



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería  
Informática**

**Desarrollo y explotación de  
un chatbot de FAQs sobre  
una asignatura de Trabajo  
Fin de Grado**

**UBU-Chatbot**



Presentado por Alfredo Asensio Vázquez  
en Universidad de Burgos — julio de 2021  
Tutor: Dr. Raúl Marticorena Sánchez



## **Agradecimientos**

Este trabajo supone la culminación de muchos años de trabajo y esfuerzo. No ha sido un camino fácil ni ha estado exento de dificultades, pero nunca he caminado solo, ya que han sido muchas las personas que me han acompañado y a las que estoy muy agradecido y espero haber sido capaz de transmitírselo así.

Pero quiero dedicar este logro que supone acabar Ingeniería a quienes más han sufrido con ello y siempre han estado ahí: mis padres. Gracias por todo vuestro sacrificio, apoyo y confianza. En ningún momento dudasteis de que lo conseguiría y el tiempo os ha dado la razón. Sin vosotros esto no hubiera sido posible y nada me hace más feliz que veros orgullosos de mí.

Sois los mejores padres que podría tener,

Os quiero.





UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



D. Raúl Marticorena Sánchez, profesor del departamento de Ingeniería Informática, área de Lenguajes y Sistemas Informáticos.

Expone:

Que el alumno D. Alfredo Asensio Vázquez, con DNI 73007684A, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado "Desarrollo y explotación de un chatbot de FAQs sobre una asignatura de Trabajo Fin de Grado".

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 29 de junio de 2021

Vº. Bº. del Tutor:

D. Raúl Marticorena Sánchez



## **Resumen**

Los estudiantes que realizan el TFG a menudo tienen dudas acerca de su organización y funcionamiento. Muchas de estas dudas están ya recogidas en la documentación de la plataforma Moodle UBUVirtual o en los distintos documentos relacionados del portal web de la Universidad.

En este trabajo, se desarrolla un bot conversacional (chatbot) cuyo objetivo es dar respuesta a todas estas dudas que los alumnos puedan tener sobre el funcionamiento de la asignatura Trabajo de Fin de Grado, reduciendo así también la carga de trabajo de los docentes.

Es también una de las primeras aproximaciones de la UBU a la implementación de una herramienta en auge como son los chatbots, que pueden aportar mucho valor en una futura explotación y expansión en las distintas plataformas que utiliza la Universidad.

El ámbito del trabajo se centra principalmente en la integración del chatbot en la plataforma Moodle de la Universidad, llamada UBUVirtual, aunque no se limita a esta única plataforma, ya que también se realizará una integración con la plataforma Slack.

## **Descriptores**

Chatbot, bot conversacional, agente inteligente, Dialogflow, Trabajo Fin de Grado, Moodle, Slack.

## **Abstract**

Students who take the End-of-Degree project often have doubts about its organization and operation. Many of these doubts are already collected in the documentation of the Moodle UBUVirtual platform or in the different related documents of the University web portal.

In this work we develop a chatbot whose objective is to answer all these doubts that students may have about the End of Degree Project subject, thus also reducing the workload of teachers.

It is also one of the UBU's first approaches to the implementation of a booming tool such as chatbots, which can add a lot of value in future exploitation and expansion on the different platforms used by the University. .

The scope of the work is mainly focused on the integration of the chatbot in the University's Moodle platform, called UBUVirtual, although it is not limited to this single platform, since an integration with the Slack platform will also be carried out.

## **Keywords**

Chatbot, intelligent agent, Dialogflow, End-of-Degree Project, Moodle, Slack.



---

# Índice general

---

<b>Índice general</b>	<b>III</b>
<b>Índice de figuras</b>	<b>V</b>
<b>Índice de tablas</b>	<b>VI</b>
<b>Introducción</b>	<b>1</b>
1.1. Contenido del trabajo . . . . .	1
1.2. Estructura de la memoria . . . . .	2
1.3. Materiales adjuntos . . . . .	2
<b>Objetivos del proyecto</b>	<b>5</b>
2.1. Objetivos generales . . . . .	5
2.2. Objetivos técnicos . . . . .	5
<b>Conceptos teóricos</b>	<b>7</b>
3.1. Chatbot . . . . .	7
3.2. Conversational UI . . . . .	9
3.3. Machine Learning . . . . .	11
3.4. NLP y NLU . . . . .	12
3.5. Flow-based chatbot vs AI Powered chatbot . . . . .	13
3.6. Intent, Entity, Context, Fallback y Event . . . . .	15
3.7. Rich responses . . . . .	16
3.8. Webhook . . . . .	17
<b>Técnicas y herramientas</b>	<b>19</b>
4.1. Plataformas para la construcción de Chatbots . . . . .	19

