018. [Online]. Disponible en: https://www.ecured.cu/Analizador_sint%C3%A1ctico_LL. [Accedido: 13- Sep- 2018].

26 "MovieLens 1M Dataset", GroupLens, 2018. [Online]. Disponible en: https://grouplens.org/datasets/movielens/1m/. [Accedido: 13- Sep- 2018].

27 E. Liu, "TF-IDF, Term Frequency-Inverse Document Frequency", Ethen8181.github.io, 2015. [Online]. Disponible en: http://ethen8181.github.io/machine-learning/clustering_old/tf_idf/tf_idf.html. [Accedido: 13- Sep- 2018].

28 "Requerimientos Funcionales y No Funcionales, ejemplos y tips", Medium, 2018. [Online]. Disponible en: https://medium.com/@requeridosblog/requerimientos-funcionales-y-no-funcionales-ejemplos-y-tips-aa31cb59b22a. [Accedido: 13- Sep- 2018].

29 C. Wong, "Using an EventBus in Android Pt 1: Why an EventBus?", medium, 2015. [Online]. Disponible en: https://medium.com/@cainwong/using-an-eventbus-in-android-pt-1-why-an-eventbus-c2c9cdff41d7https://medium.com/@cainwong/using-an-eventbus-in-android-pt-1-why-an-eventbus-c2c9cdff41d7. [Accedido: 13- Sep- 2018].

30 "greenrobot/EventBus", GitHub, 2018. [Online]. Disponible en: https://github.com/greenrobot/EventBus. [Accedido: 13- Sep- 2018].

31 "Key concepts in the Bot Framework Direct Line API 3.0 - Bot Service", Docs.microsoft.com, 2017. [Online]. Disponible en: https://docs.microsoft.com/en-us/azure/bot-service/rest-api/bot-framework-rest-direct-line-3-0-concepts?view=azure-bot-service-3.0. [Accedido: 13- Sep- 2018].

32 Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005), pp. 363-370.

33 J. Fernández, "MongoDB: Introducción - BI Geek Blog", BI Geek Blog, 2018. [Online]. Disponible en: https://blog.bi-geek.com/mongodb-introduccion/. [Accedido: 13- Sep- 2018].

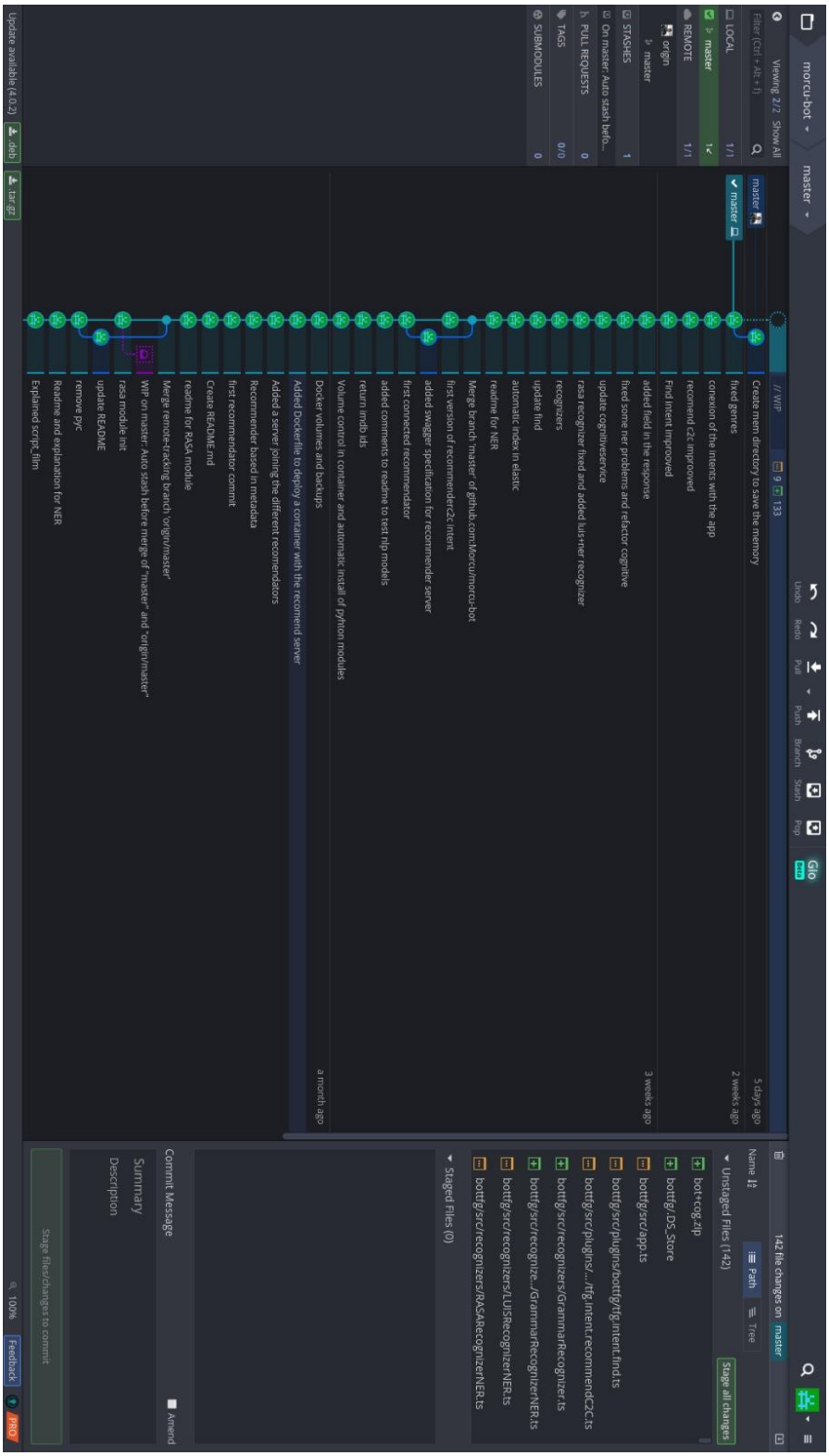
34 D. Braun, A. Hernandez Mendez, F. Matthes and M. Langen, Evaluating Natural Language Understanding Services for Conversational Question Answering Systems. Munich, 2017, pp. 6 - 7.

