



UNIVERSIDAD POLITÉCNICA DE MADRID

Máster Universitario en Automática y Robótica

## **Inteligencia Artificial**

Creación e implementación de un chatbot para la  
página de la ETSII

Autores:

Maider Arano Saldias  
Gorka Gomez Romero  
Hugo Torralba Galindo

Enero 2023

## ÍNDICE

1. INTRODUCCIÓN Y OBJETIVOS	2
2. ESTADO DEL ARTE	3
2.1. Aplicaciones de mensajería	3
2.1.1. Whatsapp Business	3
2.1.2. Telegram	5
2.1.3. Facebook messenger	6
2.2. Plataformas de procesamiento del lenguaje natural (NLP)	7
2.2.1. Watson (IBM)	7
2.2.2. DialogFlow (Google)	8
2.2.3. Microsoft Bot Framework	8
2.3. Conclusiones	9
3. DESARROLLO	10
3.1. Herramientas utilizadas	10
3.1.1. Intérprete de Lenguaje Natural: Dialogflow	10
Intents	11
Entities	15
Fulfillment	16
Integrations	17
3.1.2. MongoDB	17
3.1.3. Telegram	19
3.2. Implementación del chatbot	19
3.2.1. Comunicación de DialogFlow con MongoDB	19
3.2.2. Vinculación de Telegram con Dialogflow	21
4. ANEXOS	23
4.1. Anexo I. Manual	23
4.2. Anexo II. Demostrador	23
4.3. Anexo III. Código	23
5. REPARTO DE ROLES	24
6. CONCLUSIONES	26
7. BIBLIOGRAFÍA	27

## 1. INTRODUCCIÓN Y OBJETIVOS

En los últimos años, el tema de la inteligencia artificial ha cobrado cada vez más importancia en el debate público. Y no es para menos, ya que se trata de una tecnología que está cambiando rápidamente nuestras vidas y que promete revolucionar aún más el mundo en el futuro. Desde asistentes virtuales como Siri o Alexa, hasta coches autónomos, pasando por robots médicos o sistemas de trading automatizados, la inteligencia artificial está cada vez más presente en nuestras vidas. Mediante este trabajo se pretende profundizar un poco más en el uso y desarrollo de los asistentes virtuales o comúnmente denominados como chatbots.

Hablamos de programas de computadora diseñados para simular una conversación humana, ya sea a través de texto, voz o ambos con el fin de mejorar la experiencia del cliente y facilitar el acceso a la información necesaria. Algunos chatbots pueden realizar tareas simples como responder a preguntas básicas, mientras que otros pueden ser mucho más avanzados y capaces de mantener una conversación casi indistinguible de la de un humano. En general, estas herramientas se basan en el reconocimiento de patrones y en el uso de reglas predefinidas, para responder a las entradas del usuario.

Remontándonos a la década de 1960, estos asistentes se hicieron populares cuando el científico Alan Turing publicó su artículo "Computación e inteligencia". En ese artículo, Turing describió un experimento llamado "El test de Turing", que es una prueba para determinar si una máquina es capaz de pensar como un humano. Si dicha máquina es capaz de responder de manera indistinguible de un ser humano, se dice que el test de Turing ha sido aprobado. A partir de este experimento, se popularizó la idea de crear máquinas inteligentes que pudieran comunicarse con los humanos siendo Joseph Weizenbaum la primera persona en intentar crear un chatbot (ELIZA, 1966).

Desde su aparición, estas máquinas han ido evolucionando para convertirse en lo que hoy se conoce como chatbots inteligentes. Están equipados con Inteligencia Artificial (IA) y Procesamiento de Lenguaje Natural (NLP), lo que les permite entender el lenguaje de un usuario y ofrecer respuestas adecuadas. Según la estimación de Juniper Research, se esperaba que el número de chatbots en funcionamiento se duplicará para este año, pasando de 8 000 a 16 000. Esta tendencia se explica en parte, por el creciente interés de las empresas debido a que ya no son solo una herramienta de soporte técnico o atención al cliente, sino que les permite interactuar con sus clientes y ofrecerles una experiencia personalizada.

El enfoque principal de este trabajo es realizar un estudio de alternativas para el desarrollo de un asistente virtual y la implementación de este en la página de la ETSII de la Universidad Politécnica de Madrid. El desarrollo realizado y explicado detalladamente en los siguientes apartados tiene como fin facilitar la tarea de búsqueda de información personalizada para los alumnos del centro. De esta manera, los estudiantes podrán acceder rápidamente a las consultas de horarios para másteres, fechas administrativas y de exámenes, novedades del centro, contactos del profesorado o hasta solicitar citas previas en secretaría.

## 2. ESTADO DEL ARTE

Los creadores de chatbots tienen como objetivo cambiar la forma en que interactuamos con la tecnología y nos comunicamos en línea, pero pocos han logrado que cambiemos nuestros hábitos, como continuar escribiendo en el teclado mientras buscamos en línea. En este apartado se realiza un trabajo del estado del arte para el desarrollo de chatbots ahondando en la posibilidad de plataformas en el mercado actual tanto de mensajería como de NLP para la generación un chatbot para la ETSII.

*La manera más natural de empezar una interacción para un ser humano es comenzando a hablar, no utilizando botones, menús, ni formularios... La aparición de las interfaces conversacionales (chatbots) supone el fin de las webs y apps tal y como las entendemos actualmente.*

Eduardo Manchon, fundador de Panoramio.

### 2.1. Aplicaciones de mensajería

El objetivo principal de las apps de mensajería es llegar al mayor número de usuarios de la manera más fácil y accesible posible. Entre muchos de los beneficios de la implementación de un chatbot en un canal de mensajería instantánea se encuentran:

- Gestión de conversaciones de forma simultánea en diferentes idiomas y fuera del horario comercial proporcionando el 100% de la disponibilidad.
- Centralizar la gestión de consultas recibidas.
- Automatizar respuestas ágiles y efectivas a preguntas concretas.
- Optimizar la experiencia del usuario potenciando la Atención al Cliente.
- Simplificar la gestión y aumentar la productividad de los agentes que pueden dedicarse a tareas más complejas.

Aunque a día de hoy existe un amplio abanico de posibilidades, el análisis realizado se ha centrado sobre todo en tres de los gigantes de la mensajería y el centro de los chatbots como son WhatsApp Business, Telegram y Facebook Messenger. Aunque cualquiera de las opciones presentadas proporcionan una variada oferta de ventajas, es conveniente estudiar la opción más conveniente para nuestro caso.

#### 2.1.1. Whatsapp Business

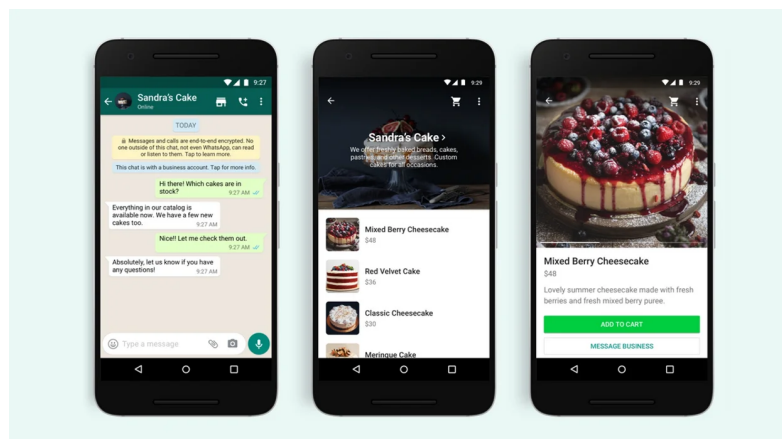
WhatsApp fue creado en 2009 por Jan Koum y Brian Acton, dos exempleados de Yahoo!. La aplicación se lanzó oficialmente en enero de 2010 y se convirtió rápidamente en una de las aplicaciones de mensajería más populares del mundo.