4.2. Documentación . . . . .	25
4.3. Control de versiones . . . . .	25
<b>Aspectos relevantes del desarrollo del proyecto</b>	<b>27</b>
5.1. Inicio y análisis de posibles soluciones . . . . .	27
5.2. Formación y metodologías . . . . .	29
5.3. Implementación básica e integración en UBUVirtual . . . . .	30
5.4. Estudio de características avanzadas e implementación en Slack . . . . .	32
5.5. Proceso de entrenamiento . . . . .	34
5.6. Análisis de resultados . . . . .	35
<b>Trabajos relacionados</b>	<b>39</b>
6.1. Proyectos de la Universidad de Burgos . . . . .	40
6.2. Proyectos externos . . . . .	42
6.3. Tabla comparativa del proyecto . . . . .	43
<b>Conclusiones y Líneas de trabajo futuras</b>	<b>45</b>
7.1. Conclusiones . . . . .	45
7.2. Líneas de trabajo futuras . . . . .	47
<b>Bibliografía</b>	<b>49</b>

---

# Índice de figuras

---

3.1. Interfaz textual . . . . .	9
3.2. Interfaz de usuario gráfica . . . . .	10
3.3. Conversational UI. . . . .	10
3.4. NLU y NLP [1] . . . . .	12
3.5. Ejemplo de flujo de conversación guiada . . . . .	13
3.6. Rich response [2] . . . . .	16
3.7. <i>Fulfillment flow</i> [3] . . . . .	17
4.1. Logos plataformas analizadas. . . . .	19
5.1. Logo del proyecto. . . . .	28
5.2. Integración en Moodle. . . . .	31
5.3. Modelo arquitectónico [21] . . . . .	33
5.4. Dialogflow Analytics: detección de intents. . . . .	35
5.5. Totales extraídos del análisis numérico. . . . .	36
5.6. Evolución temporal de la tasa de acierto. . . . .	37
6.1. UBUAssistant logo . . . . .	40
6.2. Tourist Bot logo . . . . .	40
6.3. UBUVoiceAssistant logo . . . . .	41
6.4. Logo y funcionalidades de Lola . . . . .	42

---

# Índice de tablas

---

4.1. Plataformas de desarrollo de Chatbots. . . . .	25
6.1. Plataformas de desarrollo de Chatbots. . . . .	43

---

# Introducción

---

## 1.1. Contenido del trabajo

Llegado al último curso de grado todos los alumnos de la UBU deben realizar su Trabajo de Fin de Grado (TFG) y son muchas las dudas que tienen acerca del funcionamiento del mismo, especialmente en la enseñanza online en la que al no haber un contacto tan estrecho con profesores y otros alumnos se tiene menos información. Motivo por el que los alumnos, aparte de consultar la extensa documentación aportada en las distintas plataformas de la UBU consultan a los profesores por email para despejar sus dudas.

Muchas de estas preguntas acerca del trabajo ya han sido respondidas anteriormente, o están explicadas en alguna parte de la documentación de UBUVirtual o de la página del grado. Con el propósito de ayudar a los alumnos a despejar estas dudas y reducir la carga de preguntas que reciben los profesores, se desarrolla este chatbot cuya finalidad es la de dar respuesta de forma automática a las preguntas de los usuarios acerca de todos los aspectos relacionados con la asignatura. Como restricción ha de utilizar únicamente herramientas gratuitas. El chatbot está integrado en la plataforma de UBUVirtual de la asignatura de TFG, a la que todos los alumnos matriculados en el trabajo tienen acceso.

Por medio del aprendizaje automático nuestro chatbot será capaz de entender y responder las preguntas que los alumnos le formulen, y a su vez aprenderá de aquellas que no las haya recibido anteriormente para ser capaz de responderlas en la próxima ocasión en que se le pregunten. Con ello resolveremos la doble problemática para profesores y alumnos, con la ventaja añadida de que el bot responderá ininterrumpidamente a cualquier hora del día.

