

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

GRADO EN INGENIERÍA INFORMÁTICA

CURSO ACADÉMICO 2018/2019

TRABAJO FIN DE GRADO

**META CHATBOT: CHATBOT PARA CREAR CHATBOTS**

Autor: Antonny Gerald Chumpitaz Rios

Director: Francisco Domínguez Mateos

AGRADECIMIENTOS

Después de unos intensos meses de trabajo por fin ha llegado el día en el que veo los frutos de mi curiosidad por la programación, investigación…

He podido aprender nuevas tecnologías, a dar soluciones a problemas que se planteaban a medida que realizaba el proyecto, a ver que siempre hay una forma de resolverlos, aunque no siempre de la forma que se esperaba ni la más óptima.

Primero, me gustaría a gradecer a mi tutor por los consejos que me ha dado y la paciencia que tuvo durante estos meses ya que no siempre he podido cumplir los hitos que nos planteábamos por tema de falta de tiempo al estar trabajando, por buscar tiempo para poder resolver mis dudas y plantearme nuevos desafíos para que pueda conseguir realizarlos y así mejorar el proyecto que teníamos en mano.

Agradecer a mi familia por el apoyo que me dieron siempre, aconsejándome a no desistir cuando me encontraba bloqueado, por las conversaciones ajenas al trabajo que me ayudaron a entender más cosas sobre mi futuro.

Muchas gracias…

RESUMEN

El desarrollo de este proyecto está centrado en los ChatBots. Estos son sistemas inteligentes que permiten interaccionar, mediante voz o texto, con usuarios, simulando el comportamiento humano.

Los ChatBots tienen muchas aplicaciones ya que permiten una interacción persona maquina más sencilla y fluida. Por ejemplo, sistemas de atención al usuario, en webs, sistemas de ayuda telefónica, etc.

En este caso, se ha definido un formato de representación de ChatBots y se ha desarrollado un *Motor de ChatBots,* de tal manera que se puede procesar cualquier ChatBot que se desee. La técnica utilizada está basada en redes neuronales, intenciones, sacos de palabras y acciones (estos conceptos se detallarán en la memoria). Para probar dicho motor, se creó un sencillo ChatBot de gestión de Lista de la Compra.

Lo anterior nos ha permitido crear un tipo especial de ChatBot, un *Meta ChatBot*, que permite crear ChatBots interactuando con él y generar modelos listos para ser procesador por el *Motor de ChatBots*.

Finalmente se ha añadido la posibilidad de ir mejorando de forma continua e interactiva el comportamiento de los ChatBots, modificando el *Motor de ChatBots* y añadiendo un nuevo ChatBot llamado *Solve Error* que permite detectar y solucionar problemas, mejorando así la interacción.

A lo largo de este proyecto, se ha podido adquirir conocimientos del uso y aplicación de nuevas tecnologías como la programación avanzada en Python, paquetes de código para Inteligencia Artificial tales como Keras, TensorFlow, etc., nuevas nociones de programación modular, el poder gestionar mejor el tiempo para desarrollar y mejorar el proyecto, hacer llamadas a servicios para realizar la interacción por voz.

**Palabras claves :** ChatBot, sistemas inteligentes, Motor de ChatBots, redes neuronales, Meta ChatBot, Solve Error.

CONVERSIONES

A lo largo de la memoria se mostrará distintas tipologías a la hora de escribir:

* Las fuentes, como páginas web, libros, revistas, etc., se mostrarán subrayado.
* Los nombres de autores, empresas se mostrarán con las letras en negrita.
* Títulos de proyectos, trabajos, artículos se mostrarán entre comillas.
* Los nombres de aplicaciones, programas, etc., se mostrarán en mayúsculas.
* Cita de los autores o texto proveniente de alguna fuente se mostrarán entre comillas y en cursiva.
* El código fuente se mostrará con letra cursiva *Courier New* y en un párrafo.

ÍNDICE

[ÍNDICE DE ILUSTRACIONES Y TABLAS 8](#_Toc527292422)

[CAPÍTULO 1: INTRODUCCIÓN 9](#_Toc527292423)

[1.1 INTRODUCCIÓN 9](#_Toc527292424)

[1.2 ESTRUCTURA 12](#_Toc527292425)

[1.3 OBJETIVOS 12](#_Toc527292426)

[CAPÍTULO 2: ESTADO DEL ARTE 13](#_Toc527292427)

[2.1 PANORÁMICA 13](#_Toc527292428)

[2.2 COMPARATIVA 15](#_Toc527292429)

[2.3 CRÍTICA DEL ESTADO DEL ARTE 17](#_Toc527292430)

[2.4 PROPUESTA 17](#_Toc527292431)

[2.5 RESUMEN DEL CAPÍTULO 18](#_Toc527292432)

[CAPÍTULO 3: OBJETIVOS 19](#_Toc527292433)

[CAPÍTULO 4: IMPLEMENTACIÓN DEL SISTEMA 20](#_Toc527292434)

[4.1 LISTA DE LA COMPRA 20](#_Toc527292435)

[4.1.1 Análisis 20](#_Toc527292436)

[4.1.2 Diseño 21](#_Toc527292437)

[4.1.3 Pruebas 21](#_Toc527292438)

[4.2 MOTOR DE CHATBOTS 22](#_Toc527292439)

[4.2.1 Análisis 22](#_Toc527292440)

[4.2.2 Diseño 23](#_Toc527292441)

[4.2.3 Pruebas 23](#_Toc527292442)

[4.3 META CHATBOT 24](#_Toc527292443)

[4.3.1 Análisis 24](#_Toc527292444)

[4.3.2 Diseño 25](#_Toc527292445)

[4.3.3 Pruebas 26](#_Toc527292446)

[4.4 Solve Error 26](#_Toc527292447)

[4.4.1 Análisis 26](#_Toc527292448)

[4.4.2 Diseño 27](#_Toc527292449)

[4.4.3 Pruebas 28](#_Toc527292450)

[4.5 RESUMEN DEL CAPÍTULO 28](#_Toc527292451)

[CAPÍTULO 5: EVALUACIÓN CUALITATIVA 29](#_Toc527292452)

[CAPÍTULO 6: RECURSOS EMPLEADOS 30](#_Toc527292453)

[6.1 MATERIALES 30](#_Toc527292454)

[6.2 TABLAS DE COSTES 31](#_Toc527292455)

[CAPÍTULO 7: CONCLUSIONES 33](#_Toc527292456)

[CAPÍTULO 8: TRABAJO FUTURO 34](#_Toc527292457)

[GLOSARIO 35](#_Toc527292458)

[BIBLIOGRAFÍA 37](#_Toc527292459)

# ÍNDICE DE ILUSTRACIONES Y TABLAS

Tabla 1: Comparación de entornos de creación de ChatBots 16

Tabla 2: Tabla de gastos monetarios. 31

Tabla 3: Tabla de costes temporales por mes. 31

Tabla 4: Costes finales del proyecto. 32

# CAPÍTULO 1: INTRODUCCIÓN

En este capítulo se introducirá el concepto de ChatBots así como sus características, aplicaciones, modos de uso, tecnologías, etc.

También se dará una visión general de la estructura de este proyecto al igual que los objetivos.

## 1.1 INTRODUCCIÓN

Un ChatBot es un programa informático inteligencia capaz de interaccionar con otras personas o aplicaciones de distintas formas y en distintos medios.

Estos programas pueden encontrarse en muchas sectores, por ejemplo:

* **Marketing:** los ChatBots tienen el objetivo de informar a los clientes o los futuros clientes de los productos de la empresa, ofertas, descuentos, etc.
* **Captación de leads:** en este sector, los ChatBots son capaces de contactar con personas por distintos medios para ofrecerles los servicios de la empresa con el fin de convencerlos de adquirir dichos servicios.
* **Redes sociales:** tienen como función el mantener el contacto con distintas personas, dar información, etc. Se suelen usar para hacer tareas similares y de gran volumen. Tales como responder a preguntas similares, realizar comentarios diarios a la misma hora, etc.
* **e-commerce:** se asemeja a un asistente de compras que acompaña a los clientes durante el proceso de compra. Un ejemplo es el ChatBot de **H&M**, que ayuda al usuario a buscar lo que quiere con sólo pocas preguntas ahorrando así, tiempo para el usuario.
* **Soporte:** estos ChatBots son una buena opción para las empresas que busquen dar un servicio de ayuda más rápido, mejorando la experiencia del usuario, reduciendo la cola de peticiones, tiempo de espera, etc.
* **Atención al usuario:** muchos de estos ChatBots ya están implementados y funcionando en el sector médico, comercio online, hostelería, investigación, etc. Ayudando al usuario a conseguir la información necesario, realizar pedidos, reservas, etc.
* **Asistentes:** estos ChatBots se pueden encontrar tanto en dispositivos electrónicos como móviles, ordenadores, neveras, sistemas de hogares, etc.  
  Algunos ejemplos más destacados son de los asistentes que se encuentran en los móviles y ordenadores como SIRI, CORTANA, siendo estos asistentes personales; o los que están en los sistemas del hogar como ALEXA.
* **En el trabajo:** en este sector, los ChatBots son más específicos para realizar tareas concretas que ayuden a los trabajadores interaccionando con otros sistemas informáticos.

**Entornos de ejecución de ChatBots**

Las interacciones que ofrecen los ChatBots suelen ser en entornos de texto por medio de chats, procesadores de textos, correos electrónicos, aplicaciones de mensajería, etc. Pero también se pueden encontrar en entornos de audio, usando tecnologías de reconocimiento de voz y text to speech, que son capaces de transformar el lenguaje humano a cadenas de texto comprensibles, en este caso, para el ChatBot; posteriormente las respuestas del ChatBot son producidas por estos entornos de voz para que lleguen en formato de audio a las personas.

Cabe destacar que los ChatBots, en su mayoría, son ejecutados en aplicaciones de mensajería por el gran uso de las mismas por los usuarios.