En 2014, WhatsApp fue adquirida por Facebook por una cantidad récord de 19 mil millones de dólares. Desde entonces, la aplicación ha seguido

creciendo y actualmente cuenta con más de 2 mil millones de usuarios activos al mes en todo el mundo. Es una de las aplicaciones de mensajería más utilizadas en el mundo ya que está disponible para Android, iOS, Windows Phone y otros sistemas operativos móviles, y también está disponible como una versión para escritorio que se puede usar en ordenadores con Windows o Mac. Algunas de las características más importantes de WhatsApp son:

- Mensajería instantánea entre usuarios de la aplicación
- Llamadas de voz y video a través de internet
- Generación de grupos para comunicación entre múltiples usuarios
- Estados que permiten compartir fotos, videos y mensajes de texto temporalmente
- Mensajes de voz para no escribir mensajes de texto
- Envío de archivos
- Cifrado de extremo a extremo para proteger la privacidad de los usuarios y garantizar la seguridad de los usuarios

A principios de 2018, WhatsApp presentó “WhatsApp Business” como una aplicación paralela a la original diseñada para atender las necesidades de las empresas. Ofrece todas las funciones de WhatsApp, pero también agrega algunas características útiles empresariales, como la posibilidad de crear perfiles corporativos, responder automáticamente a los mensajes, estadísticas detalladas sobre el rendimiento o la categorización de mensajes. Sin embargo, lo realmente interesante de esta aplicación es la posibilidad de crear una herramienta que permita automatizar, ordenar y responder los mensajes creando un chatbot. Aún así, y siguiendo el hilo del proyecto aquí presentado, uno de los inconvenientes para poder acceder a la API de WhatsApp Business es que está habilitada únicamente para medianas y grandes empresas. Es decir, si eres un simple aficionado a la programación o no formas parte de una empresa reconocida, no obtendrás permisos de desarrollo ni podrás acceder a la API de WhatsApp.



*Figura 1. WhatsApp Business para desarrollo de chatbots en empresas*

### 2.1.2. Telegram

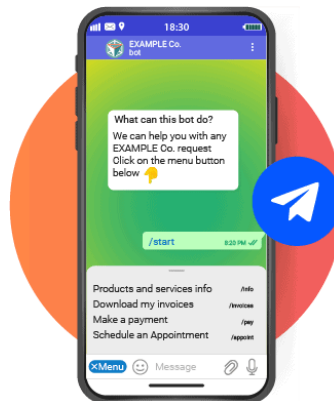
Telegram es una aplicación de mensajería gratuita y de código abierto para smartphones y ordenadores. Fue creada en 2013 por los hermanos rusos Pavel y Nikolai Durov.

Al igual que WhatsApp, permite a los usuarios enviar mensajes de texto y hacer llamadas de voz y video a otros usuarios de la aplicación a través de internet sin necesidad de pagar por las tarifas de mensajes o llamadas telefónicas. Telegram también ofrece la posibilidad de enviar mensajes de voz, imágenes, videos y documentos y tiene un límite de tamaño de archivo mayor que otras aplicaciones de mensajería.

Telegram destaca por su enfoque en la seguridad y la privacidad de los usuarios. Además del cifrado de extremo a extremo, ofrece una serie de opciones de privacidad avanzadas como la posibilidad de configurar mensajes para autodestruirse después de un periodo de tiempo específico.

Por otro lado, se trata de una plataforma que, a diferencia de otros, permite conectarse a miles de personas a nivel mundial en un mismo canal. Ofrece una gran variedad de aplicaciones más allá de chatear o enviar y recibir mensajes de voz, gracias a su plataforma de bots prediseñados.

Finalmente, la creación de chatbots propios es una funcionalidad incorporada en la misma aplicación que permite al usuario realizar sus propias creaciones con libertad de decisión absoluta sin necesidad de ningún permiso por parte de la compañía.



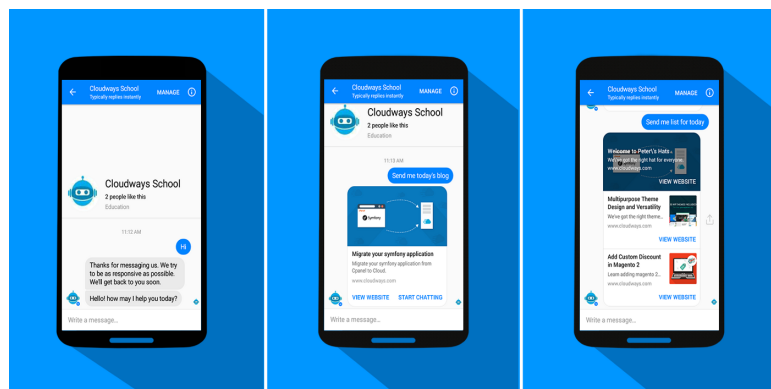
*Figura 2. Implementación y desarrollo de chatbots en Telegram*

### 2.1.3. Facebook messenger

Tal y como se ha explicado para las anteriores aplicaciones, Facebook Messenger también es una app de mensajería gratuita para smartphones y ordenadores que se basa en la red social Facebook.

Facebook Messenger se lanzó por primera vez en 2011 como una aplicación independiente, pero en 2014 se integró en la red social Facebook convirtiéndose en la aplicación de mensajería oficial de la plataforma. Actualmente, cuenta con más de 1.300 millones de usuarios activos al mes en todo el mundo y se utiliza ampliamente para comunicaciones con amigos y familiares a través de la red social.

En cuanto a características destacables, esta plataforma ofrece juegos, extensión de la funcionalidad de Facebook y desarrollo de chatbots. Sin embargo, en comparación con las otras plataformas, no ofrece la opción de búsqueda de mensajes y la capacidad de envío de archivos es mucho menor.



*Figura 3. Extensión para desarrollo de chatbots en Facebook Messenger*

## 2.2. Plataformas de procesamiento del lenguaje natural (NLP)

Las plataformas de procesamiento del lenguaje natural (NLP) son conjuntos de herramientas y tecnologías que se utilizan para analizar y procesar el lenguaje del humano. Su objetivo es permitir que las máquinas comprendan y procesen el lenguaje humano de la misma manera que lo hacen entre los humanos, lo que les permite interactuar con los usuarios de manera más natural y efectiva.

Este tipo de plataformas se utilizan en una amplia variedad de aplicaciones como el análisis de sentimientos en las redes sociales, la traducción automática y lo más importante para nosotros, la creación de chatbots para la atención al cliente. Algunas de ellas también se emplean para mejorar la accesibilidad, permitiendo a las personas que tienen dificultades para hablar o escribir, utilizar el lenguaje natural para interactuar con máquinas o sistemas.