## 1.2. Estructura de la memoria

La memoria utiliza la siguiente estructura:

- **Introducción:** resumen del contenido del trabajo: problema, proceso y solución. Estructura de la memoria y listado de los materiales adjuntos.
- **Objetivos del proyecto:** listado de objetivos generales y técnicos del proyecto.
- **Conceptos teóricos:** explicación de los conceptos teóricos necesarios para comprender el proyecto.
- **Técnicas y herramientas:** listado de técnicas y herramientas analizadas para el desarrollo. Comparativa y justificación de la elección.
- **Aspectos relevantes del desarrollo del proyecto:** resumen de las distintas fases del desarrollo. Análisis de resultados obtenidos por el programa desarrollado.
- **Trabajos relacionados:** estado del arte en el campo de los *chatbots* en la UBU y otras unionidades españolas.
- **Conclusiones y Líneas de trabajo futuras:** conclusiones extraídas de la elaboración del proyecto y pasos a seguir en la futura explotación y mejora.

La estructura de los anexos es la siguiente:

- **Plan de Proyecto Software:** planificación temporal y estudio de viabilidad económica y legal del proyecto.
- **Especificación de requisitos:** fase de análisis: objetivos generales, catálogo de requisitos y especificación de requisitos.
- **Especificación de diseño:** diseño de datos, diseño procedimental y diseño arquitectónico.
- **Documentación técnica de programación:** estructura de directorios, manual del programador, instalación del proyecto y pruebas del sistema.
- **Documentación de usuario:** manual para que el usuario sea capaz de utilizar de manera adecuada el *chatbot* en sus distintas integraciones

## 1.3. Materiales adjuntos

Los materiales adjuntos al proyecto son los siguientes:

- Proyecto de Dialogflow para el Chatbot de la modalidad online.

- Proyecto de Dialogflow para el Chatbot para la modalidad presencial.
- Listado de preguntas y respuestas para la modalidad online.
- Listado de preguntas y respuestas para la modalidad presencial.
- Código para la integración HTML en UBUVirtual modalidad online.
- Código para la integración HTML en UBUVirtual modalidad presencial.

Están disponibles a través de Internet en el siguiente recurso:

- Repositorio del proyecto en Github [\[4\]](#).





---

# Objetivos del proyecto

---

Estos son los distintos objetivos con los que se ha realizado el proyecto.

## 2.1. Objetivos generales

- Estudiar y evaluar las distintas plataformas y *frameworks* que permiten trabajar con *chatbots*, seleccionando la más adecuada de entre aquellas gratuitas y con soporte para castellano.
- Desarrollar un *chatbot* que permita interaccionar con los estudiantes y darle respuestas a sus dudas con respecto a la asignatura del Trabajo de Fin de Grado para su versión online.
- Desarrollar un segundo *chatbot* para la versión presencial con el objetivo de recibir mayor número de logs.
- Integrarlos en la plataforma de UBUVirtual, bajo las restricciones dadas por la Universidad en cuanto a la edición del código fuente de la web.
- Integrar también uno de ellos en otra plataforma.
- Implementar un sistema de *logs* que permita monitorizar las interacciones con los usuarios.
- Explotar los datos obtenidos en los *logs*, para aplicarles un análisis numérico del que extraer información.

## 2.2. Objetivos técnicos

- Implementar los *chatbots* en la plataforma Dialogflow.
- Utilizar la herramienta Dialogflow Messenger para disponer de las respuestas avanzadas.

- Utilizar las *payload response* para dotar a nuestro chatbot de funcionalidades más avanzadas.
- Obtener el código HTML para su integración.
- Generar el código CSS que le de una apariencia estéticamente agradable al integrarlo en UBUVirtual.
- Generar un Moodle propio para testear la integración.
- Integrar los *chatbots* en UBUVirtual.
- Implementar e integrar el *chatbot* de modalidad online en Slack.
- Monitorizar las conversaciones de los usuarios.
- Realizar un estudio numérico de los datos obtenidos en los *logs*.
- Utilizar el control de versiones mediante GitHub.

---

# Conceptos teóricos

---

Al tratarse de un proyecto innovador en cuanto a la tecnología utilizada, al no haberse realizado previamente ningún otro proyecto que la utilice, son muchos los conceptos a desarrollar para la comprensión general del funcionamiento de los chatbots y, especialmente, del que realizamos con la tecnología de Dialogflow.