**Puntos fuertes**

Los ChatBots pueden utilizarse en numerosas aplicaciones o entornos teniendo ciertas ventajas:

* Pueden entender lo que dicen las personas tanto por texto escrito o hablado e interpretar su significado, aunque no siempre es preciso.
* Puede buscar información basándose en los conocimientos obtenidos previamente por experiencias y ofrecerlos a las personas.
* Son un apoyo importante para las aplicaciones modernas de móviles inteligentes.
* Facilitan las tareas monótonas y repetitivas.

**Puntos débiles**

Aunque los ChatBots estén en constante desarrollo presentas ciertos puntos débiles:

* Aún no son precisos.
* Necesitan muchos datos para poder ser un poco estables.
* Necesitan numerosas actualizaciones.
* Programación complicada.

De acuerdo a la tecnología que usen, pueden estar basados en:

* **Reglas o comandos**: Utilizan palabras claves que son reconocidos por el ChatBot y su heurísticas. Estas reglas tienen que ser programadas a mano, limitando a este tipo de ChatBots.

Su funcionamiento se basa en buscar entre su banco de palabras coincidencias con la petición del usuario y el contexto, y dar una respuesta que tiene definida.

No pueden generar nuevo texto, sólo pueden dar respuesta dentro de un contexto y limitadas a las respuestas previamente definidas, sin poder realizar funciones que no estén programadas.

* **Inteligencia Artificial**: Son capaces de reconocer el contexto de la conversación, recordar conversaciones antiguas, responder a preguntas ambiguas, pueden crear textos desde cero con el uso del procesamiento del lenguaje natural, aprenden de todos los casos que se ha dado en el pasado para mejorar sus respuestas.

Dentro de la Inteligencia Artificial se encuentran distintas técnicas de Machine Learning

* **Supervisada:** esta técnica utiliza datos etiquetados diciéndole qué tipo de dato es, después de un periodo de entrenamiento puede clasificar nuevas entradas sin etiquetarse que no han sido introducidas en la etapa de entrenamiento llegando a generalizar nuevos datos.
* **No supervisada:** esta técnica no utiliza datos etiquetados. Tiene como objetivo aprender e inducir los patrones de los datos usando técnicas como el algoritmo de agrupamiento (clustering).
* **Semi-supervisado:** emplean pocos datos etiquetados y muchos datos sin etiquetar para el entrenamiento. Extraen la información de los datos sin etiquetar generando modelos que se adecuen mejor a los datos etiquetados.
* **Aprendizaje por refuerzo:** esta técnica es una de las más interesante ya que no necesita numerosos datos de entrenamiento. Aplica una técnica de prueba y error, se le da al sistema un “premio” si acierta y en caso contrario se le “castiga”, el sistema siempre buscará recibir el “premio” por lo cual después de numerosos intentos fallidos aprenderá a conseguir su objetivo, el premio.

Por lo explicado anteriormente, se entiende que ambas tecnologías presentan desventajas y ventajas. Los ChatBots por comandos dan una respuesta más confiable y comprensibles, pero no responden a preguntas fuera de su conocimiento de respuestas.

Por otro lado, los ChatBots basados en las técnicas de Machine Learning necesitan una gran cantidad de datos o numerosos entrenamientos para su aprendizaje y tienen muchas dificultades para formar respuestas, pero son capaces de dar una respuesta a preguntas que no estén en su base de conocimiento.

Este proyecto se basará en un ChatBot que utiliza la tecnología de inteligencia artificial apoyándose en la técnica de aprendizaje supervisado para crear otros ChatBots.

Se irán detallando los conceptos mencionados previamente así como las herramientas usadas.

## 1.2 ESTRUCTURA

El contenido de desarrollará en distintos aparatados, empezando por el estado del arte donde se explicará la situación actual sobre los ChatBots presentando algunas críticas y algunas propuestas para su mejora.

Seguida de las hipótesis del proyecto, es decir, los objetivos que el proyecto pretende conseguir para mejorar algunos puntos respecto del estado actual de los ChatBots.

Después de estos apartados de detallará las soluciones que se han desarrollado para lograr los objetivos, detallando por separado cada solución con sus respectivas imágenes, herramientas, diseños, etc.

Además, se planteará las conclusiones del proyecto resumiendo lo que se ha logrado con el desarrollo del mismo, para luego terminar mencionando las posibles mejoras en el apartado de trabajo futuro.

## 1.3 OBJETIVOS

Este proyecto tiene como objetivo conseguir un sistema inteligente, Meta ChatBot, capas de reconocer una estructura utilizando un motor de ChatBots para poder crearlos y ejecutarlos. Teniendo como posibles medios de interacción la voz o texto, dándole una mayor versatilidad. Este Meta ChatBot podrá generar ChatBots con las mismas posibilidades de interacciones y dará al usuario la capacidad de modificar el flujo del motor de ChatBots para terminar ejecuciones o indicarle de algún fallo para luego utilizar un ChatBot auxiliar que estará a disposición del usuario para poder mejorar cualquier ChatBot.

# CAPÍTULO 2: ESTADO DEL ARTE

En este capítulo se hablará del estado actual de los ChatBots, presentando comparaciones, un apartado de crítica de lo visto en la actualidad y unas propuestas que llevarán al desenlace de este proyecto.

## 2.1 PANORÁMICA

Un ChatBot es un sistema capaz de mantener una conversación natural con una persona, ChatBot o sistema. Está característica es gracias a que ha sido diseñado, programado y entrenado para interpretar las intenciones de los usuarios, entender sus respuestas y en base a ello decir qué responder o qué acción ejecutar. El poder entender y procesar un diálogo es debido a tecnologías NLP (Natural Language Processing, o en español Procesamiento del Lenguaje Natural) y al usar Inteligencia Artificial.

En la actualidad los ChatBots se encuentran en numerosos sectores prestando servicios de diferentes formas.

Muchos de ellos implementados de una forma específica, teniendo como objetivo realizar tareas que no implican la evolución o mejora a grandes rasgos de su arquitectura.

Otros, en cambio, realizan tareas que reclaman que el ChatBot sea flexible para poder generalizar los resultados.

Como se ha mencionado en anteriores apartados, una de las tecnologías con las que se implementan es la Inteligencia Artificial que les permite tener la flexibilidad que algunas tareas requieren.

Muchas aplicaciones que permiten crear ChatBots están dirigidos a que estos se utilicen en aplicaciones de mensajería, lo que facilita la posibilidad de interacción con más usuarios.

Otros están dirigidos a entornos en el que utilizan interfaces conversacionales, algunos de loas más conocidos son APPLE HOMEPOD de **Apple**, GOOGLE HOME de **Google**, ALEXA de **Amazon** que liberó su código en el 2015 para que se pueda añadir nuevas características a su sistema, pudiendo así, mejorarlo a la necesidad del usuario.

Numerosas empresas están apostando por este tipo de interfaces que es una evolución de la forma de usar el software en internet, ofreciendo soluciones en este sector para que muchas startups aprovechen sus infraestructuras para construir el ecosistema actual de aplicaciones basadas en Inteligencia Artificial.

Algunos ejemplos más destacados son:

**IBM**: su tecnología de computación cognitiva ofrece un conjunto de servicios en modo SaaS que se centran en el procesamiento de lenguaje, como es Watson Conversation Service, permitiendo definir intenciones y entidades.

No siempre hace uso de servicios como Alchemy Language o Natural Language Classifier. También ofrece tecnologías propias para el desarrollo de funciones concretas que pueden requerir los ChatBots como son Speech to Text y Text to Speech, Watson Virtual Agent que fue diseñado para el desarrollo de ChatBots con los que ofrecer experiencia conversacional.

**Microsoft:** dentro de su plataforma de cloud Azure ofrece una tecnología pensada para el desarrollo de Bots y por otro lado la tecnología Language Understanding Intelligence Service a la que han llamado LUIS, como una serie de servicios alrededor del procesamiento de lenguaje natural como es el análisis lingüístico y ofrece un conjunto de herramientas que permite entrenar a la plataforma en modelos de conversación.

Una aplicación de la tecnología de ChatBots desarrollada por la propia empresa Microsoft la encontramos en el proyecto CALENDAR.HELP que ofrece una utilidad para la gestión de agenda y reuniones.

**Google:** ofrece la API Natural Language sobre su plataforma en la nube y es utilizada para descubrir la estructura y el significado del texto usando modelos de aprendizaje automático.

Puede usarse para conocer las opiniones sobre un producto en las redes sociales o analizar las intenciones de los clientes a partir de las conversaciones de un centro de llamadas o una aplicación de mensajería.

Por otro lado, Google cuenta con API.ai que compró en 2016 convirtiéndose en una de las principales tecnologías para el desarrollo de ChatBots por parte de empresas, gracias a su capacidad de procesamiento del lenguaje natural; también ha creado CHATBASE desde la cual se ofrece un servicio de analítica con el cual los desarrolladores pueden analizar los resultados y aprender de las incidencias o errores que puedan aparecer en sus ChatBots.

**Amazon:** ha desarrollado la tecnología Lex como un servicio para crear interfaces de conversación en cualquier aplicación con voz y texto, con funcionalidades de aprendizaje profundo avanzadas del reconocimiento automático del habla.

Gracias a esto se pueden crear aplicaciones con interacciones de conversaciones realistas. A través de este servicio, se encuentra ALEXA a disposición de cualquier desarrollador, para que pueda crear ChatBots conversacionales con un lenguaje natural sofisticado de forma rápida y fácil.

**Facebook:** además tener las plataformas de Messenger y Whatsapp en las cuales se encuentran el mayor número de ChatBots, también compró en 2015 Wit.ai, un servicio por medio del cual los desarrolladores podían crear ChatBots autónomos y que aprende con el tiempo. Para ello se ofrecía una interfaz especialmente diseñada para simplificar el proceso de creación del bots, y varios SDK y API. Pero se ha anunciado que el servicio únicamente se ofrecerá para el procesamiento del lenguaje natural.