35 "Qué es SCRUM", Proyectos Ágiles, 2018. [Online]. Disponible en: https://proyectosagiles.org/que-es-scrum/. [Accedido: 13- Sep- 2018].

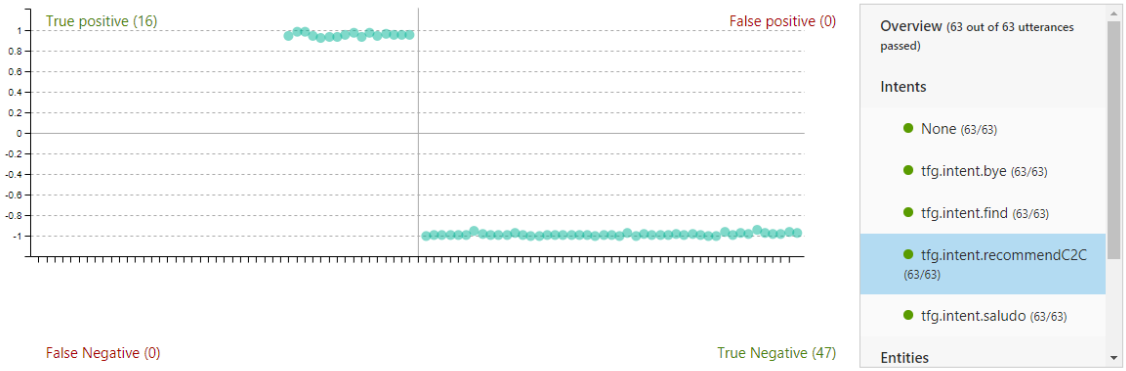
36 D. Estevez, The Git, the Bad and the Ugly, 1st ed. 2016.
37 Clucom.com. (2018). Cálculo de costes en recursos humanos de cada proyecto. [online] Disponible en: https://www.clucom.com/web/calculo-de-costes-rrhh-proyecto-partes-horarios/ [Accedido 22 Sep. 2018].
38 Ciberconta.unizar.es. (2018). EL COSTE DEL EQUIPO PRODUCTIVO: AMORTIZACIóN Y GASTOS GENERALES. [online] Disponible en: https://ciberconta.unizar.es/leccion/costprod/100.HTM [Accedido 22 Sep. 2018].
39 "Medición del Impacto social - Forética", Forética, 2018. [Online]. Disponible en: http://www.foretica.org/tematicas/medicion-impacto-social/. [Accedido: 13- Sep- 2018].
40 Ministerio de la presidencia, relaciones con las cortes e igualdad, "Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal.", BOE, 1999.
41 Unión Europea, "Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance)", Unión Europea, 2016.
42 J. Pérez Porto and A. Gardey, "Definición de habeas data — Definicion.de", Definición.de, 2009. [Online]. Disponible en: https://definicion.de/habeas-data/. [Accedido: 13- Sep- 2018].
43 "El "habeas data" en el ordenamiento jurídico de España y Venezuela | | Derecho & Perspectiva", Derechoyperspectiva.es, 2015. [Online]. Disponible en: http://derechoyperspectiva.es/el-habeas-data-en-espana-y-venezuela/. [Accedido: 13- Sep- 2018].