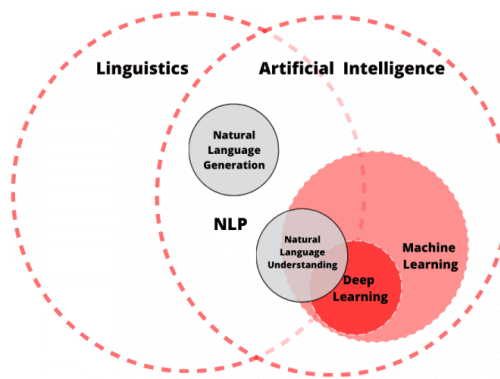


Figura 4. Procesamiento del Lenguaje Natural e Inteligencia Artificial

Las plataformas de NLP se basan en tecnologías de aprendizaje automático y utilizan técnicas como el análisis morfológico, la segmentación y la clasificación de palabras para analizar y procesar el lenguaje humano. Adicionalmente, suelen ir cogidos de la mano de bases de datos del lenguaje y diccionarios de sinónimos para mejorar la comprensión.

En este apartado se pretende estudiar la oferta de plataformas de mayor demanda en el mercado compatibles con el desarrollo de chatbots para escoger el que más se adecua a nuestro caso.

### 2.2.1. Watson (IBM)

Watson es una plataforma de Inteligencia Artificial desarrollada por IBM que se utiliza para procesar y analizar grandes cantidades de datos.

La construcción y desarrollo de chatbots o *workshops* en Watson se realiza mediante *conversation services* y se puede emplear Node.js, Python, .Net, Android o hasta iOS. El NLP se basa en el uso de *intents* y *entities* junto con sinónimos. Estos últimos son los conjuntos de palabras conocidas dentro de una misma *entity*.

Una de las mayores cualidades a destacar de IBM Watson es la función denominada *Fuzzy Matching* que es capaz de reconocer palabras o expresiones mal formuladas por el usuario sin quedarse bloqueado. Además, proporciona una herramienta de monitorización analítica utilizada sobre los resultados obtenidos de las conversaciones.

A modo de punto positivo, IBM cuenta también con una extensa documentación para la correcta creación de chatbots con diversidad de ejemplos.

En cuanto a la disponibilidad de la plataforma, IBM ofrece una versión gratuita de Watson conocida como IBM Watson Lite que incluye acceso limitado a Watson Assistant, Watson Discovery y Watson Studio. Uno de los principales problemas es el uso restringido en el tamaño de datos y modelos a procesar.



### 2.2.2. DialogFlow (Google)

DialogFlow es una NLP adquirida por Google en septiembre de 2016. Fue creada como herramienta para el desarrollo de chatbots y aplicaciones de voz y se ha convertido en una de las plataformas más populares y ampliamente utilizadas en la actualidad.

Destaca entre sus competidores debido al amplio abanico de interfaces de conversación que llega a abarcar. Esta plataforma permite el despliegue de sus chatbots en muchas aplicaciones de mensajería, páginas web, asistentes virtuales y aplicaciones particulares.

La incorporación de herramientas de análisis y monitorización, la posibilidad de integrar otros sistemas como bases de datos, APIs o servicios web gracias al uso de webhooks y la permisividad de configuración en algunos aspectos del algoritmo de Machine Learning hacen de DialogFlow una de las mejores opciones del mercado actual a la hora de desarrollo de chatbots.

Al igual que IBM Watson, también ofrece una excelente documentación aportada por Google para poder ofrecer soporte y ayuda a los usuarios para el manejo de la plataforma.

En cuanto al precio, DialogFlow contiene una versión gratuita que incluye un número limitado de solicitudes por mes o uso de la memoria de GoogleCloud y no incluye algunas características avanzadas como el soporte para reconocimiento de voz.

### 2.2.3. Microsoft Bot Framework

La plataforma de desarrollo de Microsoft divide la implementación de un chatbot en tres partes: Portal de Desarrollo mediante Node.js y .Net, *Bot Connector* para la conexión entre chatbots y desplegar estos en aplicaciones de mensajería y *Bot Directory* a modo de repositorio.

Esta plataforma permite monitorizar, hacer analíticas e integrar servicios como bases de datos, APIs o servicios web. Para el procesamiento del lenguaje natural, se utiliza un servicio basado en aprendizaje automático desarrollado por Microsoft, llamado LUIS (*Language Understanding Intelligent Service*), basado en *intents*, *entities* y *features*.

Comparando con otras plataformas, su framework se integra con diferentes servicios y aplicaciones de Microsoft (Azure, Office 365 y Dynamics), lo que permite utilizar las mismas herramientas en los chatbots. Además, existe una amplia comunidad de desarrolladores y usuarios que utilizan el Microsoft Bot Framework y comparten conocimientos y recursos en línea, lo que puede ser útil para obtener ayuda y aprender más sobre cómo utilizar el framework.

## 2.3. Conclusiones

Tras haber analizado tanto las opciones para la plataforma de mensajería como de NLP, se ha decidido trabajar con Telegram y DialogFlow.

Por un lado, a pesar de que WhatsApp es la aplicación de mensajería instantánea por excelencia, no es una plataforma que se adecúe al proyecto ya que no se obtendrían los permisos necesarios para su uso. En cuanto a la elección entre Facebook Messenger y Telegram, la decisión está bastante clara. Telegram ofrece una libertad a la hora de programación envidiable donde el usuario toma todas las decisiones y muchos extras que pueden ser interesantes a la hora de desarrollar el chatbot en cuestión.

Por otro lado, la elección de la plataforma NLP es DialogFlow. Se trata de la opción más completa en cuanto a frameworks de desarrollo, técnicas de procesamiento del lenguaje natural, integración y comunicación con otros servicios y monitorización.

### 3. DESARROLLO

Después de examinar las diferentes opciones disponibles para el desarrollo de los diferentes pilares que componen un chatbot que cumpla con los objetivos establecidos, se procederá a profundizar en el uso de cada una de las herramientas elegidas y se proporcionará una descripción detallada de lo necesario para su implementación y puesta en marcha.

#### 3.1. Herramientas utilizadas

Los cuatro elementos principales que componen el chatbot son el usuario, el sistema de interacción con el usuario, el intérprete de lenguaje natural y la base de datos.



Figura 5. Esquema básico de elementos que componen el Chatbot.

Un intérprete de lenguaje natural es un programa diseñado para facilitar la comunicación y el intercambio de datos de usuario-computadora. Tiene como objetivo analizar el lenguaje natural utilizado por los humanos y procesarlo para que las máquinas sean capaces de entenderlo. De esta forma, los usuarios pueden ejecutar con mayor facilidad consultas sin conocer un lenguaje de programación específico. Los intérpretes de lenguaje natural son utilizados principalmente en aplicaciones como asistentes virtuales, chatbots o sistemas de reconocimiento de voz.

Con el objetivo de dotar con más funciones y capacidad a nuestro chatbot, se ha optado por incluir una base de datos la cual podrá almacenar información adicional y responder con mayor y mejor capacidad los requerimientos del usuario. Un sistema de base de datos es un programa que permite almacenar y acceder de manera eficiente a grandes cantidades de datos estructurados y relacionados entre sí, permitiendo realizar consultas y modificaciones según las características específicas que el usuario desee.

El chatbot puede ser integrado en diferentes plataformas de mensajería instantánea, tal como se ha mencionado anteriormente. Desde la aplicación se podrá interactuar con el chatbot.

##### 3.1.1. Intérprete de Lenguaje Natural: Dialogflow

Dialogflow es una herramienta de creación de chatbots que entiende el lenguaje natural y proporciona una infraestructura para recrear conversaciones y estructurar conversaciones para interactuar con los

usuarios de manera fluida y correcta. Actualmente esta plataforma puede admitir más de 14 idiomas y es cada vez más capaz de manejar abreviaturas y errores ortográficos mediante la función fuzzy matching.