En primer lugar, se da una descripción de conceptos generales comunes a todos los chatbots y posteriormente se profundiza en aspectos propios de la plataforma de Google que estamos utilizando.

## 3.1. Chatbot

Un chatbot es una aplicación software que interactúa con el usuario a través de un chat, generalmente de texto, aunque también puede ser de voz, por medio de respuestas automáticas. Fueron creados para simular el modo en que se comportaría un agente humano, un reto de la Inteligencia Artificial que nunca ha sido sencillo, prueba de ello la tenemos en el test de Turing.

El test de Turing, establecido en 1950 por el británico Alan Turing, considerado uno de los pilares sobre los que se inició la ciencia de la computación, establece que para determinar que un sistema inteligente diseñado para imitar a un humano supere la prueba, debe conseguir engañar al menos al treinta por ciento de los jueces haciéndoles creer que es un humano.

Turing predijo que esto no se produciría hasta el año 2000, pero no fue hasta 2014 cuando Eugene Goostman -un chatbot creado en San Petersburgo por los programadores Vladimir Veselov (Rusia), Eugene Demchenko (Ucrania) y Sergey Ulasen (Rusia)- superó por primera vez esta prueba

tras engañar al 33 % de los jueces. Este chatbot se hacía pasar por un niño ucraniano de trece años. Este es uno de los motivos por el que ha recibido muchas críticas: la prueba se realizaba en inglés, por lo que era de suponer que un niño ucraniano cometiese ciertos errores y los jueces los atribuirían a su origen ucraniano. Además, al tratarse de un niño de 13 años se le pasaban por alto muchos errores achacables a su edad. [5]

En el mundo real, los chatbots no se están diseñando para superar este test, sino con fines comerciales, ya que con ellos se consiguen una serie de beneficios:

- Reducir el coste de la mano de obra, al poder prescindir de operarios cuya función pasa a desempeñarla el chatbot.
- Aumentar la capacidad de trabajo, ya que un agente humano solo puede atender a una conversación en cada instante, sin embargo, el chatbot puede atender a miles de potenciales clientes al mismo tiempo.
- Mayor velocidad de respuesta, especialmente si se trata de consultas en bases de datos; un chatbot la realiza de forma automatizada en un tiempo mucho menor que el que tardaría un humano.

La implementación de la Inteligencia Artificial en el mundo laboral está teniendo un gran impacto. Tal es así, que estudios como el del McKinsey Global Institute [6] afirman que en 2030, los robots e Inteligencia Artificial podrían sustituir el 30 % de los puestos de trabajos realizados por humanos. Esto supone un problema por la desaparición de millones de puestos de trabajo, pero tecnológicamente es un avance muy importante.

## 3.2. Conversational UI

Las interfaces de usuario conversacional han ido evolucionando a lo largo de los años. Inicialmente, la única forma posible de interacción entre humanos y computadores era una interfaz textual.

Esta interfaz primigenia comenzó utilizando comandos con un formato estricto y evolucionó hacia el lenguaje natural. Un ejemplo de interacción textual con lenguaje natural es la búsqueda en Google, donde introducimos una frase en lenguaje natural y obtenemos una serie de resultados como es muestra en la Figura 3.1. Actualmente este ejemplo ha evolucionado y no es estrictamente textual, ya que tanto para la entrada como salida se pueden incluir archivos multimedia.