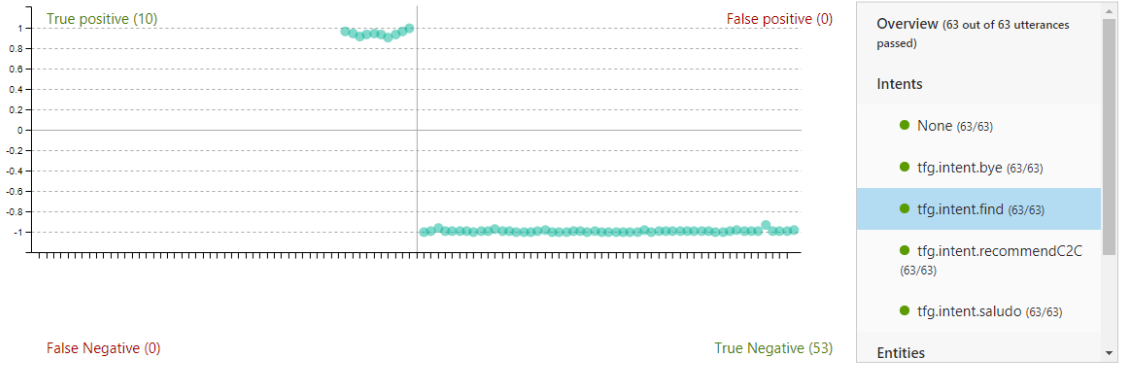
10 ANEXO

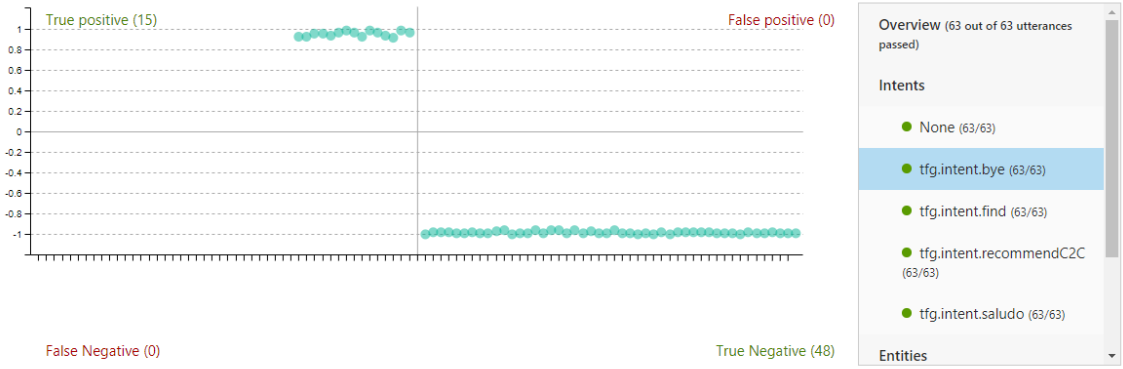
Anexo A Gitkraken dashboard

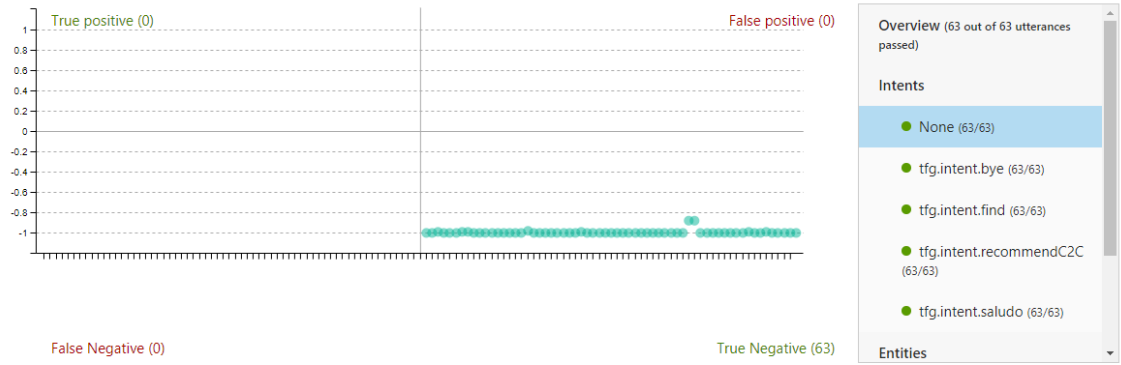


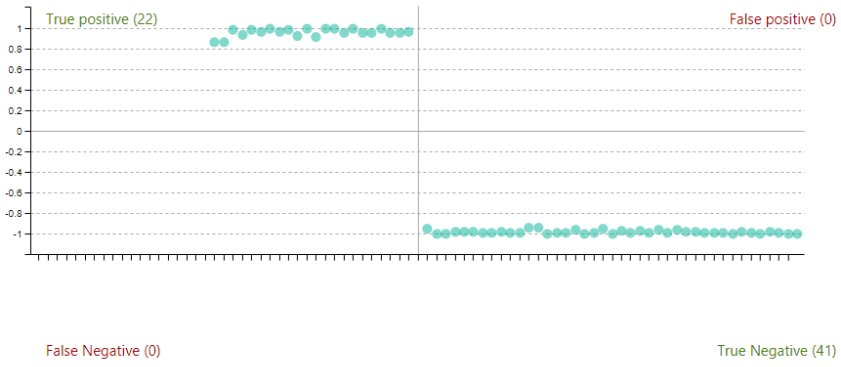
Anexo B Imágenes de test de LUIS sin NER