Esta herramienta utiliza inteligencia artificial al utilizar técnicas de procesamiento de lenguaje natural. La inteligencia artificial es un amplio campo que cubre todo el desarrollo de sistemas informáticos que deben de ser capaces de realizar tareas que normalmente requerirían inteligencia humana, como por ejemplo, la toma de decisiones o la resolución de tareas o problemas complejos. En el contexto de Dialogflow, la IA se utiliza para permitir que el chatbot o asistente de voz comprenda y responda a las entradas del usuario de forma natural y lo más cercana posible a la humana.

Dialogflow utiliza una combinación de aprendizaje automático y sistemas basados en reglas para procesar y comprender las entradas de los usuarios. Puede utilizar el aprendizaje automático para aprender a partir de datos de ejemplo y mejorar su capacidad para comprender y responder a las entradas de los usuarios con el tiempo. Dialogflow también permite a los desarrolladores especificar reglas sobre cómo un chatbot o asistente de voz debe responder a la entrada específica del usuario, proporcionando un nivel de flexibilidad y control sobre la conversación.

En resumen, Dialogflow es una plataforma que permite el desarrollo de interfaces conversacionales en lenguaje natural utilizando inteligencia artificial, lo que permite a los desarrolladores crear chatbots y asistentes de voz capaces de comprender y responder a las entradas del usuario de forma similar a la humana.

A continuación, se desarrollará el funcionamiento de Dialogflow exponiendo sus principales componentes.

### Intents

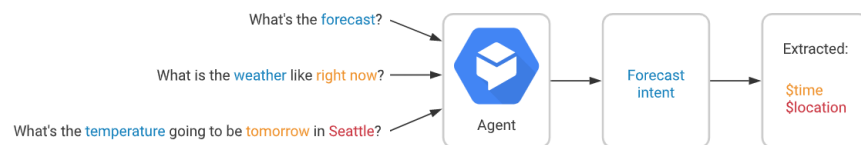
Un intent representa una correspondencia entre lo que dice un usuario y la acción que debe realizar el software. Cuando un usuario proporciona información, Dialogflow la relaciona con la mejor intent de su agente. Una vez el programa conecta la petición del usuario con un intent, Dialogflow puede realizar una acción en su nombre, como dar respuesta a una solicitud del usuario o proporcionar una contestación.

Los intents pueden utilizarse para realizar muchas acciones como satisfacer solicitud de usuarios, cómo realizar consultas en alguna base de datos o realizar una tarea. También pueden utilizarse para proporcionar información a los usuarios, como responder a una pregunta o proporcionar una definición.

Los intents consisten en una lista de frases de entrenamiento, que son frases de ejemplo que un usuario podría decir, y la correspondiente

información de acción y respuesta. Dialogflow utiliza estas frases de entrenamiento para hacer coincidir la entrada del usuario con la intent correcta.

En la figura 6 se muestra un esquema básico de funcionamiento de un intent. En este caso, se ha creado un agente meteorológico que será capaz de reconocer preguntas sobre pronóstico del tiempo y responderlas. Dialogflow extraerá de la frase información útil para responder a la solicitud del usuario, como en éste caso el tiempo y la hora, y la vincula al intent “Forecast” creado previamente, gracias a las frases de entrenamiento previamente definidas.



*Figura 6. Esquema funcionamiento de un intent.*

La configuración básica de un intent pasa por cuatro aspectos:

- **Frases de entrenamiento (Training phrases):** Las frases de entrenamiento son ejemplos de entradas de usuario que proporciona para ayudar al sistema a entender la intención del usuario. Se utilizan para entrenar el modelo de procesamiento de lenguaje natural (NLP), para reconocer y clasificar la entrada del usuario de acuerdo con la intent específica que ha definido para su chatbot.

Es importante proporcionar suficientes frases de entrenamiento para que el modelo de NLP comprenda bien las entradas de usuario que puede encontrar, pero no tantas, ya que el modelo puede verse abrumado o ser demasiado específico. También es útil incluir variaciones y sinónimos en las frases de entrenamiento para ayudar al modelo a comprender las distintas formas en que los usuarios pueden expresar la misma intención.

La figura 7 muestra un ejemplo de las frases de entrenamiento para consultar el horario académico. Se muestran las diferentes frases de entrenamiento programadas para este intent.

Training phrases ?
Search training | 🔍 ^

**⚠** Template phrases are deprecated and will be ignored in training time. More details [here](#).

When a user says something similar to a training phrase, Dialogflow matches it to the intent. You don't have to create an exhaustive list. Dialogflow will fill out the list with similar expressions. To extract parameter values, use [annotations](#) with available [system](#) or [custom](#) entity types.

” Add user expression

” cuales son mis clases de hoy

” Que clases tiene el master de robótica el lunes

” que clases tengo el martes

” quiero saber el horario del lunes

Figura 7. Ejemplo de frases de entrenamiento.

- **Acción y parámetros (Actions and parameters):** Una intent puede tener acciones y parámetros asociados que se utilizan para especificar las acciones concretas que deben llevarse a cabo y los datos necesarios para completar dichas acciones. Las acciones son definidas por el programador y representan las tareas específicas que el chatbot debe realizar cuando se activa la intent. En cambio, los parámetros se utilizan para indicar que información específica hay que extraer de la entrada del usuario, necesaria para completar la acción que viene asociada en el intent. Siguiendo el ejemplo, si se desea saber el horario de un día de la semana para un curso específico, serán esos los dos parámetros indispensables para realizar la consulta.

Action and parameters
^

Enter action name

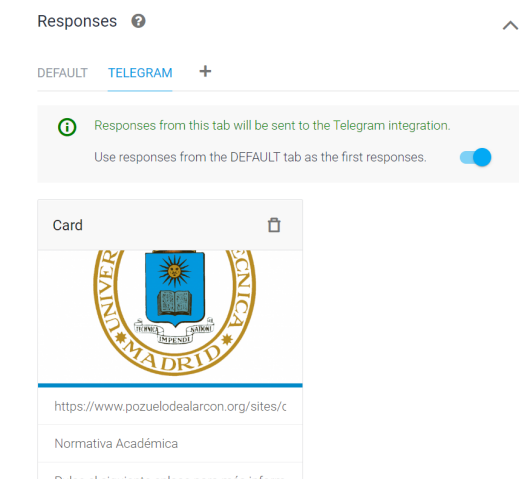
REQUIRED ?	PARAMETER NAME ?	ENTITY ?	VALUE	IS LIST ?	PROMPTS ?
<input checked="" type="checkbox"/>	cuatrín	<span style="background-color: #bbdefb;">@cuatrimestre</span>	\$cuatrimestre	<input type="checkbox"/>	¿De qué cuatrimestre...
<input checked="" type="checkbox"/>	DíasDe	<span style="background-color: #bbdefb;">@DíasDeLaSemana</span>	\$DíasDeLaSemana	<input type="checkbox"/>	¿De qué día de la semana...
<input checked="" type="checkbox"/>	Master	<span style="background-color: #bbdefb;">@Master</span>	\$Master	<input type="checkbox"/>	¿Sobre qué Master...
<input type="checkbox"/>	Enter r	Enter e	Enter value	<input type="checkbox"/>	—

[+ New parameter](#)

Figura 8. Ejemplo de acción y parámetros.