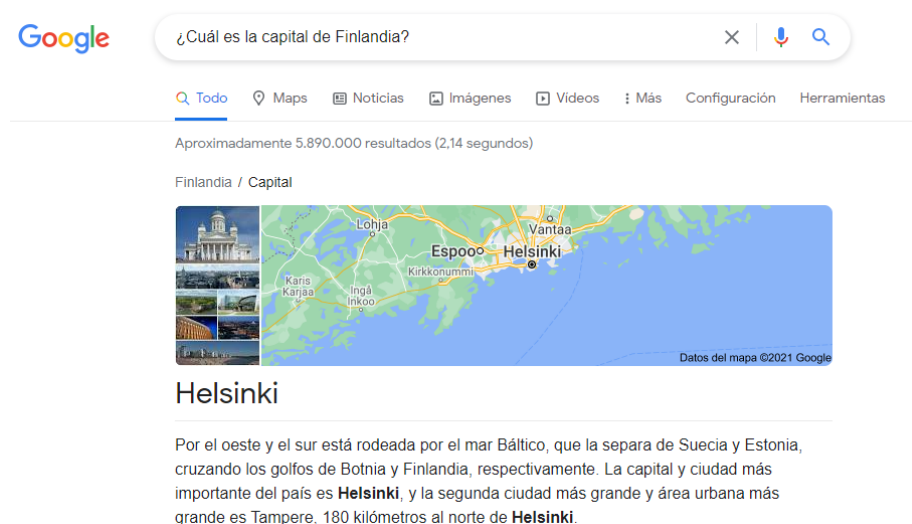


Figura 3.1: Interfaz textual

La siguiente evolución en este campo fueron las interfaces gráficas *Graphical user interface*. Esta interfaz imita el modo en que realizamos las acciones en la vida real y reemplaza la interacción textual. Ver Figura 3.2.

Con esta interfaz podemos realizar acciones como pulsar un botón de la pantalla o seleccionar una opción en una lista.

Con la introducción de Windows en la década de los 90 esta interfaz adquirió una gran popularidad. Siguieron evolucionando con la introducción de elementos táctiles, que evitan la necesidad de utilizar un ratón.

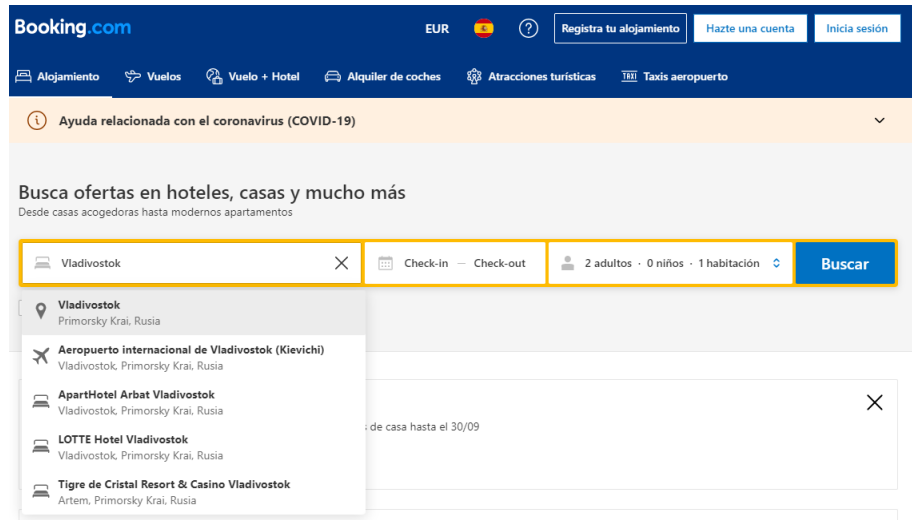


Figura 3.2: Interfaz de usuario gráfica

Finalmente, la última evolución de las interfaces conversacionales ha sido la *Conversational UI*, que es sobre la que se diseña este proyecto. Conceptualmente es parecida a la interfaz textual, pero en este caso se produce una interacción de una manera similar a la que realizarían los humanos entre sí. A este computador humanizado se le denomina *bot*. Sigue el esquema mostrado en la Figura 3.3

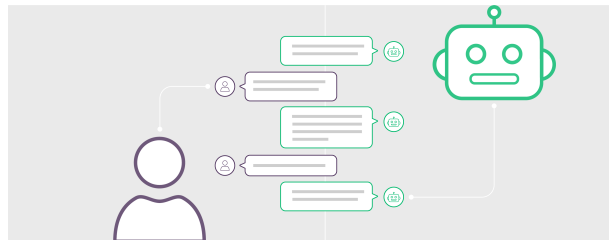


Figura 3.3: Conversational UI.

Finalmente, existe otro tipo de *Conversational UI*, cuya entrada son mensajes de audio en lugar de mensajes de texto, y se denomina *Voice-enabled conversational UI*. Este nuevo modelo imita la interacción humana más que ningún otro.

Amazon Alexa, Google Assistant y Siri son dos de los ejemplos más populares de sistemas que implementan este tipo de interfaz.