False Negative (0)

True Negative (41)

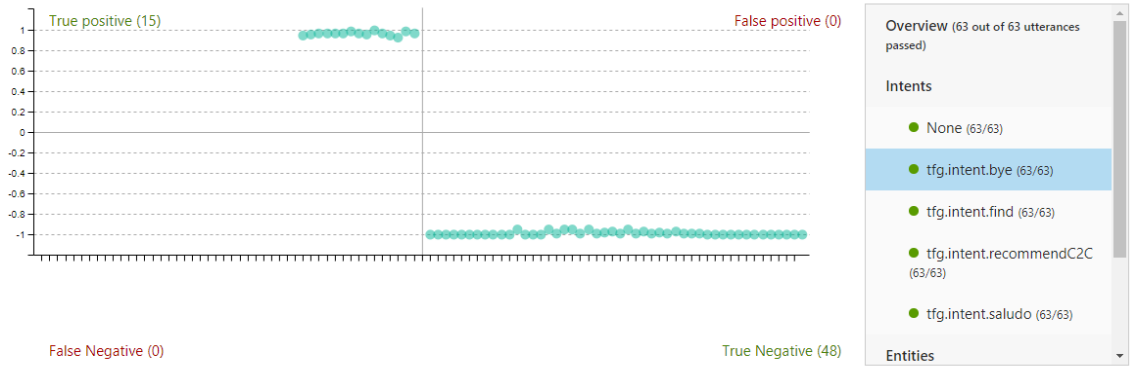
Overview (63 out of 63 utterances passed)

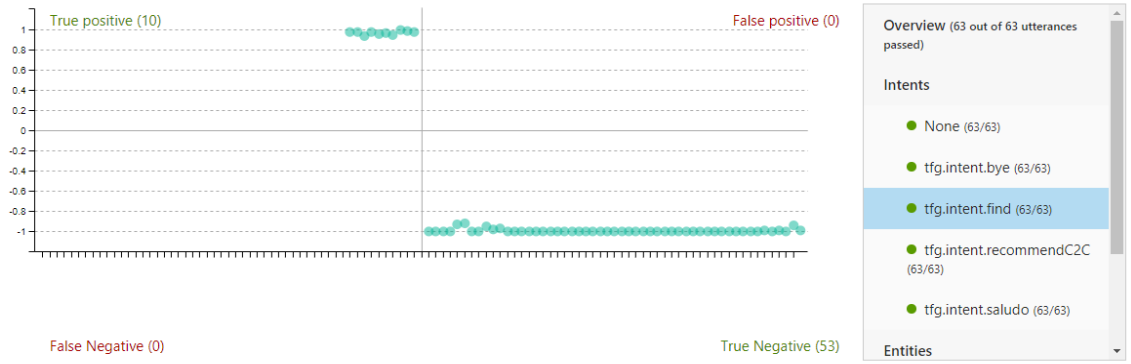
Intents

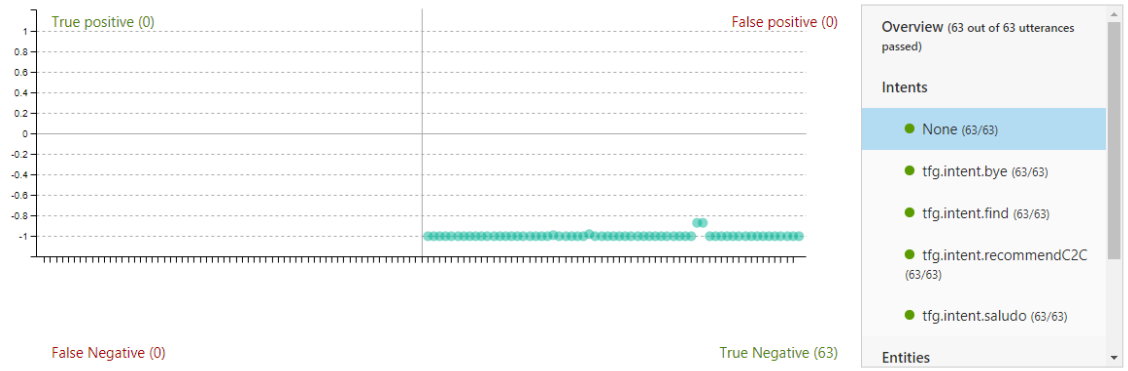
- None (63/63)
- tfg.intent.bye (63/63)
- tfg.intent.find (63/63)
- tfg.intent.recommendC2C (63/63)
- tfg.intent.saludo (63/63)