Otras aplicaciones alternativas son:

**Chatfuel**: trabaja sobre tecnología NLP. Se necesita establecer las reglas a través de su interfaz para que el ChatBot vaya reconociendo las frases y pueda reconocer las variaciones de los usuarios al utilizarlas.

Aunque es gratuita, tiene una versión de pago que ofrecen complementos como la posibilidad de realizar chat en vivo.

**Pandorabots**: es una empresa de inteligencia artificial que presta servicios web para construir y desarrollar ChatBots, siendo una de las más antiguas y grandes del mundo.

Es compatible con el nuevo AIML 2.0 como lenguaje de marcado de contenido de conocimiento, y muchas de sus fortalezas provienen de la potencia y la sencillez de AIML y un motor de análisis y traducción basado en Lisp.

**MobileMonkey**: en el mes de junio del 2018, la empresa de Chattypeople fue adquirida por MobileMonkey que pretende crecer dentro de la plataforma de Facebook Messenger ofreciendo su herramienta para crear ChatBots.

Estos tienen una función llamada "Chat Blast", que es la capacidad de enviar mensajes a múltiples usuarios, podría ser a todos sus usuarios o un segmento en particular.

Esta característica es, sin duda, valiosa para los vendedores aunque no es la única herramienta con esta característica. Por ejemplo, Chatfuel tiene una característica similar que llaman "transmisión".

De manera similar, la función de chat en vivo de MobileMonkey es similar a la de Chatfuel, disponible a través de un complemento.

**Plataformas de ejecución de ChatBots**

La mayor parte de empresas han mejorado sus plataformas de mensajerías para ofrecer entornos de trabajo sobre los cuales se pueden ejecutar los ChatBots. Algunas de estas aplicaciones además han creado ChatBots Stores y sistemas avanzados de búsqueda y valoración de ChatBot, con el objetivo de que los desarrolladores sean quienes generen las utilidades que potencien el uso de dichas aplicaciones.

Las principales plataformas de mensajería a nivel mundial en las que se encuentran los ChatBots son:

* Messenger
* Telegram
* Slack
* Skype
* Kik
* Wechat

Aunque la plataforma de mensajería es la más conocida, no es la única en donde se pueden encontrar los ChatBots. Como se han explicado en anteriores apartados y capítulos, los ChatBots pueden estar dirigidos a diferentes áreas.

Existen otras plataformas en las que los ChatBots pueden cumplir con su rol asignado. Por ejemplo, se encuentran en la propia web de una tienda online, dando soporte para solicitar alguna cita médica, en sistemas inteligentes en nuestros móviles y dispositivos del hogar, dando apoyo en sistemas implantados en centros de salud, investigación, etc.

## 2.2 COMPARATIVA

A continuación se llevará a cabo una comparativa entre ChatBots creados con las herramientas que se han especificado en el anterior apartado y el sistema implementado con este trabajo.

Las características a tratar serán:

1. Comunicación con otros sistemas, aplicaciones, etc. (Carct.1)
2. Uso en plataformas de mensajería. (Carct.2)
3. Interacción por texto del sistema. (Carct.3)
4. Interacción por voz del sistema. (Carct.4)
5. Interfaz para su uso. (Carct.5)
6. Permite programación básica del código. (Carct.6)
7. Permite programación avanzada del código. (Carct.7)
8. Facilidad de uso. (Carct.8)
9. Permite probar el ChatBot creado. (Carct.9)
10. Mejora del ChatBot. (Carct.10)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Carct.1 | Carct.2 | Carct.3 | Carct.4 | Carct.5 | Carct.6 | Carct.7 | Carct.8 | Carct.9 | Carct.10 |
| **IBM Watson Conversation** | X | X | X |  | X | X |  | X | X | X |
| **Microsoft Azure Bot Service** | X | X | X |  | X | X | X |  | X | X |
| **Google API.ai** | X | X | X |  | X | X |  | X | X | X |
| **Amazon Lex** | X | X | X |  | X | X | X | X | X | X |
| **Facebook Messenger API** | X | X | X |  |  | X | X |  | X | X |
| **Chatfuel** |  | X | X |  | X | X |  | X | X | X |
| **Pandorabots** |  | X | X |  | X | X |  |  | X | X |
| **APPSUMO MobileMonkey** | X | X | X |  | X | X |  | X | X | X |
| **Meta ChatBot** | X |  | X | X |  | X | X |  | X | X |

Tabla 1: Comparación de entornos de creación de ChatBots

Como se puede apreciar, el sistema que se ha construido en este proyecto ofrece una característica que no presentan los otros entornos con los que se ha comparado.

Siendo éste uno de los objetivos para dar mayor facilidad al usuario a la hora de crear ChatBots.

## 2.3 CRÍTICA DEL ESTADO DEL ARTE

En la actualidad las empresas están enfocando el desarrollo de entornos donde se puedan crear ChatBots para que sean usados por cualquier persona sin tener que programar y que los ChatBots creados funcionen en aplicaciones de mensajería y así llegar a más usuarios.

Las herramientas que las empresas utilizan u ofrecen para crear ChatBots presentan una interfaz con muchas características, aumentando la curva de aprendizaje. Algunas de estas herramientas están diseñadas para que el usuario no tenga que aprender a programar, lo que es una gran ventaja aunque limita mucho a los programadores que optan por usar aquellas herramientas que admiten programación un poco avanzada, pero su interfaz es confusa.

Muchos de los ChatBots que se encuentran en distintas plataformas son sencillos, es decir, no tienen una implementación profunda con tecnologías de Machine Learning. Teniendo una interacción sólo por texto sin poder reconocer las peticiones ambiguas de las ya definidas.

También se puede encontrar ChatBots que utilizan Machine Learning haciéndolos más versátiles a la hora de aprender.

Un entorno para crear ChatBots debería facilitar la creación de los mismos, ofreciendo distintas formas de interactuar con él, ya sea por texto o voz. Ser capaz de dar la posibilidad de mejorar los ChatBots creados, entrenarlos, probarlos, programar algunas características, eficiente en tiempo de ejecución; creando, así, ChatBots flexibles y programables que usen técnicas de Machine Learning que puedan ser usados tanto en plataformas de mensajería, webs, sistemas del hogar, trabajo, etc.

El objetivo sería crear ChatBots totalmente funcionales en distintas plataformas, programables, accesibles a cualquier perfil.

Con este trabajo se pretende crear un entorno, que en sí es un ChatBot, capaz de ser utilizado por voz o texto. Aunque no está enfocado para que los ChatBots creados por este entorno se puedan usar en plataformas de mensajería, se pueden usar en distintos sistemas de trabajo ya que la estructura es programable.

## 2.4 PROPUESTA

Con todo lo explicado anteriormente se deduce que un buen entorno de creación de ChatBots tiene que ofrecer la flexibilidad de aprendizaje, diversos tipos de interacción con los usuarios, dar la posibilidad de reprogramar los ChatBots creados, aportar un soporte.

Este trabajo tiene como objetivos proporcionar distintas formas de interacción, para que el usuario se sienta más cómodo y sea más rápido el poder crear ChatBots, dar la libertad de uso del entorno al usuario a la hora de mantener su ChatBot, afianzar la idea de que un ChatBot es un sistema el cual necesita una interacción humana para poder realizar sus tareas.

Estas interacciones persona-máquina irán evolucionando con la tecnología y hará que este tipo de sistema sean aún más usadas y cómodas.

## 2.5 RESUMEN DEL CAPÍTULO

Los entornos que ofrecen la posibilidad de crear ChatBots facilitan a usuarios comunes y otros a programadores para generar ChatBots.

Pero no hay que quedarse con la idea que los ChatBots sólo están en las plataformas de mensajería. Se ha llegado a una era en la que se vive rodeado de sistemas inteligentes que observa el comportamiento de las personas para obtener datos y llevar a cabo, con mayor precisión, su objetivo.

# CAPÍTULO 3: OBJETIVOS

En este capítulo se dará una visión global de lo que tratará el proyecto enfocado a los objetivos a conseguir y se hablará de cómo es la estructura del proyecto.

El proyecto se enfocará en conseguir un sistema con las siguientes características:

1. Capaz de comunicarse con los usuario por texto y voz.
2. Tiene que reconocer una estructura de ChatBots para su procesamiento.
3. Debe proporcionar un servicio de mejora.
4. El usuario podrá realizar cambios programando el sistema.
5. El sistema tiene que generar los ficheros necesarios automáticamente.
6. Cada vez que se mejore el sistema, se volverá a entrenar el ChatBot que se ejecute.
7. Se podrá testear el ChatBot que se esté creando.

A continuación se especificará el funcionamiento general del sistema y una visión global de su arquitectura.

El sistema completo está compuesto por tres distintos sub-sistemas que se complementan para ofrecer un mejor servicio al usuario. Estos sub-sistemas son el motor de ChatBots, Meta ChatBot y Solve Error.

Cuando el usuario ejecute el programa, entra en funcionamiento el Meta ChatBot. Éste a su vez usará el motor de ChatBots para reconocer la estructura de los ChatBots.

El usuario al indicar por texto o voz sus intenciones, el Meta ChatBot lo procesará con ayuda del motor de ChatBot para dar una respuesta o ejecutar una acción.

Al finalizar todas las tareas que el usuario haya indicado, éste podrá generar el ChatBot que está construyendo haciendo que el Meta ChatBot genere todos los archivos necesarios, modelo, código, etc.

El usuario también podrá probar el ChatBot creado desde el Meta ChatBot.

Por otro lado, si al probar su ChatBot no reconocer alguna intención podrá mejorarlo usando el ChatBot Solve Error que tras mejorar el ChatBot actualiza el modelo volviéndolo a generar automáticamente, o si lo prefiere el usuario, puede programar directamente la estructura del ChatBot para que pueda reconocer las intenciones fallidas y generar el modelo a mano.

# CAPÍTULO 4: IMPLEMENTACIÓN DEL SISTEMA

En este capítulo se especificará las herramientas, diseño, pruebas, arquitectura, etc., de cada sub-sistema: motor de ChatBots, Meta ChatBot y Solve Error.

