

Grado Universitario en Ingeniería Informática  
2017-2018

*Trabajo Fin de Grado*

“Título del trabajo”

David Morcuende Cantador

Tutor

Araceli Sanchis de Miguel

Leganés, 2018



Esta obra se encuentra sujeta a la licencia Creative Commons **Reconocimiento – No Comercial – Sin Obra Derivada**

**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 entre películas e información sobre: películas, series, actores y directores.

Para desarrollar este sistema ha sido necesario desarrollar diversos módulos: un sistema de reconocedores de lenguaje natural castellano, una aplicación Android, un Bot y un sistema recomendador. 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 hecho análisis sobre las tecnologías actuales y las que más beneficiarían al sistema en su conjunto para que este proyecto transcienda mas 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

**ÍNDICE DE CONTENIDOS**

[1. INTRODUCCIÓN 1](#_Toc522349648)

[2.1 CONTEXTO 1](#_Toc522349649)

[2.2 OBJETIVOS 1](#_Toc522349650)

[2.3 HARDWARE Y SOFTWARE UTILIZADO 1](#_Toc522349651)

[2.4 ESTRUCTURA DE LA MEMORIA 1](#_Toc522349652)

[2. ESTADO DEL ARTE 1](#_Toc522349653)

[3. BOT 1](#_Toc522349654)

[4. EVALUACIÓN 1](#_Toc522349655)

[5. GESTIÓN DEL PROYECTO 2](#_Toc522349656)

[6.1 PRESUPUESTO 2](#_Toc522349657)

[6.2 PLANIFICAIÓN 2](#_Toc522349658)

[6.3 ENTORNO SOCIO-ECONÓMICO 2](#_Toc522349659)

[6.4 MARCO REGULADOR 2](#_Toc522349660)

[6. CONCLUSIONES 3](#_Toc522349661)

[7. TRABAJO FUTURO 3](#_Toc522349662)

[8. BIBLIOGRAFIA 3](#_Toc522349663)

[9. ANEXO 3](#_Toc522349664)

[Figura 1.1 Usuarios de teléfonos Inteligentes [] 2](#_Toc522452418)

[Figura 2: Porcentaje de subscripciones plataformas de streaming en Europa [] 2](#_Toc522452419)

[Figura 3: Cuota de mercado de sistemas operativos en España (%) [] 7](#_Toc522452420)

[Figura 4: Interacción humano-Bot 8](#_Toc522452421)

[Figura 5: Ejemplo Word2vec [] 10](#_Toc522452422)

[Figura 6: Ejemplo de uso de modelo de word2vec [] 11](#_Toc522452423)

[Figura 7: Ejemplo calculo word2vec [] 11](#_Toc522452424)

[Figura 8: Ejemplo calculo final word2vec 12](#_Toc522452425)

Índice de tablas

# INTRODUCCIÓN

En este apartado se describe el contexto y la motivación de este Trabajo de fin de grado que explica la temática estudiada e implementada y el por qué. Además de instanciar los objetivos marcados y que serán ampliamente explicado 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.

Responde a la pregunta ¿qué se estudia, y por qué? La introducción debe hacer breve referencia a los siguientes aspectos del trabajo:

Motivación o justificación (*Por qué hemos elegido este tema para el trabajo, por qué es relevante esta investigación*)

Objetivo concreto de este estudio (*Antecedentes, estado de la cuestión, supuesto o hipótesis de trabajo*)

## 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 más fáciles y cómodas para solucionar sus problemas, mucha gente ya se apoya en asistentes virtuales véase Sriri 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 siguiente gráfica 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.