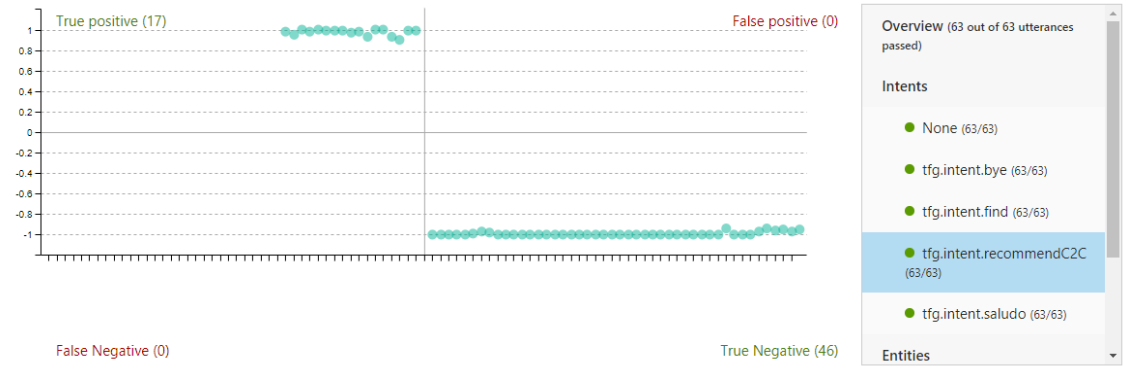
Entities

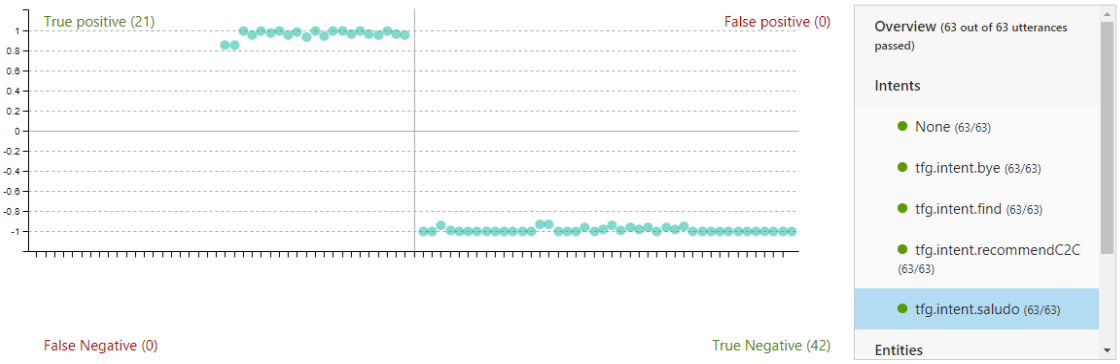
Anexo C Imágenes de test de LUIS con NER








True positive (17)
False positive (0)
False Negative (0)
True Negative (46)
Overview (63 out of 63 utterances passed)
Intents
None (63/63)
tfg.intent.bye (63/63)
tfg.intent.find (63/63)
tfg.intent.recommendC2C (63/63)
tfg.intent.saludo (63/63)
Entities


True positive (21)
False positive (0)
False Negative (0)
True Negative (42)
Overview (63 out of 63 utterances passed)
Intents
None (63/63)
tfg.intent.bye (63/63)
tfg.intent.find (63/63)
tfg.intent.recommendC2C (63/63)
tfg.intent.saludo (63/63)
Entities

11.ANEXO EN INGLÉS

Este anexo recoge las competencias en ingles de este trabajo de finde grado (TFG) en el que se incluye la introducción, los objetivos, los resultados y las conclusiones en inglés.

uc3m | Universidad Carlos III de Madrid

Computer Engineering Degree

2017-2018

Final Degree Project

Design and implementation of an entertainment Bot

David Morcuende Cantador

Tutor

Araceli Sanchis de Miguel

Leganés, 2018

Abstract

This Final Degree Project (TFG) is about the implementation of a ChatBot that allows users to ask using both text and voice (in Spanish language), for recommendations by gender and similarity, information about: movies and series.

To develop this system has been necessary, to develop some microservices: a natural language recognizer system, an Android application, a Bot, and a recommender system. The main function of the natural language recognizer system is process the user's text input and extract information from the user's intentions and keywords in the text, in this system statistic models, artificial neural networks and grammars compete among themselves.

The Android application is the information input and output interface, it is the user's first access point with the system, and it is the interface that shows the user's information.

The recommender system is responsible for giving recommendations to users, using pretrained models, these models are based on the application of cosine similarity and the polynomial Kernel function.

It should be noted that the infrastructure of the entire system has been designed to be executed in the cloud with automated deploy scripts using Docker container technology to make the systems efficient, effective and easy to maintain and monitor.