De igual forma se hablará del ChatBot “Lista de la Compra” que ha servido de ayuda para entender el funcionamientos de los ChatBots.

En todos los ChatBots se utiliza una estructura hecha con JSON y el lenguaje de programación Python. En cada apartado se presentará la estructura con el que se ha desarrollado. De igual forma se mencionarán las herramientas usadas.

## 4.1 ESTRUCTURA EN COMÚN

En este apartado se mostrará la estructura, interacciones de clases, esquemas del sistema que son comunes entre los sub-sistemas. En los siguientes apartados se detallarán las estructuras de los sub-sistema para dar una visión más profunda de la implementación del sistema.

### 4.1.1 JSON

A continuación de muestra una imagen en el que presenta la estructura JSON que tienen todos los ChatBots, en este caso de ha tomado como ejemplo el ChatBot “Lista Compra”.



Ilustración 1: Estructura del ChatBot "Lista Compra"

En la imagen anterior se presenta la estructura que tienen todos los ChatBots del sistema y de los que se crearán.

A continuación se detallará las partes de la estructura común a todos los ChatBots.

Como se puede apreciar, los “**tags**” contienen las intenciones de los usuarios, es decir, la idea de lo que el usuario pretende realizar.

Los “**patterns**” son las petición que el usuario utilizará para indicar su intención al ChatBot.

Los “**responses**” son las respuestas asociadas a las peticiones que el ChatBots mostrará al usuario.

El “**action**” será aquella acción que ejecutará el ChatBot al reconocer la intención del usuario.

### 4.1.2 Diagrama de clases del Sistema

En la siguiente imagen se mostrará la interacción interna del sistema. Se tomará como ejemplo la interacción de un ChatBot. El motor de ChatBots en el diagrama de clases está representado por “**CProcessor**”.

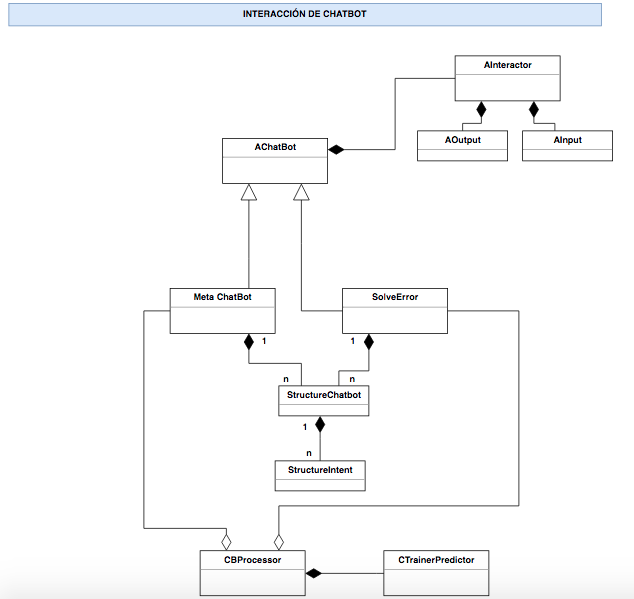


Ilustración 2: Interacción de clases de un ChatBot.

### 4.1.3 Flujo de ejecución

En esta sección se presenta una imagen con el funcionamiento en general y la interacción de las distintas partes del sistema que tiene el flujo de los ChatBots del sistema o creados por él.

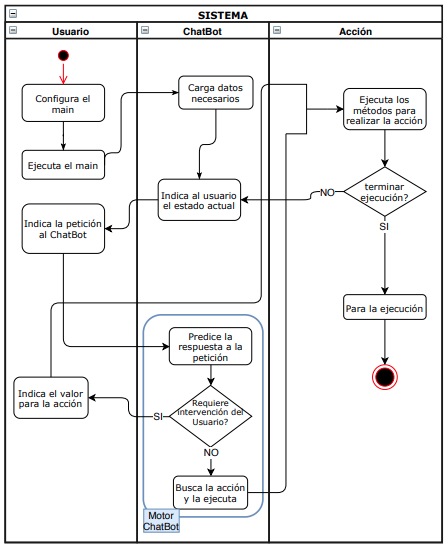


Ilustración 3: Flujo general de un ChatBot.

## 4.2 LISTA DE LA COMPRA

Este ChatBot se ha realizado con el fin de servir de guía para entender el funcionamiento de un ChatBot.

Utiliza el motor de ChatBots para reconocer las peticiones de los usuarios.

### 4.2.1 Análisis

El ChatBot “Lista de la Compra” presenta una estructura sencilla, ya que desde un inicio fue implementado para responder a las intenciones de los usuarios y mostrar los resultados de los mismos, con una interacción sólo por texto.

La funcionalidad de este ChatBot es poder guardar en su estructura el listado de productos que el usuario quiere comprar, para luego editarla según la conveniencia el usuario, para luego realizar la compra o cancelarla.

#### 4.2.1.1 Requisitos Funcionales

Los requisitos funcionales son aquellas funciones que realiza el sistema, ya sea entradas, acciones y salidas.

Entre los requisitos funcionales, se encuentran:

* RF01: Debe permitir añadir nuevos productos, mostrando un mensaje de éxito.
* RF02: Se podrá eliminar los productos ya introducidos, mostrando un mensaje de éxito.
* RF03: Dará un mensaje de error en caso de que no exista el producto a eliminar.
* RF04: Permitirá mostrar todos los productos de la cesta.
* RF05: En caso de realizar la compra, dará opciones de pago.
* RF06: Tras seleccionar el modo de pago se mostrará un mensaje de éxito.
* RF07: En caso de cancelar la compra, eliminará todos los productos guardados mostrando un mensaje de aviso.
* RF08: La interacción será por texto.
* RF09: Todos los mensajes de aviso se visualizarán en pantalla.
* RF10: Se podrá parar la ejecución del ChatBot.

#### 4.2.1.2 Requisitos No Funcionales

Los requisitos no funcionales son las características de funcionamiento, es decir, atributos de calidad del sistema.

Entre los requisitos no funcionales para este ChatBot se encuentran:

* RNF01: El usuario debe saber en cada momento en qué estado se encuentra el sistema.
* RNF02: El lenguaje que se usará para la interacción será el español.
* RNF03: Los mensajes de respuesta deberán ser claros.
* RNF04: El ChatBot no debe tardar más de cinco segundos en dar una respuesta.
* RNF05: La interacción debe ser fácil de realizar.

### 4.2.2 Diseño

El diseño de este ChatBot se basa en la comunicación con usuario para recibir las intenciones y procesarlas con el motor de ChatBots para dar una respuesta.

El lenguaje con el que se implementó es Pythony su estructura está hecha con JSON.

En el ChatBot “Lista Compra” se ha utilizado una estructura basada en JSON que es procesada por el motor de ChatBots para dar una respuesta o ejecutar una acción.

### 4.2.3 Pruebas

En las siguientes imágenes se verá el funcionamiento básico de este ChatBot, mostrando el cumplimiento de requisitos funcionales y no funcionales de algunas intenciones.

## 4.3 MOTOR DE CHATBOTS

Este sub-sistema no es un ChatBot. Es un programa implementado para poder realizar las tareas de reconocimiento de intenciones y de la estructura de los ChatBots.

Siendo el encargado de utilizar técnicas de Machine Learning que se detallarán más adelante.

### 4.3.1 Análisis

El motor de ChatBots implementado para este proyecto es capaz de reconocer las ambigüedades de las intenciones de los usuarios gracias al Procesamiento Natural del Lenguaje (PNL, o en inglés NLP).

El PNL estudia la interacción entre los ordenadores y el lenguaje humano, cuyos modelos se enfocan en el comportamiento cognitivo en general y a la organización de la memoria. Al principio se basaba en un conjunto de reglas hechas a mano, pero en 1980 se introdujo el aprendizaje automático.

Además de reconocer el lenguaje de los humanos, reconoce la estructura que usan los ChatBots del sistema y los creados por dicho sistema.

#### 4.3.1.1 Requisitos Funcionales

Los requisitos funcionales son aquellas funciones que realiza el sistema, ya sea entradas, acciones y salidas.

Entre los requisitos funcionales para este programa se encuentran:

* RF01: Tiene que reconocer las ambigüedades de las peticiones.
* RF02: Procesará las intenciones para dar una respuesta.
* RF03: En caso de no encontrar coincidencias no dará ninguna respuesta.
* RF04: Escogerá la respuesta con mayor porcentaje de acierto.
* RF05: Generará el modelo de los ChatBots.
* RF06: Guardará el modelo generado del ChatBot.
* RF07: En el caso de que el ChatBot no tenga un modelo, lo generará usando la estructura de dicho ChatBot.
* RF08: Se encargará de entrenar los ChatBots.

#### 4.3.1.2 Requisitos No Funcionales

Los requisitos no funcionales son las características de funcionamiento, es decir, atributos de calidad del sistema.

Los requisitos no funcionales del motor de ChatBots son:

* RNF01: Mostrará mensajes del estado en el que se encuentra la ejecución del entrenamiento.
* RNF02: Mostrará un mensaje al finalizar de generar el modelo.

### 4.3.2 Diseño

El lenguaje con el que se implementó es Python.

Este sub-sistema realiza el procesamiento del lenguaje natural (PLN) gracias a la librería NLTK.

Esta librería se utiliza para analizar las intenciones y generar así una lista de posibles entradas y salidas. Estas se consiguen gracias al saco de palabras, que contiene las raíces de las intenciones que están relacionadas a unas respuestas, ambas se encuentran la estructura JSON.

Tras conseguir estas listas de entradas y sus salidas, se hace uso de las redes neuronales con ayuda de KERAS.

Es aquí donde se entrena las neuronas usando las entradas y salidas, la finalizar el entrenamiento se guarda el modelo resultante para su posterior uso.

Cuando se ejecuta el programa, el motor de ChatBots carga el modelo generado del ChatBot en ejecución. Al introducir una petición, realiza una clasificación para hallar la mejor coincidencia con los datos de entrenamiento y poder dar la respuesta correcta.