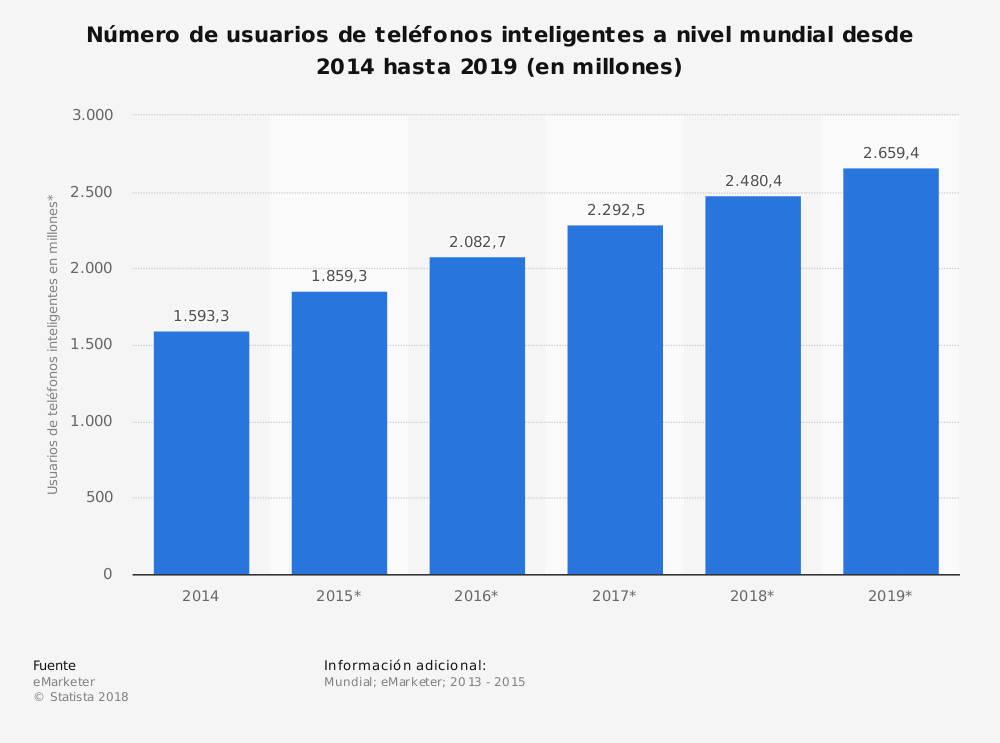


Figura 1.1 Usuarios de teléfonos Inteligentes []

Por otro lado, la gente cada vez consume más video bajo demanda ya sea porque tienen un tiempo limitado o porque la serie que quiere ver no la emiten en un horario conveniente, esto dio 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 siguiente gráfica el porcentaje de subscripciones en Europa está creciendo exponencialmente.

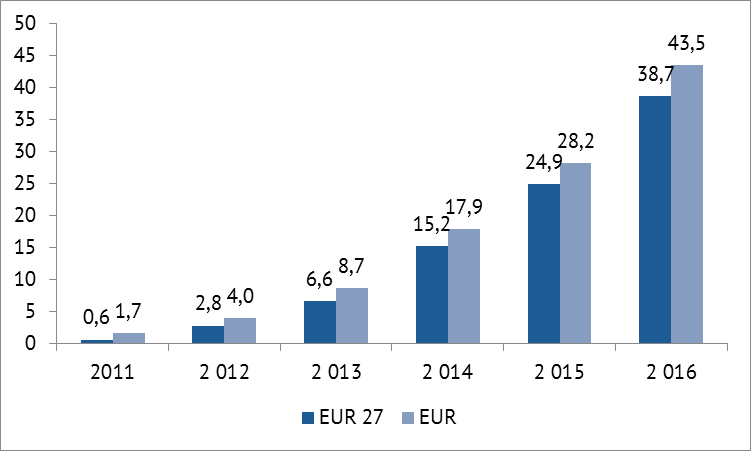


Figura 2: Porcentaje de subscripciones plataformas de streaming en Europa []

Aunando estos dos datos, se llegó a la conclusión de que cada vez es más necesario un asistente, que este embebido en un smartphone, que te permita de manera rápida preguntar por un contenido que se quiera visionar, o si no tiene claro que es lo que se quiere ver, preguntar por una recomendación en base por ejemplo a otra película o a tus gustos.