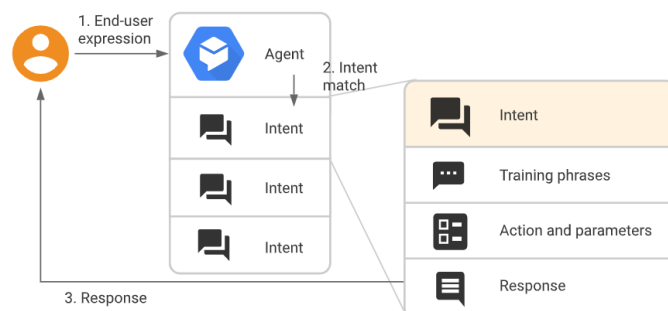
- **Fulfillment:** Esta herramienta ha de ser configurada si se desea recibir información proveniente de un webhook para generar una respuesta al intent. Para ello, se debe activar “Enable webhook call for this intent” y así se podrá desplegar el intent a través de un webhook para proporcionar datos a un usuario. Este apartado se desarrollará en profundidad más adelante.
- **Respuestas (Responses):** Sirve para programar respuestas automáticas. En el caso de que el texto introducido coincide con alguna de las frases de entrenamiento de una intet, la herramienta responderá de manera predefinida con algunas de las opciones programadas. Es decir, es un banco de respuestas por defecto.

La figura 9 muestra la respuesta en modo “card” a una consulta sobre normativa.



*Figura 9. Ejemplo de respuesta predefinida tipo “card”*

La figura 10, muestra un esquema del flujo básico del proceso de configuración de un intent.



*Figura 10. Esquema de flujo básico para explicación de Intent.*

## Entities

Las entities permiten especificar los datos concretos que el chatbot debe buscar y extraer de la información introducida por el usuario una vez reconocida la intent.

Las entities pueden estar predefinidas, lo que significa que Dialogflow las proporciona y están disponibles para que las utilice en sus intents, como por ejemplo fecha, hora etc. Y además, Dialogflow también permite definir entities personalizadas para tipos de datos específicos de su aplicación.

Mediante el uso de entities, es posible especificar las piezas concretas de información que el chatbot debe buscar y extraer de la entrada del usuario, lo que permite construir interfaces conversacionales más interactivas. Además, es posible añadir sinónimos, así, cada vez que un usuario incluya palabras presentes en el repertorio de sinónimos, Dialogflow deducirá que se refiere a la palabra definida, y la relacionará con ella.

Las entities suelen combinarse con variables para crear respuestas más dinámicas. Por ejemplo, se puede utilizar una variable para insertar el valor de entity extraído en una respuesta, como "El próximo [día de la semana] las asignaturas programadas son [asignaturas]". Esto permite crear respuestas personalizadas que incluyen información específica que el usuario ha proporcionado.

La figura 11 que se muestra a continuación, se diseña una entity para los días de la semana. Como se puede apreciar, se definen una serie de sinónimos o palabras mal escritas, para que siempre que se reciba esa información, se vincule al día de la semana que corresponda.

DiasDeLaSemana SAVE ⋮

☒ Define synonyms ⓘ ☐ Regexp entity ⓘ ☐ Allow automated expansion  
☐ Fuzzy matching ⓘ

l	lunes, luns, Lunes, LUNES
m	martes, Martes, MARTES
x	miércoles, Miércoles, Miercoles, miercoles, MIERCOLES, Miércoles, los miércoles
j	jueves, JUEVES, Jueves
v	viernes, Viernes, VIERNES
t	t, todos, toda la semana, todos los dias, la semana entera, de toda la semana

[Click here to edit entry](#)

Figura 11. Ejemplo de entidad con días de semana.



## Fulfillment

En Dialogflow, el apartado de fulfillment permite gestionar el procesamiento de las solicitudes de los usuarios que se realizan a un agente. Permite integrarse con servicios externos, como bases de datos externas, para proporcionar respuestas a las solicitudes de los usuarios y desencadenar acciones, como búsquedas en bases de datos, basadas en la entrada del usuario. Por ejemplo, si un usuario final desea saber el horario de un día específico para un máster específico, mediante la configuración del apartado Fulfillment es posible verificar una base de datos con la información aportada por el usuario y responder con información acerca de las asignaturas y horarios programados para el día solicitado.

Se puede habilitar el cumplimiento para una intent en la configuración de la intent y habilitando la opción "Usar webhook". A continuación, se puede especificar la URL del webhook y configurar cualquier parámetro o encabezado adicional que deba enviarse con la solicitud.

Cuando esto ocurre, Dialogflow envía un mensaje de solicitud de webhook a tu servicio de webhook escrito en URL. Este mensaje contiene información sobre la intent coincidente, la acción, los parámetros y las respuestas definidas para esa intención. A continuación, en el caso de éste Chatbot, se realiza una consulta a base de datos con entrada de los parámetros especificados, y el servicio envía un mensaje de respuesta de webhook a Dialogflow. Este mensaje contiene la respuesta que se debe enviar al usuario final en Dialogflow.

### ⚡ Fulfillment

#### Webhook

ENABLED 

Your web service will receive a POST request from Dialogflow in the form of the response to a user query matched by intents with webhook enabled. Be sure that your web service meets all the [webhook requirements](#) specific to the API version enabled in this agent.

URL\*

BASIC AUTH

HEADERS

 [Add header](#)

*Figura 12. Configuración del webhook.*

## Integrations

Las integraciones son una forma de conectar tu chatbot a servicios y aplicaciones externas. Esto le permite acceder a funciones y datos adicionales, y realizar tareas como enviar notificaciones, actualizar bases de datos o simplemente tener más accesible la información. Es decir, es una forma de integrar Dialogflow a muchas plataformas de conversación populares como Google Assistant, Slack, Telegram o Facebook Messenger. Existen varios tipos de integraciones de Dialogflow como las integraciones de telefonía, o las basadas en texto.

La configuración de la integración de Dialogflow a Telegram se tratará más a fondo en capítulos posteriores al desarrollar la implementación del chatbot.

### Integrations

Text based

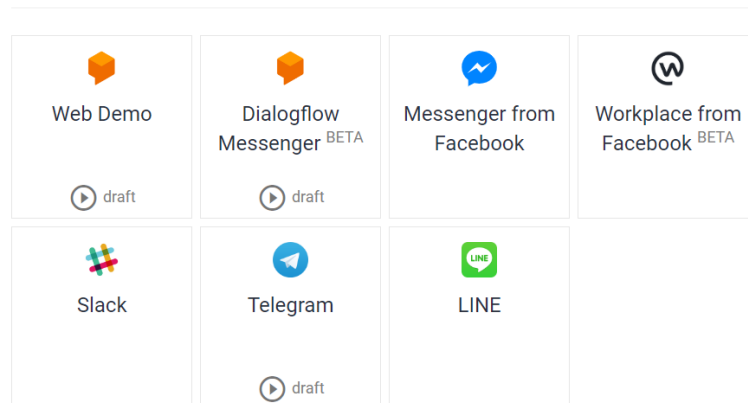


Figura 13. Integraciones basadas en texto.

### 3.1.2. MongoDB

El segundo pilar del Chatbot programado para la ETSII es la popular base de datos en la nube MongoDB. Ésta es una popular base de datos NoSQL que puede utilizarse con Dialogflow de diversas maneras. Entre otras cosas, existe la posibilidad de crear un entorno en la nube de 5 GB de almacenamiento gratis.