Si encuentra alguna respuesta, el motor de ChatBots devuelve una acción. Esta acción puede ser devolver la respuesta asociada o desencadenar otras acciones.

En la siguiente imagen se muestra la interacción interna del ChatBot utilizando el motor de ChatBots.

El motor de ChatBot se encarga de comunicarse con el modelo del ChatBot para encontrar la mejor respuesta. Una vez encontrado, comunica al ChatBot qué acción ejecutar y éste lo realiza a través de su estructura.

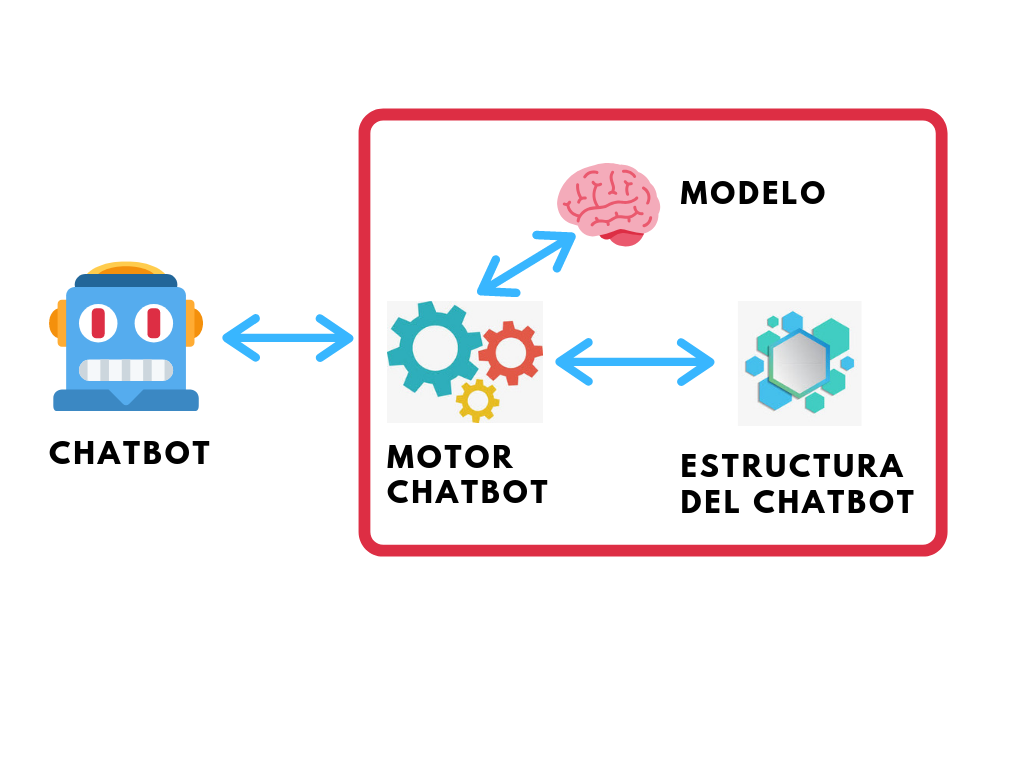


Ilustración 4: Modelo del Motor de ChatBots.

Otra característica de este programa, es que entrena y realiza el modelo del ChatBot para luego utilizarlo para reconocer nuevas entradas de intenciones por parte del usuario. El siguiente esquema muestra este funcionamiento.

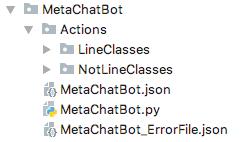
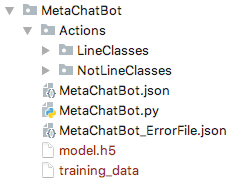


Ilustración 5: Entrenamiento y creación del modelo de un ChatBot.

Para el entrenamiento, el motor de ChatBot lee el JSON del ChatBot. Con esto se genera unos datos necesarios para generar el modelo del ChatBot que luego estará a disposición del motor del ChatBot.

### 4.3.3 Pruebas

En las siguientes imágenes se mostrarán el cumplimiento de algunos requisitos funcionales y no funcionales.



## 4.4 META CHATBOT

Este sub-sistema es el encargado de la interacción con el usuario y el de generar la estructura de los ChatBots.

El Meta ChatBot tiene como objetivo crear ChatBots por voz utilizando la estructura definida en JSON, haciendo uso del motor de ChatBots.

### 4.4.1 Análisis

Desde la ejecución del Meta ChatBot, éste se encargará de cargar todos los ChatBots creados hasta el momento a su lista de ChatBots. Es decir, tiene la capacidad de pasar la estructura JSON a código Python.

Y a medida que se vaya creando un ChatBot, internamente el Meta ChatBot está generando la estructura en JSON para que después se generen los ficheros de código, carpetas, etc., del ChatBot que se esté creando.

Además, proporcionará la opción de ejecutar el ChatBot creado para testearlo y ver si falla.

También es el encargado de llamar a los métodos del motor de ChatBots para que se genere el entrenamiento y modelo del ChatBot que se esté creando.

Ofrecerá la posibilidad de realizar estas tareas con la interacción por voz.

#### 4.4.1.1 Requisitos Funcionales

Los requisitos funcionales son aquellas funciones que realiza el sistema, ya sea entradas, acciones y salidas.

Dentro de los requisitos funcionales para este sub-sistema están:

* RF01: Tendrá interacción por voy o texto.
* RF02: Convertirá la estructura JSON a código Python.
* RF03: Convertirá código Python a estructura JSON.
* RF04: Permitirá crear distintos ChatBots.
* RF05: Permitirá mostrar el ChatBot actual, es decir, el ChatBot que se está editando.
* RF06: Permitirá mostrar todos los ChatBots que se están creando y/o se han creado.
* RF07: Cargará automáticamente todos los ChatBots que se hayan creado a la lista de ChatBots.
* RF08: Permitirá cambiar de ChatBot.
* RF09: Guardará los cambios de un ChatBot en todo momento.
* RF10: Permitirá crear Intenciones para cada ChatBot.
* RF11: Mostrará la Intención actual, es decir, la Intención que se está editando.
* RF12: Mostrará la lista de todas las intenciones de un ChatBot.
* RF13: Podrá mostrar la Acción creada por cada Intención.
* RF14: Podrá añadir, eliminar, cambiar un ChatBot.
* RF15: Podrá añadir, eliminar, cambiar una Intención.
* RF16: Podrá añadir, eliminar una acción.
* RF17: Podrá generar el ChatBot actual creando sus ficheros Python y JSON.
* RF18: Mostrará mensajes indicativos del estado de la creación del ChatBot.
* RF19: Permitirá elegir el ChatBot que desee ejecutar.
* RF20: Permitirá parar la ejecución del ChatBot que ha ejecutado.
* RF21: Podrá guardar las entradas no reconocidas, usando una palabra clave “Guardar sentencia”, en un fichero JSON.
* RF22: Podrá parar su ejecución.
* RF23: Permitirá guardar las sentencias no reconocidas.

#### 4.4.1.2 Requisitos No Funcionales

Los requisitos no funcionales son las características de funcionamiento, es decir, atributos de calidad del sistema.

Entre los requisitos no funcionales se encuentran:

* RNF01: Los mensajes mostrados deben ser claros.
* RNF02: El tiempo de espera no debe ser mayor a cinco segundos.
* RNF03: El lenguaje usado para las interacciones debe ser español.
* RNF04: Debe indicar en qué estado se encuentra el usuario.

### 4.4.2 Diseño

Para que este sub-sistema pudiese generar todos los ficheros necesarios se utilizó código Python que facilita esta tarea. Se implementó métodos que convierten la estructura JSON de los ChatBots en una cadena de texto, con la sintaxis de Python para después incrustar dicha cadena en un fichero Python.

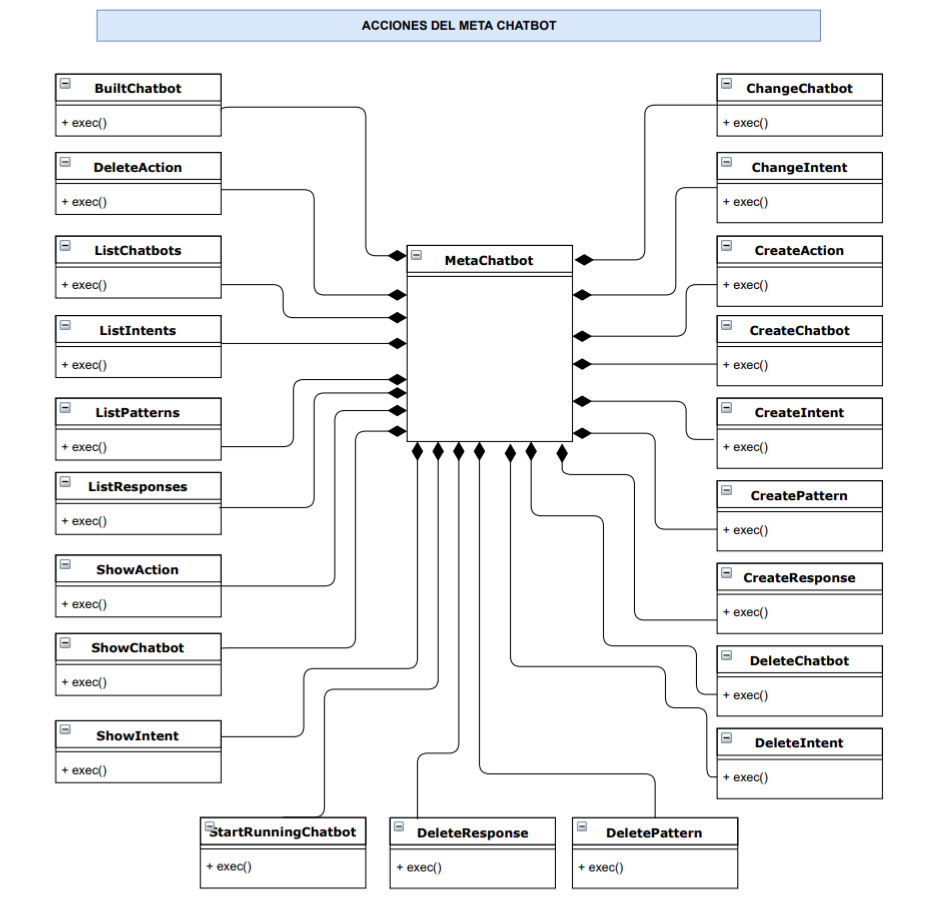
Otra característica es que puede leer y añadir caracteres especiales a la estructura JSON, lo que se implementó con el método *“json.dump()”* de la librería JSON de Python, añadiendo ciertos parámetros para que guardase los caracteres especiales.