La creación de un sistema como este me permite situarme 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 desdibujada 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, series, actores y/o directores, y pedir recomendaciones basadas en otras películas y 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, para poder llevar esto a cabo será necesario generar un **módulo cognitivo** que se encargue del procesado textual y 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 [] y además España es el quinto país del mundo que más tiempo pasa con el teléfono con una media de 2 horas y 11 minutos diarios por usuario [].
* Dicha aplicación deberá tener un diseño sencillo y minimalista, siguiendo las buenas prácticas dictadas por *material design* [] y las reglas de la Heurística de Nielsen []. 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 la petición del usuario.
* Será necesario construir un sistema recomendador que sea capaz de dar recomendaciones correctas frente a la petición de género, o 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 una cosa distinta a la esperada.
* 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. []

## 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.

## 1.4 Estructura de la memoria

# ESTADO DEL ARTE

En este apartado se desarrollará de una manera más minuciosa 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 solo llamar o mandar mensajes 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 y si buscamos los datos de España, un 88,9% de los adultos españoles poseen uno de estos teléfonos [].

Este auge masivo viene dado porque los smartphones cada vez son mas 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 la cosa no queda ahí, 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.

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… de media un usuario tiene 17,8 aplicaciones instaladas [].

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.

### 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 una 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 gráfica 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%

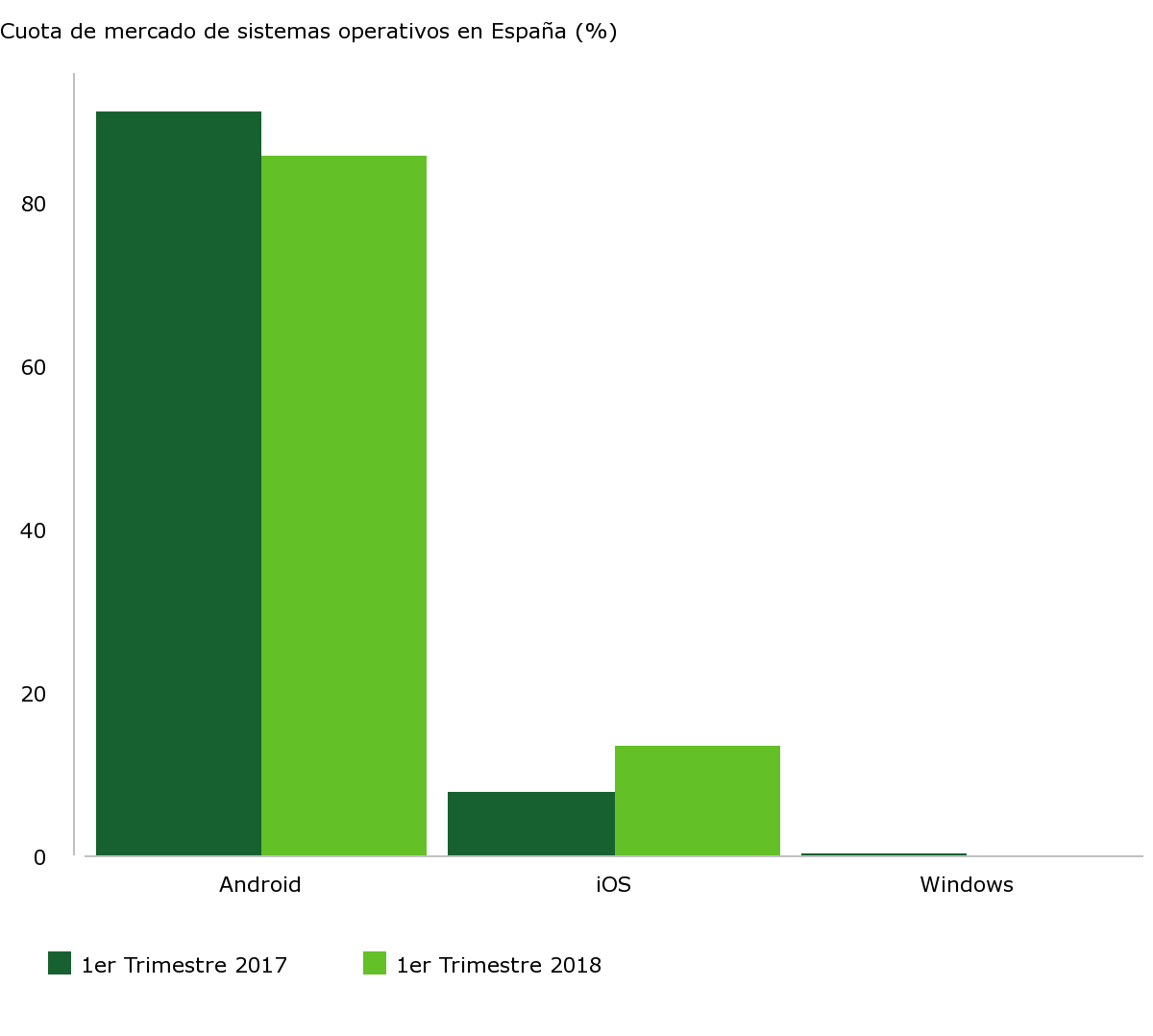


Figura 3: Cuota de mercado de sistemas operativos en España (%) []

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 fácil de definir en pasos concretos.