### 3.3. Machine Learning

El *Machine Learning*, de aquí en adelante Aprendizaje Automático, hace referencia a los sistemas que están basados en algoritmos que permiten al sistema aprender de manera autónoma a partir de la información analizada, sin necesitar la intervención de un operario humano.

El Aprendizaje Automático esta fundamentado en la estadística, por medio del reconocimiento de patrones. Parte de la idea de que un conjunto de reglas podría representarse de una forma menos explícita. Algunos algoritmos de aprendizaje automático son: regresión, bayesianos, agrupación, árboles de decisión y redes neuronales.

En nuestro proyecto, cuando introducimos los datos de miento, Dialogflow crea un modelo de Aprendizaje Automático diseñado específicamente para nuestro agente. Cada vez que guardemos cambios en nuestro agente, se producirá automáticamente un entrenamiento.

En el caso de nuestro agente, responde FAQs a partir de unas frases de ejemplo que le demos para referirnos a las posibles distintas formulaciones de una misma pregunta que nos puedan hacer. El agente, por medio de un Aprendizaje Automático, generará información sobre el tipo de preguntas que nos puedan hacer similares a las introducidas que busquen obtener la misma información.

Esto es de gran utilidad, ya que una misma pregunta se puede formular de miles de maneras diferentes y sería una tarea prácticamente imposible introducir a mano cada una de ellas.

### 3.4. NLP y NLU

NLP: *Natural language processing* es un campo de las ciencias de la computación, inteligencia artificial y lingüística que estudia la interacción entre el lenguaje de las computadoras y el lenguaje de los humanos. Tiene como propósito analizar la capacidad de una máquina para procesar texto en lenguaje natural, descomponerlo, entender su significado y determinar la acción que debe tomar como respuesta al texto recibido.

NLU: *Natural Language Understanding* es una rama dentro de NLP cuyo objetivo es comprender el significado de la entrada de texto recibida como lo harían los humanos. Convierte las entradas no estructuradas en una información estructurada que es capaz de manejar. Esto se refiere a, entre otras labores, que sea capaz de entender una frase con faltas de ortografía, con palabras colocadas en un orden intercambiado, coloquialismos y contextos.

Como ejemplo práctico de aplicación de este concepto en nuestro proyecto, nuestro chatbot debe ser capaz de entender que le están preguntando por la fecha de matriculación independientemente de si se lo preguntan de cualquiera de estas maneras: *Fecha de matriculación*, *¿Cuándo es la fecha de matriculación?* *Fecha para matricularme*. O de cualquiera de las millones de formas diferentes de hacer esa pregunta. El NLU es el que se encarga de cumplir esta tarea.

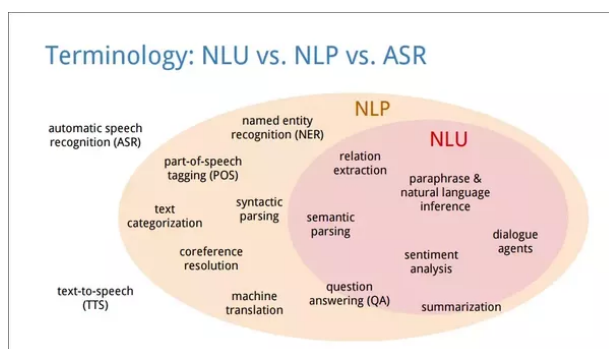


Figura 3.4: NLU y NLP [1]

En la Figura 3.4 se muestra la relación de estos conceptos y las funciones de cada uno de ellos. ASR (*Automatic speech recognition*) es un sistema de reconocimiento de habla (que queda fuera del marco de este trabajo).



## 3.5. Flow-based chatbot vs AI Powered chatbot

### *Flow-based chatbot*

No tiene componente NLU, por lo que no puede entender los mensajes que escriba el usuario. Consiste en un flujo de conversación que sigue una estructura de árbol, y en la que se le sugiere al usuario una serie de opciones que corresponderán a la operación que quiera hacer el usuario en cada momento, el usuario en ningún momento va a poder introducir texto, únicamente irá eligiendo entre las opciones que el chatbot le ofrezca tras cada elección. En la Figura 3.5 se muestra un ejemplo.