Como se ha tratado en el apartado anterior, al tratarse de un servicio externo, como es una base de datos, existe la posibilidad de integración de la misma a Dialogflow con la herramienta fulfillment. De esta forma, es posible utilizar MongoDB junto con fulfillment para proporcionar respuestas a las solicitudes de los usuarios consultando la base de datos con parámetros de entrada enviados por el webhook, dando respuesta a las entradas de los usuarios. Por ejemplo, se puede utilizar MongoDB para

buscar información en una base de datos o para actualizar un registro en respuesta a una solicitud del usuario.

Para ello, habrá que habilitar el fulfillment para una intent en Dialogflow y crear un webhook. Posteriormente, se programará un código en NodeJS para conectarse a la base de datos y realizar acciones como leer y escribir datos, que luego será enviada a la plataforma Dialogflow para ser mostrada al usuario. Utiliza los valores de entities extraídos de la entrada del usuario para consultar la base de datos y recuperar información relevante. Por ejemplo, puedes utilizar la entrada de fecha y ubicación para consultar el parte meteorológico en una base de datos.

Se podrán crear diferentes bases de datos, con estructuras y tipos de datos diferentes, tal y como se muestra en las figuras 14 y 15.

ChatBotUPM

LOGICAL DATA SIZE:

STORAGE SIZE:

INDEX SIZE:

TOTAL COLLECTIONS:

CREATE

19.68KB

144KB

144KB

4

Collection Name	Documents	Logical Data Size	Avg Document Size	Storage Size	Indexes	Index Size	Avg Index Size
<a href="#">contactodepartamentos</a>	34	5.93KB	179B	36KB	1	36KB	36KB
<a href="#">exámenes</a>	6	1.21KB	207B	36KB	1	36KB	36KB
<a href="#">horarios</a>	64	12.1KB	194B	36KB	1	36KB	36KB
<a href="#">ubicacion</a>	2	450B	225B	36KB	1	36KB	36KB

Figura 14.

```
_id: ObjectId('63977fed37025b7e0f5a5c0c')
master: "MUAR"
asignatura: "Robótica Aplicada"
fecha: "12 de enero de 2023"
hora: "18:30"
aula: "36"
convocatoria: 1
createdAt: 2022-12-12T19:24:29.359+00:00
updatedAt: 2022-12-12T19:24:29.359+00:00
__v: 0
```

Figura 15. Ejemplo de un archivo de una colección de la base de datos.

Utilizar MongoDB con fulfillment de esta forma permite acceder y manipular los datos almacenados en una base de datos y utilizarlos para ofrecer respuestas más dinámicas y personalizadas a las solicitudes de los usuarios. En próximos apartados, se detallará cómo se implementa MongoDB con Dialogflow y cómo se programa la comunicación entre ambos.

### 3.1.3. Telegram

Telegram es una plataforma de mensajería instantánea, que servirá para interactuar con el usuario. Esta plataforma podrá ser utilizada como consola conectada con Dialogflow para reconocer y responder a las aportaciones de los usuarios y proporcionar información o asistencia de forma oportuna y útil.

Esta plataforma ofrece muchas ventajas, ya que, entre otras cosas llega a un público muy amplio, siendo una popular plataforma de mensajería con millones de usuarios en todo el mundo y proporciona una interfaz fácil de usar.

## 3.2. Implementación del chatbot

En el siguiente apartado, se resumirá el proceso de implementación de las diferentes herramientas que componen el chatbot para su lanzamiento. En primer lugar, se detallará la comunicación entre Dialogflow y MongoDB, dos herramientas clave en la creación y funcionamiento del chatbot. A continuación, se explicará cómo vincular Telegram, una popular aplicación de mensajería, con Dialogflow para permitir que el chatbot pueda ser utilizado a través de esta plataforma. Estos dos subapartados son esenciales para comprender cómo se ha llevado a cabo la implementación del chatbot de forma práctica y cómo se ha preparado para ser lanzado al público.

### 3.2.1. Comunicación de DialogFlow con MongoDB

En este subapartado se explicará cómo se ha llevado a cabo la comunicación entre o configuración de Dialogflow y MongoDB para que el chatbot pueda acceder a los datos necesarios para responder a las solicitudes de los usuarios. Como se ha detallado previamente, se creará una base de datos en MongoDB donde se almacenarán los datos que son solicitados por los intents de Dialogflow, tales como exámenes o datos de contacto de profesores. Luego, se utilizará Node.js para programar la respuesta del sistema ante las solicitudes de los usuarios y para preparar la respuesta más adecuada que ha de ser enviada a Dialogflow. Por último, se utilizará ngrok para lanzar la respuesta a Dialogflow y hacerla disponible para los usuarios.

Para empezar, es necesario tener una base de datos donde poder almacenar datos que puedan ser solicitados por los usuarios. Para éste proyecto, como se ha especificado previamente, se ha elegido MongoDB, debido a su gratuidad y su facilidad de vinculación y manejo con Dialogflow.

A continuación, para programar la respuesta del chatbot a las solicitudes de los usuarios se utilizará Node.js, para consulta a la base de datos MongoDB y para recuperar los datos necesarios de la misma. Node.js es

un entorno de ejecución JavaScript. Se trata de una plataforma muy popular para crear aplicaciones de servidor, sobre todo para escenarios que transcurren en tiempo real, como chatbots, juegos en línea y herramientas de colaboración en tiempo real. Una de las características clave de Node.js es su modelo no bloqueante y basado en eventos, que lo hace eficiente y ligero. Esto significa que puede manejar un gran número de conexiones coincidentes sin necesidad de múltiples hilos o procesos, lo que lo hace muy adecuado para aplicaciones en tiempo real, como es el caso de este chatbot.

Para ello, hay que instalar diferentes drivers y añadir comandos al programa para conectar el servidor de la base de datos de MongoDB. Utilizando la función `"mongoose.connect()"`, junto con algunas otras, se hará posible recopilar datos de MongoDB en el código.

En este momento, el programa será capaz de acceder al servidor de MongoDB y a las diferentes bases de datos que formen parte de él. En ese momento, cuando la herramienta Dialogflow reconoce una intent que forme parte de la lista de intents y en el caso de tener el webhook de fulfillment activado, solicitará los parámetros que sean necesarios para la petición, y los enviará al programa mediante Ngrok.

Ngrok conecta tu servidor de desarrollo local a Internet, lo que permite probar y depurar el chatbot desde cualquier lugar. Esta es una herramienta que permite crear túneles seguros, permitiéndoles conectar un servidor local a Internet. Para ello, crea un túnel mediante la creación de una URL pública que se asigna al servidor de desarrollo local. Cuando se realiza una solicitud a la URL de ngrok, se reenvía al servidor de desarrollo local, que procesa la solicitud y envía la respuesta al cliente a través del túnel.

Una vez recibida la información del nombre de la intent que se ha detectado en Dialogflow y de los diferentes parámetros necesarios para dar respuesta a la consulta, es necesario programar diferentes funciones para cada intent, donde se especificará cuál será la respuesta del sistema a la petición. Esta respuesta puede ser desde un texto básico, hasta la consulta de información o modificación de la base de datos.

En el programa crearemos distintas funciones asíncronas (`'async'`) para cada uno de los intents que pueden ser llamados desde Dialogflow, o que simplemente queramos programar la respuesta en Node.js, siendo imprescindible que el nombre de las funciones sea el mismo que el nombre de la intent en Dialogflow. Las funciones asíncronas se utilizan con el objetivo de que si en algún momento la ejecución del programa se queda bloqueada, pueda continuar ejecutándose sin ser obstaculizada o bloqueada. Además, se añade igualmente la función `'await'`, que hace esperar al programa hasta que la promesa u objeto que se requiere sea devuelto.