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

Pero el problema llega cuando quieres 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 [], 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 de la siguiente manera:

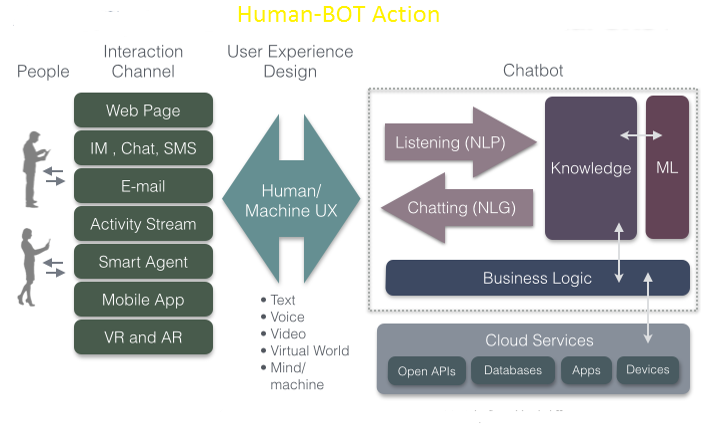


Figura 4: Interacción humano-Bot

### 2.3.1 Procesamiento del lenguaje natural

La primera parte, y principal, del ChatBot es la entrada, que será 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 aun no hay ningún método que destaque claramente sobre los demás. 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) [] provisto por Microsoft.

Es un método de aprendizaje no supervisado en el que para una frase dada hay que definirle 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 por 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…
  + *Thor*: entidad detítulo, ya que es el título de una película.

Ya que LUIS es un software comercial no hay 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)

Para la implementación de estos algoritmos se ha usado el software libre Rasa NLU []. Como el anterior también es un método de aprendizaje no supervisado, y los modelos de entrenamiento se generan igual, 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 neuronal 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:

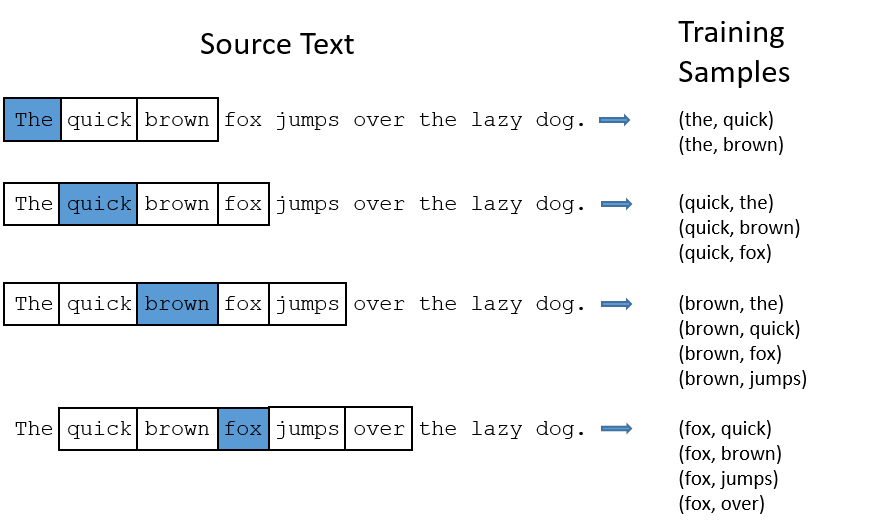


Figura 5: Ejemplo Word2vec []

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.