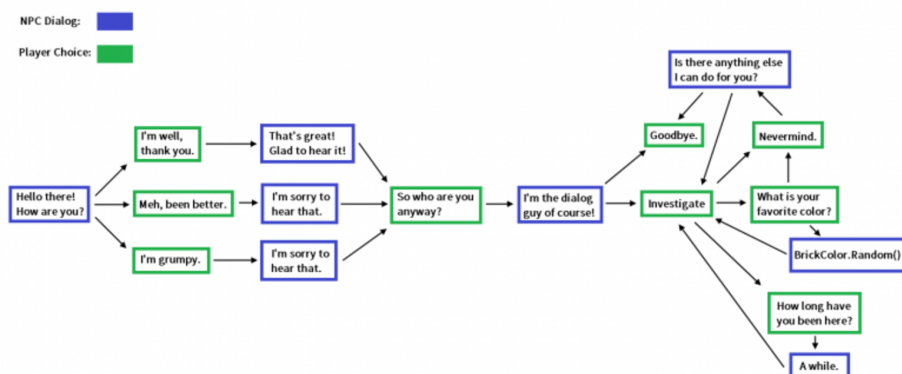


Figura 3.5: Ejemplo de flujo de conversación guiada

Está diseñado para realizar un número limitado de operaciones y no ofrece margen de maniobra al usuario. Sin embargo, tiene la ventaja de que al ser una conversación inflexible se evitan errores y dar una mala experiencia al usuario, algo muy habitual en un chatbot abierto. Son más sencillos de implementar que un chatbot que siga el modelo de NLU que se explica a continuación.

### *AI Powered chatbot*

Su componente NLU es el elemento principal en torno al cual gira el motor del chatbot. Por medio del NLU procesa y analiza el mensaje que introduce el usuario. Se le brinda al usuario total libertad para introducir el texto que desee. Está basado en el Aprendizaje Automático, lo que significa que va aprendiendo en base a sus ejecuciones.

Aporta mucho más margen de maniobra, más posibilidades de recibir incrementos en sus funcionalidades, ofrece una experiencia más completa y puede detectar patrones de comportamiento. En su contra acostumbra a no detectar correctamente lo que el usuario quiere decir con relativa frecuencia.

Nuestro chatbot se basa en una solución de este tipo, ya que un FAQ que estuviera implementado basado en un flujo de conversación, estaría muy limitado, ya que el abanico de posibles preguntas de los alumnos es muy amplio, y solo se puede cubrir con un motor basado en NLP

## **Solución híbrida**

Una tercera opción es la unión de las dos soluciones anteriores, siendo un chatbot de texto abierto del usuario que en ocasiones tenga un flujo de conversación guiado.

## 3.6. Intent, Entity, Context, Fallback y Event

**Intent:** es el objetivo que tiene un cliente cuando formula una pregunta. Se implementan asociando a cada *intent* varias frases de entrenamiento (*training phrases*) cuyo objetivo sea el de realizar la misma pregunta que la del propósito de dicho *intent*. Cada vez que un usuario hace una pregunta el NLU determina cuál es el *intent* que más se aproxima a ella y será la respuesta asociada a dicho *intent* la que responda, por lo que los *intents* son un factor crítico para el correcto flujo de la conversación. Debe generarse un buen modelo con muchos datos de entrenamiento. [7]

**Entity:** es una clase de dato que puede extraerse del mensaje del usuario. El NER (*Named Entity Recognition*) es la parte del NLP encargada de extraer entidades del texto. Ejemplos de entidades: teléfono, email, ciudad y números.

**Context:** similar a los contextos de una conversación real. Son variables que sirven para determinar el estado de la conversación, utilizándose para establecer cuál es el camino que debe seguir la conversación. [8]

**Fallback:** caso en el que el chatbot no ha sido capaz de reconocer la entrada del usuario y asociarla a un *intent*. Se pueden generar *intents* de tipo *fallback* para gestionar el flujo de la conversación en caso de que se produzcan.

**Event:** los eventos sirven para disparar *intents* sin necesidad de recibir una entrada por parte del usuario. Un ejemplo de evento es cuando el usuario abre el chatbot.

### 3.7. Rich responses

Son respuestas que incorporan más elementos que el texto plano. Existen muchos tipos: botones de sugerencia, imágenes, mensajes de audio, enlaces personalizados, etc.