To develop all the technology, an analysis has been carried out on the current technologies and those that most benefit the system as a whole so that this system transcends beyond this project, and to be easily maintainable and updatable as a market product, benefiting and helping any user who wants to use the application.

Keywords

ChatBot; Natural Language Processing; Recommender systems; statistical models; artificial neural networks; grammars.

TABLE OF CONTENTS

1. INTRODUCTION

This section describes the context and motivation of this Final Degree Project, TFG, which explains the studied topics and its implementations. In addition to explain the target objectives, which will be widely explained in the following sections, its research, design and subsequent implementation. This section also clarifies all the technology used in the realization of the TFG, both Hardware and Software. Finally, the structure of this report is described, indicating the sections that compose it and a brief summary of them.

1.1 Context and motivation

Nowadays and with the rise of automation, virtual assistants, home automation ... people are looking for easier and comfortable ways to solve their problems. So, many people already rely on virtual assistants such as Siri or Google Assistant to solve small mundane tasks quickly, like, setting an alarm at a certain time, or ask about todays weather... these assistants are embedded in the smartphones that today the vast majority of people carry in their pockets, as you can see in Figure 1 the total number of smartphone users in the world grows every year in several hundred thousand and it is estimated that by 2019, users will reach more than 2 and a half million.

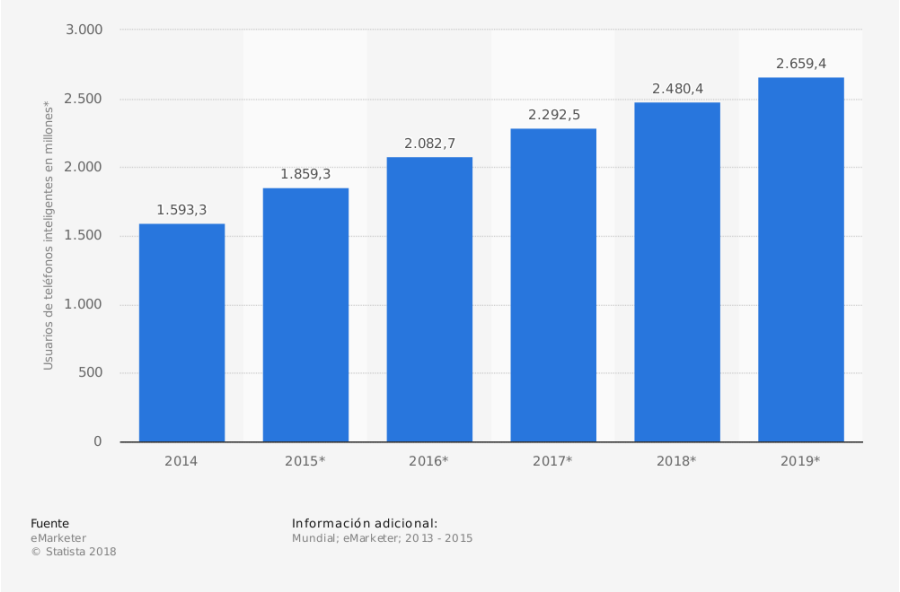


Figure 31 Smartphone users [1]

On the other hand, more and more video on demand is consumed, either because people have less time or because the series they want to watch is not broadcasted at a

convenient time. So, platforms that, under the payment of a subscription, either annual or monthly, has been created, allowing the users to view content on demand through Streaming video. As seen in Figure 2, the percentage of subscriptions in Europe is growing exponentially.

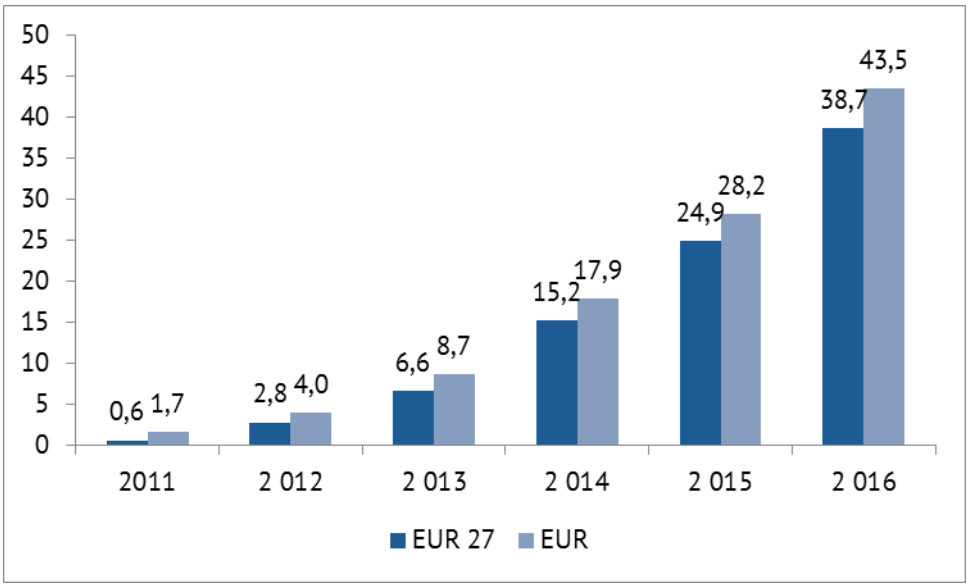


Figure 32: Europe streaming platforms subscription percentage [2]