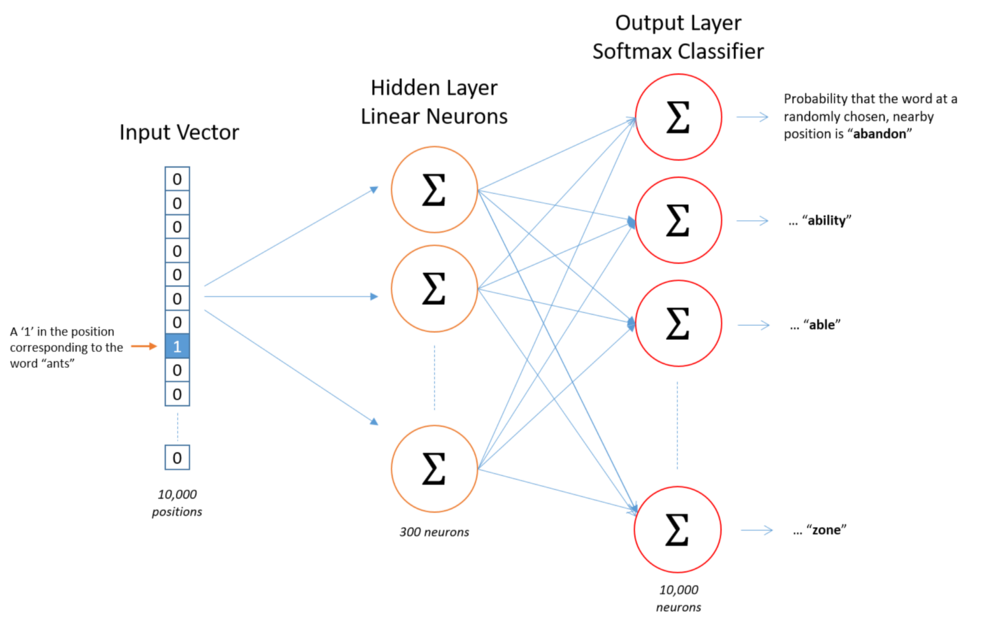


Figura 6: Ejemplo de uso de modelo de word2vec []

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 uso en el dataset que uso con Google News [], aunque este hiperparámetro puede ser modificado a voluntad para buscar mejoras y/u optimizaciones.

Un ejemplo matemático usando:

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

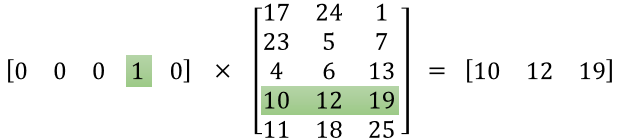


Figura 7: Ejemplo calculo word2vec []

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 [].

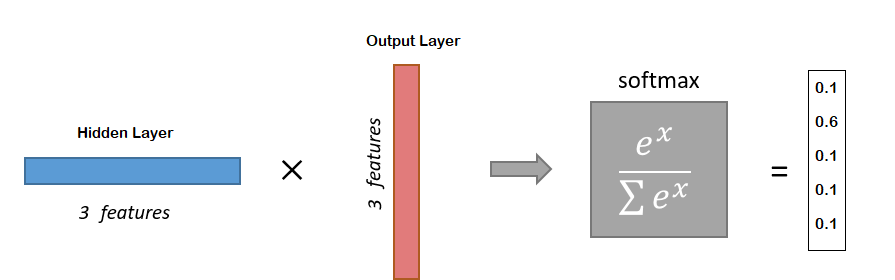


Figura 8: Ejemplo calculo final word2vec

Gramáticas

FALTA ESTE APARTADO

## 2.4 Recomendador

Antes de diseñar el recomendador se deben extraer los datos sobre los que se va a trabajar y revisar los mismos. Para este proyecto se ha usado un dataset de MovieLens [] 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.

*Cabe destacar que toda la información de este dataset se encuentra en inglés.*

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 un ratio de cuan buena es una película, para ello a todas las películas se le aplicará la siguiente formula:

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 esta fórmula y dado un género 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 son 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. []

Siendo:

* t: el termino
* d: el texto
* D: número total de textos
* : 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:

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 en base a 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 pueden 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.