El Meta ChatBot guarda la estructura de los ChatBots en objetos que mantienen la información, gracias a estos objetos el Meta ChatBot es capaz de añadir, editar y eliminar nuevos ChatBots, intenciones, respuestas y acciones, según lo vaya solicitando el usuario.

Este sub-sistema, al igual que los otros ChatBots, puede comunicarse por voz. Esto lo realiza gracias a un servicio XML-RPC.

El SimpleXMLRPCServer hace uso del threading para comunicar el host, previamente configurado con el código, con el cliente. Puede hacerce con un móvil desde el cual el usuario podrá ejecutar un script que enviará la petición al puerto del thread para ejecutar el código del módulo de entrada, que responderá ejecutando un código en el módulo de salida para que se produzca una respuesta con audio.

En la siguiente imagen se muestra un diagrama con las acciones implementadas.



Todas estas acciones permiten el cumplimiento de los requisitos especificados anteriormente.

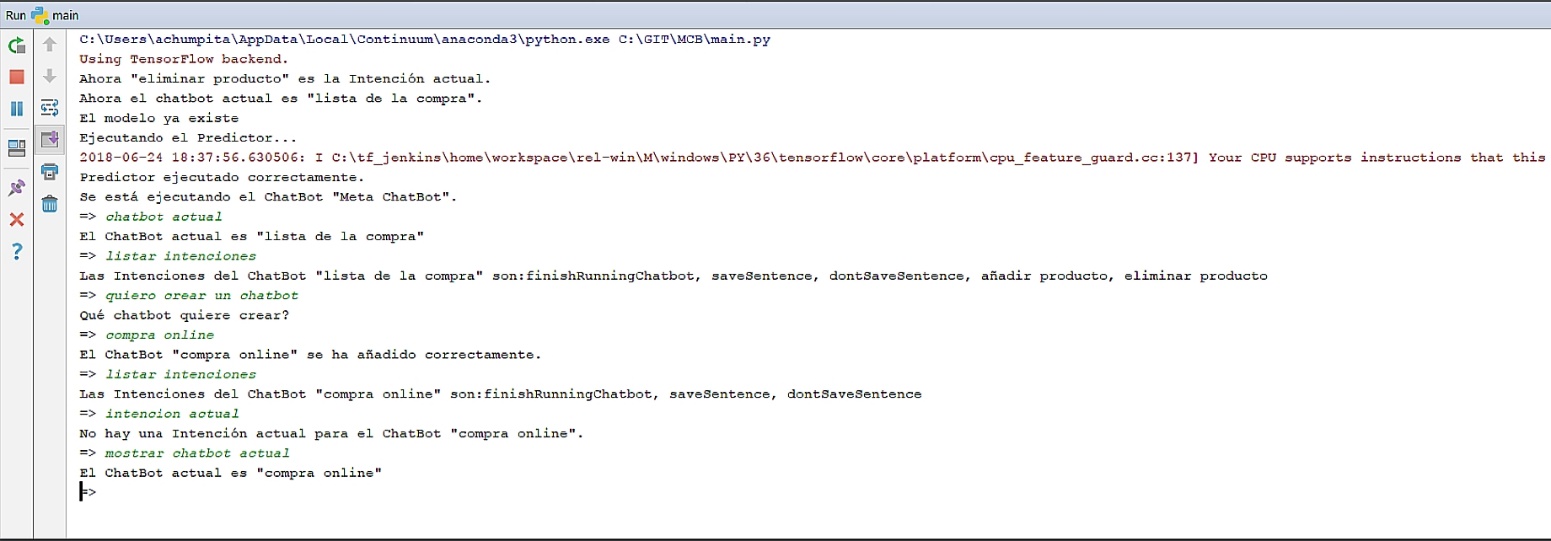
### 4.4.3 Pruebas

En las siguientes imágenes se mostrarán algunos ejemplos de la funcionalidad lograda que cumplen con los requisitos funcionales y no funcionales.

Las peticiones del usuario estarán en color verde y las respuestas del sistema en color negro.

* Crear el ChatBot “compra online”

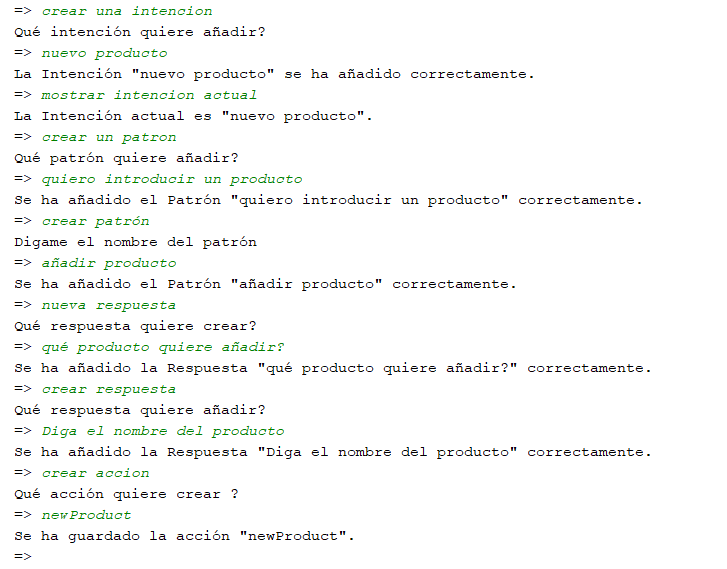
Para la ejecución de este ejemplo, en el sistema ya existe otro ChatBot, “lista de la compra”. Se mostrará la interacción teclado/pantalla.



Como se puede ver, las acciones implementadas han sido ejecutadas gracias a que se ha podido reconocer la sintáxis del lenguaje.

* Crear una Intención, Patrones, Respuestas y Acción:

Este ejemplo es la continuación del anterior.

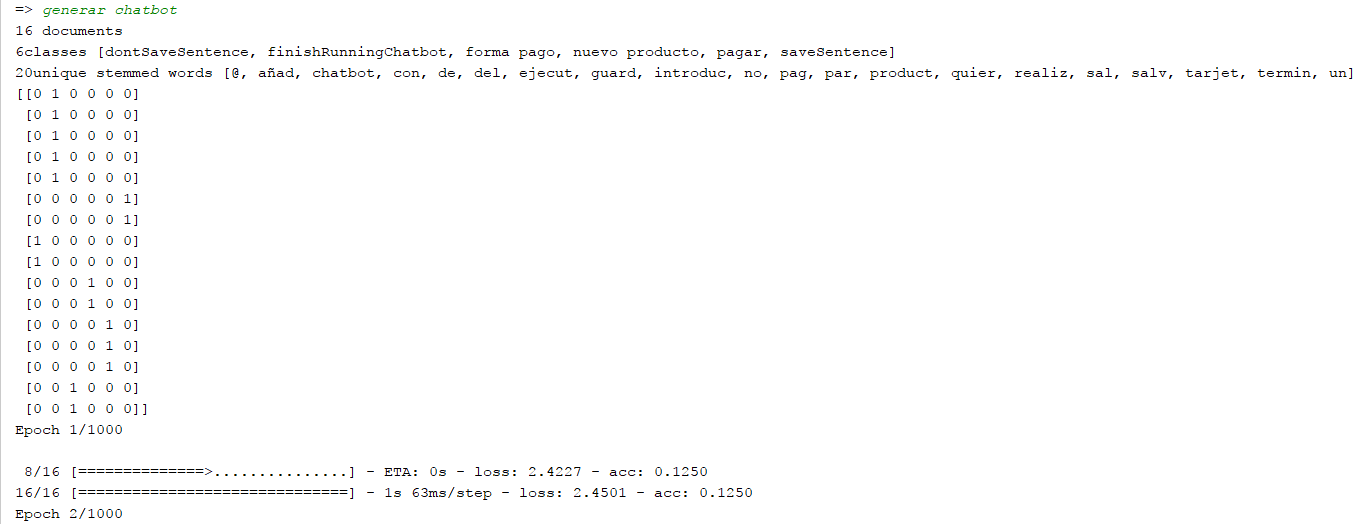


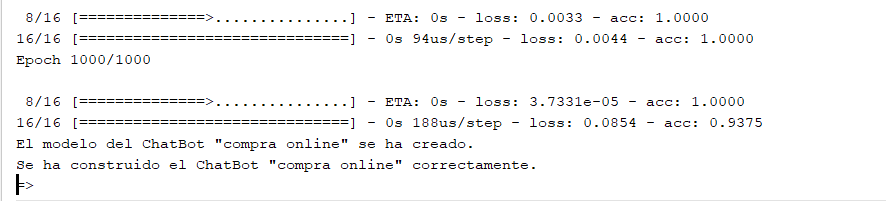
El sistema reconoce distintas formas de patrones y puede dar una respuesta al azar entre la lista de Respuestas. A su ves, puede mostrar y reconocer caracteres especiales tales como las tíldes, eñes, signos, etc.

* Generar el ChatBot

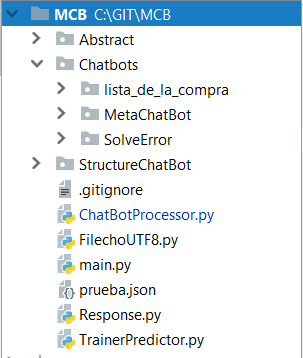
Para realizar este ejemplo, se han creado más intenciones.