La función `.find()` se usará para consultar la base de datos y recuperar documentos que cumplen ciertos criterios especificados entre paréntesis. Al proporcionar un valor de variable escrito o numérico como entrada, la función buscará en la base de datos y devolverá el conjunto de resultados como una matriz de documentos. Este array se puede almacenar en un parámetro de tipo "let" para su posterior procesamiento.

Posteriormente, mediante el uso de la función `agent.add`, la función asíncrona devolverá un mensaje el cual será enviado a Dialogflow como respuesta al intent, y mostrado por la plataforma correspondiente.

El código 1 muestra un fragmento de una función del código empleado para el Chatbot de la ETSII. Si solicitas el mail de algún profesor, Dialogflow enviará el nombre de éste mediante el webhook. Además de eso, se buscará el nombre del intent que Dialogflow ha relacionado con la petición en nuestro programa. Una vez encontrada la función, en este caso, `ConsultarMailProfesor`, el nombre proveniente como parámetro de Dialogflow, se almacenará en una variable. Ésta variable de tipo "let" será la que se utilice para mediante la función `.find` acceder a la base de datos con nombre "contactodepartamentos" y buscar algún profesor con ese nombre. Posteriormente, guardará en la variable "resn" un array con la información encontrada y, en su caso, se devolverá a Dialogflow mediante la función `agent.add`

```
async function ConsultarMailProfesor(agent) {  
  
    // Se almacenan los parámetros necesarios para realizar la consulta:  
    let nombre = agent.parameters.nombreprofesor;  
    let resn =await contactodepartamentos.find({Nombre : nombre});  
    //Consulta a la base de datos  
    console.log(resn)  
    return agent.add( resn[0].Nombre+ ' pertenece al departamento de '  
    '+resn[0].Departamento+ ' y su mail es ' +resn[0].Mail);  
}
```

*Código 1. Ejemplo función consultar mail de profesor*

### 3.2.2. Vinculación de Telegram con Dialogflow

Telegram es la aplicación de mensajería instantánea elegida para poder utilizar Dialogflow de una manera más interactiva y simple. El proceso seguido para la integración de Dialogflow en Telegram es muy simple. Para empezar, se ha de crear el Bot en Telegram. Para ello se utilizará el `BotFather`. Éste es un gestor de bots, el cual permite, entre otras cosas, crear nuevos bots o configurar alguno existente. Al crear un nuevo bot, se generará un token. Un token de Telegram es un identificador único que se utiliza para autenticar y autorizar el acceso a la API de Telegram.

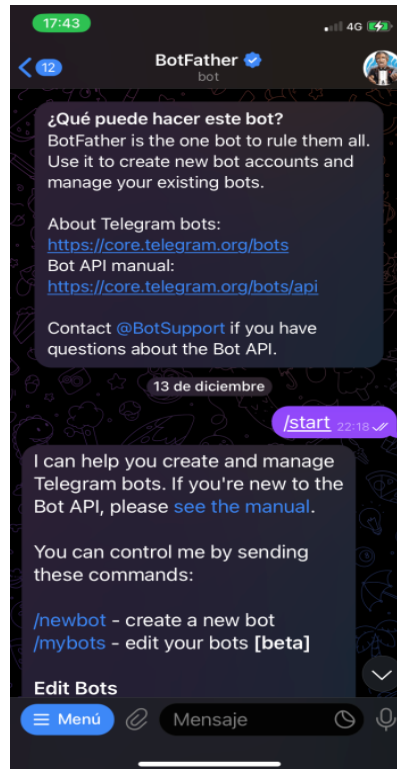


Figura 16. Creación de bot.

Como se muestra en la figura 16, el token facilitado por la aplicación, se escribirá en el correspondiente apartado de Dialogflow. Concretamente en la ventana de “Integrations” de Dialogflow, como se muestra en la figura 17.

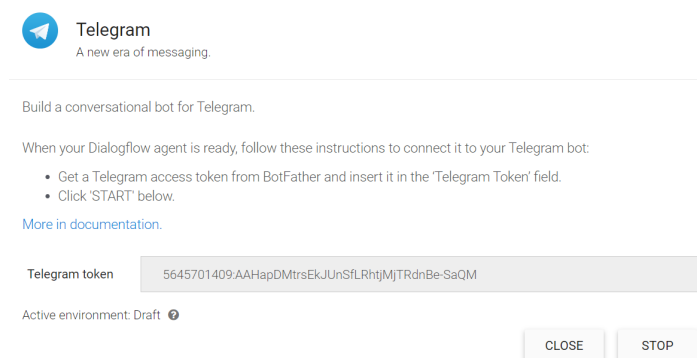


Figura 17. Integración de Bot en Telegram mediante token.

## 4. ANEXOS

Este apartado tiene como fin englobar todos los materiales complementarios o documentos utilizados para la elaboración de este proyecto, además de evidenciar el procedimiento seguido durante el mismo. Todos los ficheros mencionados a continuación se entregarán de manera conjunta a este documento.

### 4.1. Anexo I. Manual

Con el fin de facilitar el uso del ChatBot al usuario, se ha elaborado un manual de usuario donde se realiza una breve introducción sobre el concepto de ChatBot para que el usuario pueda conocer su utilidad y funcionalidades en diferentes ámbitos.

Por otro lado, se ha realizado un pequeño resumen del funcionamiento del ChatBot de la ETSII desarrollado y de sus requisitos para que el usuario pueda comenzar una conversación de forma inmediata.

Por último, se han introducido diversos ejemplos de las diferentes funcionalidades a las que el ChatBot es capaz de responder de forma que sirvan como guía para el usuario.

**Fichero:** *ANEXO\_I\_Manual\_ChatBot\_ETSII.pdf*

### 4.2. Anexo II. Demostrador

Debido a que el código debe estar en ejecución para la comprobación del chatbot, se ha generado un demostrador en formato de video que permite comprobar las diferentes funcionalidades del asistente virtual.

**Fichero:** *ANEXO\_II\_Demostrador.mp4*

### 4.3. Anexo III. Código

Finalmente y a modo del último anexo, se presentan los ficheros del código desarrollados para la base de datos y la ejecución del chatbot.

**Fichero:** *ANEXO\_III\_Codigo.zip*



## 5. REPARTO DE ROLES

El reparto de roles en un trabajo en grupo es quizás uno de los puntos claves y menos valorados para el correcto desarrollo del trabajo y el logro del objetivo final. Se basa en la asignación de responsabilidades y tareas a cada miembro del equipo con el fin de asegurar que todos los integrantes del equipo estén involucrados en el proyecto y que cada uno tenga un papel específico que llevar a cabo.

Las tareas a desempeñar y los roles varían en función del proyecto que se aborde. Así pues, siendo el caso de un proyecto de ingeniería, más concretamente de inteligencia artificial, antes de repartir los roles, se han analizado las posibles tareas que se deberían de abordar.