# BOT

**MÉTODOS.**Responde a la pregunta ¿cómo se estudia?  En esta sección se deben presentar los aspectos esenciales para entender los resultados de la investigación. Algunos ejemplos:

* *puede tratarse de una revisión de la literatura sobre un tema, por lo que se deben exponer las principales aportaciones teóricas y relacionarlas con el objeto de nuestro estudio*
* *o un estudio de caso, donde se expondrán las técnicas cuantitativas y cualitativas de toma y análisis de datos*
* *o el desarrollo de una aplicación o un programa informático, que se debe detallar con aspectos técnicos*
* *o un análisis de población, donde se describirá la muestra, las variables, las técnicas seguidas para obtener la información y el procedimiento para la validación de los instrumentos*.

# EVALUACIÓN

RESULTADOS. Responde a la pregunta ¿qué se ha encontrado? En esta sección se presentan de forma descriptiva (no interpretada o discutida) los resultados obtenidos de la investigación descrita en el apartado anterior, mediante texto, y en su caso tablas y figuras.

# GESTIÓN DEL PROYECTO

## 6.1 PRESUPUESTO

## 6.2 PLANIFICAIÓN

## 6.3 ENTORNO SOCIO-ECONÓMICO

El TFG deberá deberá incorporar el desarrollo de los siguientes apartados:

• Presupuesto de la elaboración del TFG.

• Impacto socio-económico (impacto económico, social, medioambiental, ético, etc.) esperado de la aplicación del resultado del proyecto, plan de explotación del mismo, o consideraciones sobre aspectos económicos de la temática del trabajo. Para trabajos teóricos, se debe detallar en qué aplicaciones prácticas podría utilizarse y qué impacto socio-económico podría generar en el sector de aplicación. Nótese que en la rúbrica de TFG se evaluará el ANÁLISIS del impacto socio-económico, no que el impacto sea positivo. Es decir, que se puede obtener la máxima calificación para este apartado con un impacto socio-económico esperado de 0 – o negativo – siempre que el análisis sea correcto. En la ‘Introducción’ debe quedar indicado claramente cuáles son los apartados en los que se reflejan los contenidos del ‘Entorno socio-económico’, siendo recomendable que haya un apartado específico sobre el entorno socio-económico.

## 6.4 MARCO REGULADOR

Marco regulador El TFG deberá incorporar el desarrollo de uno o varios de los siguientes apartados:

• Análisis de la legislación aplicable sobre la implementación descrita en el trabajo (riesgos, responsabilidades profesionales, responsabilidades éticas, riesgos laborales, privacidad y seguridad, etc.).

• Estándares técnicos, si son aplicables (sobre tecnología desarrollada, implantada, sobre lenguajes de programación o herramientas utilizados, etc.).

• Estudio de las cuestiones relacionadas con la propiedad intelectual de la idea (patentabilidad, protección…), por ejemplo, si es un trabajo teórico.

En la ‘Introducción’ debe quedar indicado claramente cuáles son los apartados en los que se reflejan los contenidos del ‘Marco regulador’, siendo recomendable que haya un apartado específico

# CONCLUSIONES

DISCUSIÓN/CONCLUSIONES. Responde a la pregunta ¿qué significa? El apartado final del cuerpo del TFG debe contener las conclusiones que dan respuesta a las preguntas planteadas en la introducción del trabajo -particularmente el objetivo, con el que debe mantener una coherencia- a partir del análisis e interpretación de los datos presentados en el apartado anterior. Opcionalmente puede incluirse un apartado de discusión, que será reflexión final basada en los argumentos expuestos en el trabajo, que tenga una aportación personal del autor aunque sin cargas subjetivas -ideológica o moral, por ejemplo- que valore los resultados alcanzados, reconozca las limitaciones y dificultades encontradas, y esboce una propuesta de una hipotética investigación futura.

# TRABAJO FUTURO

# BIBLIOGRAFIA

La**Bibliografía** o **Lista de referencias**debe figurar a continuación del cuerpo del trabajo*.*En este apartado se recogen los detalles de aquellas fuentes de información, tanto en formato impreso como electrónico, que han sido previamente citadas en el cuerpo del trabajo.  
  
La ordenación de las referencias dependerá del tipo de sistema de citación utilizado en el trabajo:

Cuando el sistema de citación sea Nombre y fecha, las referencias se ordenarán en orden alfabético.

Cuando el sistema de citación sea numérico, las referencias se ordenarán por orden de aparición en el texto.

# ANEXO