Combining these two data, it was concluded that an assistant embedded in a smartphone is increasingly necessary. An assistant that allows to ask quickly for content that you want to watch, or if there are not any specific content that you want to watch, ask for a recommendation based on, for example, another movie or the user's preferences.

Build a system like this allows the developer to situate himself in a field that is still being developed and thus be able to contemplate from inside as they are evolving this type of systems.

Having analyzed data and conclusions, provides an idea of the problem and how to offer a solution, but to be able to give a good solution you have to establish some objectives.

1.2 Objectives

The objective of this project is to develop a ChatBot that allows a textual interface both written and spoken in Spanish language that gives users the possibility to ask questions about movies and series and ask for recommendations based on other films and

genders. In addition to prepare the infrastructure for automatic deployment in servers or Cloud environments.

To do this, it is necessary to design an Android application that acts as a bidirectional intermediary between the user and the Bot, which will return one or more responses to a text entry.

Apart from the design of the Android application, it will be necessary to design the Bot that will be responsible for processing the text and give a consistent response to the purpose sought by the user. This requires the generation of a cognitive module that is responsible for the textual processing and a recommender module in case the user asks for recommendations.

With all this information, the project should focus on:

- Design and build a ChatBot that is able to give accurate and concrete answers to a user request, thus solving the problem of having to navigate between hundreds of movies to read your synopsis and maybe find something that you like having lost a lot of time.
- Develop a mobile application that allows the user to interact with the Bot. The mobile is chosen as the interface, since in Spain the use of smartphones has doubled in the last 5 years [3] and Spain is also the fifth country in the world in which more time is spent with the telephone, with an average of 2 hours and 11 daily minutes per user [4].
- This application must have a simple and minimalist design, following the good practices dictated by material design [5] and the rules of Nielsen's Heuristics [6]. This design is designed so that the interface is easy to use, although when it is a ChatBot with a conversational channel, its design must also be taken into account so that the Bot generates appropriate phrases that guide the user to formulate questions that generate correct answers and the user is not frustrated if the Bot at some point in the conversation does not understand perfectly his request.
- It will be necessary to build a recommender system that is able to give correct recommendations to the gender request, or to find similarities between films.
- One of the main components will be Natural Language Processing, since it is what will dictate the user's intention when making the request, if the system manages to

interpret it correctly, it will be able to give a correct answer, if not, the system will not know what to answer or will respond to anything other than expected.

- All the modules will be based on Docker container technology so that the control of each system is exhaustive and can be maintained easily. [7]
- Once the objectives have been established, the necessary hardware and software are described.

1.3 Hardware y software used

For the development of the project it has been necessary to have the following resources, both hardware and software.

1.3.1 Hardware

- Computer: Asus R510V
- Smartphone: Samsung Galaxy S7 Edge

1.3.2 Software

- Visual Studio Code development environment.
- PyCharm development environment.
- Node.js: Execution environment for JavaScript, asynchronous with input and output architecture based on events.
- Python: Interpreted programming language
- Git: Version control software
- Gitkraken: Graphical interface for Git.
- Elasticsearch: open source RESTful search and analysis engine
- Kibana: Graphic assistant for real-time analysis of Elasticsearch data.
- Docker: Open source software that favors and automates the deployment of applications, providing a layer of virtualization.
- Portainer.io: graphical interface for the maintenance of Docker containers.
- Mongodb: NoSQL database oriented to documents.
- Microsoft Bot Framework: Bots development framework provided by Microsoft
- ANTLR: software for generating and processing grammars.
- Trello: project management software with a Kanban type board.

1.4 Document structure

This section explains the structure of the memory, which in this case is divided into 5 sections, each section is explained separately below:

1 **Introduction**: section describing the context in which the project is framed, the motivation that led to its selection and execution, the objectives set to achieve the project, the software and hardware necessary to carry out the project and finally a description of the structure of the memory.
2 **Conclusions**:
3 **Future work**: chapter in which the tasks to be done are described if you want to continue with the project and improve it.
4 **Glossary**: chapter in which technical terms used in this document are collected together with their definition to facilitate the reader's understanding.
5 **Bibliography**: section where the list of bibliographic references of the sources of information used in this document can be found. The ordering of the same follows a system of numerical citation, the references will be ordered in order of appearance in the text.