- **Coordinador.** Es la persona que se encarga de planificar el trabajo, dirigir y comprobar que los objetivos se cumplen. Algunas de sus tareas principales son:
  - Establecer objetivos y metas para el proyecto y asegurarse de que todos entiendan qué se espera de ellos.
  - Mantener el seguimiento del progreso del proyecto y realizar un registro con las tareas completadas.
  - Garantizar una comunicación efectiva entre los integrantes del grupo.
  - Resolver de conflictos o desafíos que surgen en el camino.
  - Convocar reuniones y encargarse de que los miembros se enteren.
- **Investigador o especialista.** Es la persona cuya tarea principal es recopilar información relevante sobre el ámbito en el que se está trabajando y comunicársela al grupo. Entre otros, su deber es:
  - Realizar una investigación inicial del tema o análisis del estado del arte con el fin de entender mejor el contexto y el alcance del proyecto.
  - Establecer una fuente fiable de información y referencias a la que acudir cuando sea necesario.
  - Organizar y documentar de manera ordenada la información encontrada para la comprensión de los otros miembros del equipo.
  - Comunicar los resultados obtenidos durante la investigación.
- **Analista.** Se trata de la persona responsable de examinar y evaluar la información recopilada por el equipo. Algunas de las tareas comunes que puede llevar a cabo un analista son:
  - Revisar la información recopilada y evaluarla en función de su veracidad e importancia.
  - Utilizar la información evaluada para sacar conclusiones.
  - Ayudar a tomar decisiones basándose en los resultados de sus análisis.
- **Documentador.** Es la persona responsable de crear y mantener documentos y registros en orden. Sus quehaceres principales son:
  - Documentar las decisiones tomadas durante las reuniones para poder llevar un seguimiento y poder comprobar si se cumplen los objetivos o no.

- Mantener la documentación ordenada y accesible para cualquier integrante en cualquier momento.
- Redactar la memoria final del trabajo y comprobar que se encuentra con un correcto formato y sin erratas.

Con el fin de poder gestionar de una manera eficiente el desarrollo del trabajo aquí presentado, se ha asignado un rol principal a cada uno de los integrantes del grupo. La asignación del rol significa que la persona en cuestión es la encargada de asegurarse de que todos los puntos se cumplen. Esto no implica que el resto del grupo no pueda participar en tareas que no se ajusten a su rol. De esta manera, por su capacidad de aprender, sintetizar y comunicar, Gorka ha sido el investigador o especialista. Debido a su previo conocimiento en el ámbito de la programación, Hugo ha sido la persona que se ha encargado de coordinar el grupo y asignar los objetivos a cumplir respecto al desarrollo de código. Finalmente, Maider es quien ha realizado las tareas de documentación necesarias durante el proceso por ser una persona ordenada, clara y concisa a la hora de redactar y emplear un uso correcto del idioma.

Por último, ya que se trata de un papel clave para una buena gestión de la información y para que no haya futuros errores en el desarrollo, no se ha querido dejar sin ocupar el rol del analista. Por esta razón, y ya que el grupo está constituido por tres personas, dicho papel ha sido repartido entre los tres miembros.

## 6. CONCLUSIONES

Para concluir con el trabajo, se puede afirmar que los chatbots inteligentes son una herramienta cada vez más popular para automatizar la comunicación con los clientes y mejorar la eficiencia en las tareas repetitivas. Se basan en tecnologías de procesamiento del lenguaje natural y aprendizaje automático para comprender y generar texto, por lo que, a medida que estas tecnologías continúen mejorando, se espera que los chatbots se vuelvan cada vez más útiles para una amplia variedad de aplicaciones como el servicio al cliente, la atención médica y la educación. Sin embargo, todavía hay desafíos por superar como la comprensión del lenguaje natural y la personalización de la experiencia del usuario.

Tal y como se ha comentado al inicio, el objetivo principal de este trabajo era realizar un estudio de alternativas para el desarrollo de un asistente virtual y la implementación del mismo en la página de la ETSII con el fin facilitar la tarea de búsqueda de información personalizada para los alumnos del centro. Así pues, a modo de resultado principal se ha obtenido un prototipo de chatbot personalizado e implementado en la aplicación de mensajería instantánea Telegram para todos los usuarios.

Con la intención de poder abarcar diferentes aspectos de información variada y debido a la extensión de la información de la página web de la ETSII, se han escogido tres másteres iniciales para realizar el prototipo del asistente (Máster Universitario en Automática y Robótica, Máster Universitario en Ingeniería Mecánica y Máster Universitario en Ciencia y Tecnología Nuclear). Así pues, una vez generado el prototipo, se podría ir añadiendo de una manera muy sencilla y mecánica la información relativa al resto de grados y másteres impartidos en la escuela.

Las principales consultas que es capaz de atender el chatbot son las siguientes:

1. Consulta de horarios
2. Consulta de mails de profesores
3. Conocer ubicación de la Escuela
4. Consulta de fechas de exámenes
5. Consulta de información sobre departamentos
6. Generación de una cita con la secretaría del centro
7. Obtención de información de la web de la Escuela

En cuanto a las herramientas empleadas para el desarrollo, cabe destacar que existen muchas alternativas diferentes en el mercado. Sin embargo, se ha demostrado que cualquier usuario sin necesidad de suscribirse a ninguna aplicación de pago puede crear un asistente virtual desde cero y con unos resultados prometedores en cuanto a comprensión del lenguaje natural, aprendizaje automático y personalización de la experiencia empleando Telegram, DialogFlow y MongoDB.

## 7. BIBLIOGRAFÍA

- [1] E. D. Gamboa Teneta, “Prototipo de un Chatbot para Compras Online Utilizando Bot Framework”, 2019.
- [2] G. Bonales, N. Pradilla y E. Citlali, “Chatbot como herramienta comunicativa durante la crisis sanitaria de la COVID-19 en España”, 2020.
- [3] K. Nimavat and P. Tushar, “Chatbots : An Overview Types , Architecture , Tools and Future Possibilities,” vol. 5, 07, pp. 1019–1024, 2017.
- [4] MongoDB Atlas. 2022. “Get Started with Atlas” [En línea] Disponible en: <https://docs.atlas.mongodb.com/getting-started/>
- [5] Hub, R., 2021. “La evolución de los chatbots: cómo han llegado para ser esenciales en nuestra vida” - Contact Center Hub. [En línea] Disponible en: <https://contactcenterhub.es/la-evolucion-de-los-chatbots-como-han-llegado-para-ser-esenciales-en-nuestra-vida-2021-25-30406/>
- [6] Builtin.com. 2022. “What is Artificial Intelligence? How Does AI Work?”. [En línea] Disponible en: <https://builtin.com/artificial-intelligence>
- [7] Brita Inteligencia Artificial. 2022. “¿Qué es un Asistente virtual?”. [En línea] Disponible en: <https://brita.mx/que-es-un-asistente-virtual>
- [8] Google Cloud. 2022. “Concepts | Dialogflow ES | Google Cloud”. [En línea] Disponible en: <https://cloud.google.com/dialogflow/es/docs/concepts>
- [9] Chatimize. 2022. “Dialogflow Review 2021: Should you use this chatbot builder?”. [En línea] Disponible en: <https://chatimize.com/reviews/dialogflow>
- [10] Tech Target. 2022. “What is Natural Language Processing? An Introduction to NLP”. [En línea] Disponible en: <https://www.techtarget.com/searchenterpriseai/definition/natural-language-processing-NLP>
- [11] P.B. Brandtzaeg y A. Følstad. “Why people use chat-bots. International Conference on Internet Science”, pp. 377-392, 2017.
- [12] Telegram. 2022. “Preguntas frecuentes - Telegram”. [En línea] Disponible en: <https://telegram.org/faq/es>
- [13] WhatsApp. 2022. “Business API”. [En línea] Disponible en: <https://www.whatsapp.com/business/api>
- [14] Datahack. 2022. “Desarrollo avanzado de chatbots: bot frameworks programables”. [En línea] Disponible en: <https://www.datahack.es/desarrollo-chatbots/>