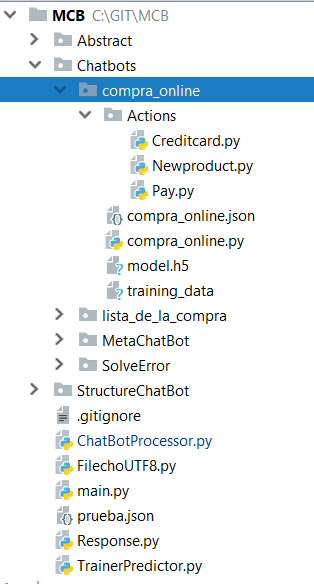
En las siguientes imágenes se puede ver que el Meta ChatBot genera el modelo del ChatBot “compra online” reconociendo las raíces de los patrones.





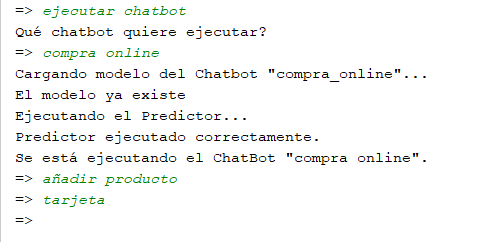
A su vez, va generando los ficheros necesarios tales como el fichero JSON, los ficheros Python de las acciones y el del programa principal.





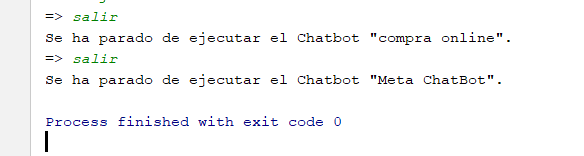
* Ejecutar “compra online” desde el Meta Chatbot

En este ejemplo después de ejecutar el ChatBot “compra online” se ejecutarán sus acciones. Ya que, en los ficheros de las acciones no está implementado el código que debe ejecutar cada uno de ellos, no mostrará una respuesta.



* Terminar las ejecuciones

En siguiente imagen se muestra la finalización de la ejecución tanto del ChatBot “compra online” como el del Meta ChatBot. Se muestra un mensaje de aviso antes de salir de la ejecución de cada ChatBot.



## 4.5 Solve Error

Este sub-sistema proporciona al usuario la capacidad de resolver fallos de las entradas no reconocidas.

Solve Error, al igual que el Meta ChatBot, es un ChatBot. Por lo que también utiliza la estructura definida para un ChatBot, el motor de ChatBots y el lenguaje Python.

También ofrece la interacción por texto o voz, haciendo más cómo el resolver los errores.

### 4.4.1 Análisis

Este ChatBot tiene la funcionalidad de editar la estructura JSON del ChatBot a mejorar sin que el usuario tenga que tocar directamente la estructura.

Tras seleccionar todos los fallos, indicar los valores correctos y editar el JSON, el Solve Error vuelve a entrenar y generar el modelo del ChatBot que se esté mejorando.

#### 4.5.1.1 Requisitos Funcionales

Los requisitos funcionales son aquellas funciones que realiza el sistema, ya sea entradas, acciones y salidas.

Entre sus requisitos funcionales se encuentran:

* RF01: Permitirá mostrar todos los ChatBots, incluso el Meta ChatBot y él mismo.
* RF02: Permitirá seleccionar la petición que no se ha reconocido y una intención a la que se le quiera asociar para resolverlo.
* RF03: Al seleccionar el ChatBot a resolver, cargará sus intenciones y sus errores automáticamente.
* RF04: Mostrará los errores e intenciones del ChatBot a resolver.
* RF05: Permitirá guardar las soluciones.
* RF06: Permitirá mostrar las soluciones guardadas.
* RF07: Permitirá mostrar el último error e intención seleccionado.
* RF08: Resolverá los errores que se han resuelto y guardado en la lista.
* RF09: Editará el JSON del ChatBot que se está resolviendo para introducir los cambios.
* RF10: Volverá a entrenar el ChatBot editado.
* RF11: Volverá a generará el modelo del ChatBot editado.
* RF12: Mostrará mensajes indicativos del estado en el que se encuentra el usuario.
* RF13: Podrá guardar las entradas no reconocidas, usando una palabra clave “Guardar sentencia”, en un fichero JSON.
* RF14: Podrá parar su ejecución.

#### 4.5.1.2 Requisitos No Funcionales

Los requisitos no funcionales son las características de funcionamiento, es decir, atributos de calidad del sistema.

Presenta los siguientes requisitos no funcionales:

* RNF01: El lenguaje, tanto escrito como hablado, debe ser en español.
* RNF02: Las respuestas deben ser claras en cada situación.
* RNF03: Los mensajes de error deberán ser claros.
* RNF04: El tiempo de respuesta no deberá ser muy larga.

### 4.5.2 Diseño

Este ChatBot, al iniciarse lista los nombres de los ChatBots creados hasta el momento con el fichero de errores, para que el usuario pueda escoger el que necesite mejorar.

Cuando el usuario selecciona un ChatBot, el Solve Error carga todas sus intenciones y la lista de errores a las estructuras correspondientes para facilitar el manejo de las mismas.

Por último , cuando el usuario decida que ya ha terminado, y haya al menos un fallo arreglado, el Solve Error procederá a actualizar el JSON del ChatBot que se está resolviendo, y volverá a entrenar y generar el modelo para que los cambios se carguen.

La edición del JSON se facilita gracias a la librería JSON de Python.

A continuación se mostrará unas imágenes del diseño de las clases implementadas y sus relaciones.

### 4.5.3 Pruebas

En este apartado, se mostrarán imágenes de la ejecución del Solve Error.

## 4.6 RESUMEN DEL CAPÍTULO

El desarrollo el sistema se divide en 3 sub-sistemas. Un motor de ChatBots que se encarga del procesamiento del lenguaje natural usando técnicas de Machine Learning, aprendizaje supervisado, para entrenar y generar el modelo. El Meta ChatBot que hace uso de este motor para reconocer las peticiones de los usuarios, pero también controla la estructura de los ChatBots para su creación. Y por último, el Solve Error que es un ChatBot de ayuda para mejorar aquellas peticiones que no hayan sido reconocidas.

Con estos tres sub-sistemas se obtiene un sistema capas de realizar tareas que otros entornos ofrecen, pero con la diferencia de que puede hacerlo con una interacción por voz para crear y mejorar los ChatBots.

# CAPÍTULO 5: EVALUACIÓN CUALITATIVA

Evaluación de la forma, medios para lograr los resultados

# CAPÍTULO 6: RECURSOS EMPLEADOS

En este capítulo se detallará los materiales usados, ya sea a nivel de software o hardware, explicando el por qué de su uso frente a otras opciones.

Se planteará una tabla de costes de implementación del sistema.

## 6.1 MATERIALES

Dentro de los materiales usados para el desarrollo de este sistema, se puede diferenciar entre materiales software y hardware.

**Materiales Software:**

Los recursos software usados son:

1. **Python**: se ha utilizado este lenguaje de programación, ya que es donde se encuentran la mayor parte de librerías para la programación con inteligencia artificial. Por otro lado, según numerosos post en *es.quora.com*, resulta más fácil el programar con este lenguaje, ya que es un lenguaje no tipificado, por la legibilidad de código, etc.
2. **PyCharm:** este entorno de programación se escogió porque presenta una interfaz amigable y porque ya se tenía conocimiento del mismo.
3. **Anaconda 3:** Instalador de componentes para Python.
4. **GitHub (web):** Se ha utilizado este repositorio de versiones porque ofrece herramientas para facilitar su uso. Además de aportar una gran comunidad para cualquier duda.
5. **GitHub Desktop:** Aplicación de escritorio para la actualización del repositorio.
6. **OS HighSierra:** Sistema operativo con el que se realizó el proyecto.
7. **Keras:** Esta es una librería de Python para utilizar las redes neuronales. Se optó por ella en vez de TensorFlow por la incompatibilidad de esta última con la versión de Python
8. **NLTK:** Esta herramienta sirve para reconocer el lenguaje natural, consiguiendo los conjuntos de entradas y salidas necesarias para el entrenamiento de las neuronas. Aplicando la técnica de aprendizaje supervisado.
9. **JSON:** Este formato de texto se ha utilizado para la estructura de los ChatBots, ya que es más ligera que XML.

**Materiales Hardware:**

Durante todo el proceso se ha contado con los siguientes componentes hardware:

1. **Ordenador portátil:** Para la programación del sistema. Preferiblemente con más de 8GB de RAM.
2. **Teléfono móvil:** Para la prueba de interacción por voz.
3. **Memoria RAM (8GB)**: Se aumentó la memoria del ordenador porque no tenía la suficiente para trabajar con muchas aplicaciones a la vez y soportar las características de la versión actualizada del sistema operativo.
4. **Disco de estado sólido (500GB)**: Se tuvo que cambiar el disco duro por uno de estado sólido porque estaba dañado.

## 6.2 TABLAS DE COSTES

A continuación se detallarán los costes económicos y temporales que han sido invertidos en el desarrollo del proyecto.

En la siguiente tabla se aprecia los gastos monetarios realizados para la obtención de los materiales usados para el sistema.

|  |  |
| --- | --- |
| Material | Precio en Euros |
| Memoria RAM de 8GB x 1 | 55,45 € |
| Disco de estado sólido 500GB x 1 | 92,99 € |
| Ordenador portátil x1 | 0 € |
| Teléfono móvil x1 | 0 € |
| Adquisición de todo el material software | 0 € |
| TOTAL | **148,44 €** |

Tabla 2: Tabla de gastos monetarios.