2. CONCLUSIONS

This project has been a very ambitious project in which a complete system was defined from scratch starting from the design phases, sometimes undervalued phases in pursuit of direct implementation, after having made the project I understood the importance of a clear vision of all the modules and their interrelations.

Before the implementation per se, it was necessary to have the basic infrastructure of the system, it was necessary to learn to use the Docker virtualization architecture and its implementation in Cloud environments, until now I had no record of how these tools could be instantiated.

Regarding the development of the CatBot it was necessary to understand from the first phases the interaction of the ChatBot with the user since the ideas of automation and information processing of the Bot were very powerful, but the addition of the human-machine interface was a real challenge. Above all, in the evaluation of the intentions of the users.

The natural language processing systems have required a study and a previous evaluation using the knowledge acquired in the university on how to evaluate systems and how to train and test the models, such as separating the data into training and test sets or making a cross validation to obtain data on whether the models are generalizing.

Finally, the review of the REST services that connect all the modules has allowed us to understand the network connections between different environments and how to secure them.

This project is a compilation of the knowledge acquired in the degree and its extension either with new information about the topics that I already knew or with new topics about which I did not have any information, developing a complete system, self-managed and who communicates with users using an application that they can carry everywhere with their mobile.

3. FUTURE WORK

In the realization of the whole project some compromise decisions have been taken to be able to carry out a project in its totality with the time that was counted. Some ideas have not been exploited completely. In this section we will discuss those ideas or functionalities that would allow improving the application both at the functionality level and at the level of user:

3.1 Integration with streaming video on demand services

One of the points that would bring great value to the application could be an integration with streaming platforms, so the application will not only show the information but could also play the content.

If the application had the ability to link the user's platform subscription to the content, it could play that content with only a voice command or a button when the search or the recommendation have been placed.

3.2 Improvements in the graphic interface

Continue improving in the main communication interface with the user, to make it more attractive. A good interface provides value to the user, since, on the one hand, he will be more predisposed to use it and on the other it will be easier to use.

3.3 Keep training natural language processing's models

Natural language processing's models must learn from users, it is the Bot that must learn how users speak and not the other way around.

To perform this task, the application reports must be analized to discover what style of sentences are the most repeated by the users to be able to perform a learning flow, in which the Bot is continuously learning.

Another method may be to conduct surveys to potential users to obtain more information about how users would talk to the Bot.

3.4 Automation mechanisms for updating information

Currently there is no automatic mechanism that allows the upload of updated data to the knowledge bases, both in the database and in the cognitive systems.

4. GLOSARRY

- Docker: open source virtualization software.
- Framework: set of good practices and standardized processes that provide a standar on which to work on a specific topic or project.
- NoSQL: also known as "not just SQL" is a database system that does not use SQL as the primary query language.
- Streaming: digital distribution of multimedia content.

5. BIBLIOGRAPHY

1 "Smartphones: número de usuarios mundiales 2014-2019 | Estadística", Statista, 2018. [Online]. Available at: https://es.statista.com/estadisticas/636569/usuarios-de-telefonos-inteligentes-a-nivel-mundial--2019/. [Accessed: 29- Aug- 2018].

2 A. Hindhaugh, "Subscription VOD service revenue in Europe grew by 128% annually between 2011 and 2016", European Audiovisual Observatory, 2018. [Online]. Available at: https://www.obs.coe.int/en/web/observatoire/2018-press-releases/-/asset_publisher/qCvKtWM6Klji/content/subscription-vod-service-revenue-in-europe-grew-by-128-annually-since-2011?inheritRedirect=false. [Accessed: 29- Aug- 2018].

3 D. Justo, "El uso de 'smartphones' en España se duplica en los últimos cinco años", Cadena SER, 2018. [Online]. Available at: http://cadenaser.com/ser/2017/02/28/ciencia/1488281552_888684.html. [Accessed: 29- Aug- 2018].

4 R. Rodríguez, "Enganchados al móvil: España, 5º país del mundo que más tiempo pasa con el teléfono. Noticias de Tecnología", El Confidencial, 2018. [Online]. Available at: https://www.elconfidencial.com/tecnologia/2017-05-26/movil-uso-exceso-espana-salud-enganchados-smartphone_1389117/. [Accessed: 29- Aug- 2018].

5 "Guidelines", Material Design, 2018. [Online]. Available at: https://material.io/design/guidelines-overview/#addition. [Accessed: 29- Aug- 2018].

6 B. Martinez, "La lógica de la usabilidad: las 10 heurísticas de Jakob Nielsen", BEEVA | Soluciones de tecnología e innovación para empresas, 2018. [Online]. Available at: https://www.beeva.com/beeva-view/diseno-y-ux/la-logica-de-la-usabilidad/. [Accessed: 29- Aug- 2018].

7 "Docker", Docker, 2018. [Online]. Available at: https://www.docker.com/. [Accessed: 29- Aug- 2018].

8