En la siguiente tabla se mostrará el coste temporal que ha requerido el proyecto. Mostrando las horas totales mínimas y máximas por cada mes.

|  |  |  |
| --- | --- | --- |
| Mes | Horas mínimas (total) | Horas máximas (total) |
| Octubre 2017 | 36 | 38 |
| Noviembre 2017 | 35 | 36 |
| Marzo 2018 | 32 | 33 |
| Abril 2018 | 30 | 32 |
| Mayo 2018 | 27 | 29 |
| Agosto 2018 | 38 | 40 |
| Septiembre 2018 | 41 | 43 |
| Octubre 2018 | 46 | 48 |
| TOTAL | **285** | **299** |

Tabla 3: Tabla de costes temporales por mes.

Con el objetivo de proporcionar el coste total del proyecto en la unidad monetaria, se convertirán las horas trabajadas a euros multiplicándolos por un valor de 38 € la hora.

La siguiente tabla mostrará el coste total del proyecto, sumando los gastos monetarios y los costes temporales

|  |  |  |  |
| --- | --- | --- | --- |
|  | | **Coste temporal mínimo** | **Coste temporal máximo** |
| **10.830,00 €** | **11.362,00 €** |
| **Gastos monetarios** | **148,44 €** | 10.978,44 € | 11.510,44 € |

Tabla 4: Costes finales del proyecto.

Las celdas amarillas de esta tabla muestran el resultado de sumar los costes temporales acumulados (mínimos y máximo) con los gastos monetarios.

# CAPÍTULO 7: CONCLUSIONES

Este proyecto se planteó para desarrollar un sistema capaz de crear ChatBots.

El sistema creado permite que el usuario configure interacción por texto o voz. La estructura diseñada en la que se basan todos los ChatBots es usado en este sistema. El Meta ChatBot se encarga de procesar esta estructura para generar todos los ficheros necesarios automáticamente y de testear, es decir, ejecutar un ChatBot que se esté creando o ya creado previamente.

Además, se ha implementado un ChatBot, Solve Error, con el que se puede mejorar cualquier ChatBot del sistema. Tras esta mejora, el Solve Error vuelve a entrenar y generar el modelo del ChatBot que se esté mejorando.

Para aquellos usuarios que tengan más conocimiento de programación, podrán programar directamente sus acciones para añadirlos al sistema para su uso.

Con esta implementación se ha podido satisfacer la finalidad del proyecto, cumpliendo con todos los objetivos planteados.

Este trabajo puede ayudar a crear distintos ChatBots se puede comunicarse con otros sistemas, aunque limitados, para aumentar su funcionalidad.

Los beneficios frente a otros entornos de desarrollo de ChatBots es que presenta una interacción por voz, lo que lo hace más cómodo a la hora de usarlo. También se puede programar las acción a ejecutar. Esta característica es lo que beneficia a los desarrolladores para poder usar los ChatBots creados para comunicarse con otros sistemas.

Aunque el uso no es para todos los usuarios, con futuras mejoras se puede conseguir este objetivo.

Como valoración final, aunque ya existan programas propios de empresas de renombre que permiten hacer lo que se planteó con este proyecto, no se ha podido encontrar uno que permita crear cualquier tipo de ChatBot utilizando la voz.

Sin duda, de lo aprendido con este proyecto, lo destacado es el desarrollo del sistema de forma modular para poder realizar cambios/mejoras de forma más fácil como la interacción por voz que se logró a mitad del proyecto.

Otro de los puntos a tener en cuenta, es que este proyecto está planteando cambiar el modo de uso de software como muchas otras aplicaciones que usan sus sistemas por voz, asistentes personales.

Primero fue la interacción por consola, luego se implementó interfaces de usuario y se usó el ratón, ahora se está intentando implantar la idea de que el software se puede usar con la voz.

[[[[Objetivos eran:

De los que se ha hecho:

Con esto hemos satisfecho los objetivos:

Implicaciones para otros que lo quieran utilizar – Beneficios

Limitaciones

Valoracion personal]]]]

# CAPÍTULO 8: TRABAJO FUTURO

En este capítulo se dará una visión de las mejoras que se pueden dar en este trabajo para mejorar sus funcionalidades. Pudiendo así, reforzar los puntos débiles que presenta para obtener un sistemas robusto.

Como se ha podido observar, este sistema aún no está completo. Presenta puntos que necesitan ser implementados y otros por mejorar. Algunas ideas para su mejora son:

1. Programar una aplicación móvil que use este sistema para aumentar la facilidad de uso.
2. Si se quiere enfocar a una aplicación web, el desarrollo de la parte FrontEnd de la misma ayudaría a que la aplicación sea más fácil de entender.
3. Además, la programación de un nuevo ChatBot de ayuda sería ideal para aquellas personas que no entendiesen mucho del tema, lo que haría llegar este sistema a muchas más personas sin que éstas tuviesen mucho conocimiento del tema.
4. También se podría incluir un ChatBot “escritorio”, que permitiese al usuario poder seleccionar el ChatBot que quiera ejecutar.

Siguiendo el símil del escritorio, el nuevo ChatBot sería el escritorio de un ordenador cualquiera y los demás ChatBots (generales o creados por el Meta ChatBot) sería las aplicaciones que el usuario podrá ejecutar seleccionándolas.

1. Tocando la parte funcional interna, se podría cambiar las acciones por otros ChatBots para conseguir encapsular aún más la funcionalidad. Por lo que se conseguiría simular la contextualización del lenguaje.
2. Mejorar las interacciones con el sistema.
3. Permitir al usuario seleccionar dinámicamente el modo de interacción para acercar el sistema a la usabilidad universal.
4. Permitir al usuario indicar la ruta donde desea guardar la estructura del ChatBots que está creando y, además darle la opción de introducir una ruta para que el Meta ChatBot cargue todos los ChatBots que se encuentren en esta ruta.
5. Dar la facilidad al usuario para que pueda introducir el código de las acciones de sus ChatBots.

Este proyecto se puede enfocar en muchas áreas, si se llega a desarrollar más puede llegar a ser una herramienta competitiva entre los demás entornos para crear ChatBots.

# GLOSARIO

A

acciones

Son las tareas a realizar por el ChatBot tras identificar la(s) intención(es) del usuario. *4*

algoritmo de agrupamiento

Es un procedimiento de agrupación de una serie de vectores de acuerdo con un criterio. Esos criterios son por lo general distancia o similitud. 11

C

ChatBots

Un chatbot es un programa informático con el que es posible mantener una conversación, tanto si queremos pedirle algún tipo de información o que lleve a cabo una acción. *4*

computación cognitiva

Sistemas de autoaprendizaje que utilizan la minería de datos, el reconocimiento de patrones y el procesamiento del lenguaje natural para imitar el funcionamiento del cerebro humano. *13*

H

heurísticas

Puede ser aplicada en diversas ciencias con el objetivo de crear medios, estrategias y principios como ayuda para alcanzar la solución más eficaz y eficiente al problema. *10*

hitos

Los hitos son una serie de etapas dentro de un mismo proyecto. Se determinan desde la planificación previa del mismo, se van revisando a medida que avanza nuestro trabajo y se pueden ir modificando según las necesidades del proyecto o cliente 3

I

Inteligencia Artificial

Es la inteligencia exhibida por máquinas. En ciencias de la computación, una máquina "inteligente" ideal es un agente racional flexible que percibe su entorno y lleva a cabo acciones que maximicen sus posibilidades de éxito en algún objetivo o tarea. *4*

intenciones

Es la idea que el usuario le comunica al ChatBot para que lleve a cabo la acción correspondiente. *4*

J

JSON

Es un formato de texto ligero para el intercambio de datos. *20*

K

KERAS

Es una biblioteca de red neuronal de código abierto escrita en Python. Es capaz de ejecutarse sobre TensorFlow , Microsoft Cognitive Toolkit o Theano. *23*

M

Machine Learning

Es una rama de la inteligencia artificial, cuyo objetivo es desarrollar técnicas que permitan que los ordenadores aprendan. 11

*Motor de ChatBots*

Es el programa capaz de procesar la arquitectura definida para un chatbot y ejecutar las diferentes tareas definidas él para que el chatbot funcione correctamente. *4*

Es el programa capaz de procesar la arquitectura definida para un chatbot y ejecutar las diferentes tareas definidas para que el chatbot funcione correctamente. *4*

P

Procesamiento Natural del Lenguaje

Es un campo de las ciencias de la computación, inteligencia artificial y lingüística que estudia las interacciones entre las computadoras y el lenguaje humano. *22*

Python

Es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible. *21*

R

redes neuronales

Una red neuronal artificial es un grupo interconectado de nodos similar a la vasta red de neuronas en un cerebro biológico. *4*

S

SaaS

Es un modelo de distribución de software donde el soporte lógico y los datos que maneja se alojan en servidores de una compañía de tecnologías de información y comunicación (TIC), a los que se accede vía Internet desde un cliente. *13*

sacos de palabras

El modelo de bolsa de palabras o saco de palabras es una representación simplificada utilizada en el procesamiento del lenguaje natural y la recuperación de información. *4*

T

text to speech

Un sistema text-to-speech (TTS) convierte el lenguaje de texto normal en habla *10*

thread

Es una secuencia de tareas encadenadas muy pequeña que puede ser ejecutada por un sistema operativo. *26*

X

XML-RPC

Es un protocolo de llamada a procedimiento remoto que usa XML para codificar los datos y HTTP como protocolo de transmisión de mensajes. *26*

# BIBLIOGRAFÍA

NLTK: Bird, Steven, Edward Loper and Ewan Klein (2009), *Natural Language Processing with Python*. O’Reilly Media Inc.

elEconomista. «¿Qué son exactamente los chatbots y para qué sirven?» *elEconomista*, 13 de Abril de 2016: 1.

gk\_. «Contextual Chatbots with Tensorflow.» *chatbotmagazine*, 2017: 1.

HAJ-SALEH, ALBERTO. «Qué son exactamente los 'bots' y cómo funcionan.» *GQ*, 2017.

Wikipedia. «Artificial Linguistic Internet Computer Entity.» 2018.

Wikipedia. «Bot Conversacional.» 1. 2018.

Wikipedia. «Test de Turing.» 1. 2018.