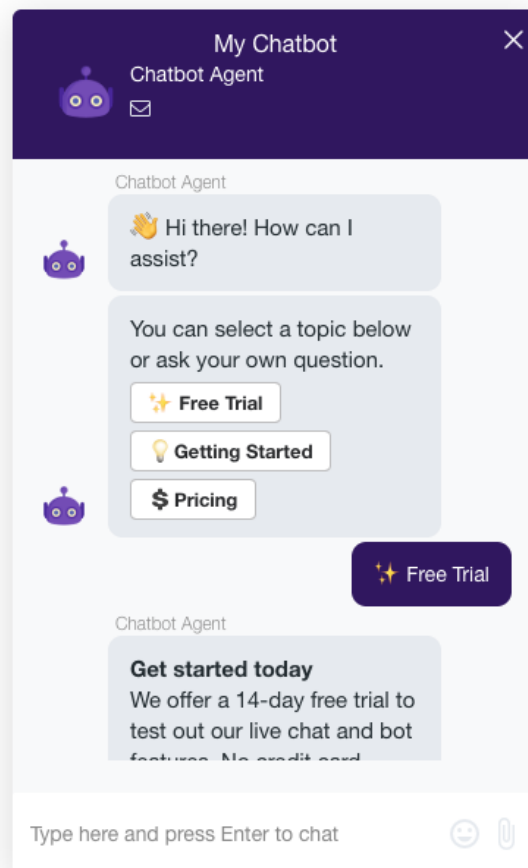


Figura 3.6: Rich response [2]

En la Figura 3.6 se muestra una respuesta con botones de sugerencia (*suggestion chips*). Por medio de este tipo de respuestas se facilita mucho el diseño guiado por flujos.

Las *rich responses* disponibles dependen de la integración que se utilice.

## 3.8. Webhook

Un *webhook* es un mecanismo para enviar información al sistema sobre la información generada por eventos de usuario mediante una petición HTTP. [9]

Dialogflow utiliza *fulfillment* que es el código que se despliega y permite al agente ejecutar la lógica del programa como respuesta a un *webhook* que puede ser desencadenado tras el reconocimiento de un *intent*. Esto permite utilizar la información obtenida por el NLP para generar respuestas dinámicas en el *back-end*.

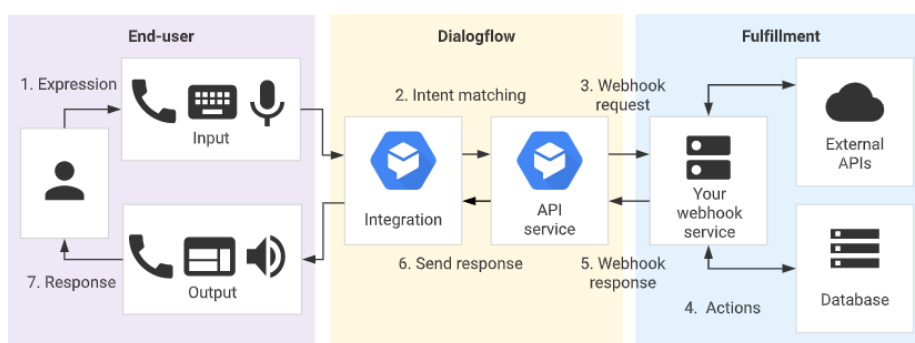


Figura 3.7: *Fulfillment flow* [3]

En la Figura 3.7 se muestra el flujo por el que Dialogflow manda una petición al *webhook* con la información asociada al *intent*. El sistema por medio del *fulfillment* puede realizar cualquier acción sobre ella, como consultas a APIs externas o a base de datos.

Un ejemplo de uso podría ser que un usuario preguntase al chatbot el horario de tutorías de un profesor. En primer lugar, se detectaría un *intent* de consulta de horarios de tutoría que activaría una petición *webhook* por la que se ejecutaría en nuestro *fulfillment* un código que haría una consulta a una API externa en la que estaría la información de los horarios de cada profesor, para devolver esa información como un mensaje de texto en el chatbot a nuestro usuario.



---

## Técnicas y herramientas

---

### 4.1. Plataformas para la construcción de Chatbots

En esta sección vamos a analizar algunas de las herramientas (ver Figura 4.1) más extendidas para el desarrollo de chatbots. Dialogflow queda exento de esta sección ya que al ser la plataforma elegida será desarrollada más ampliamente en el siguiente apartado.

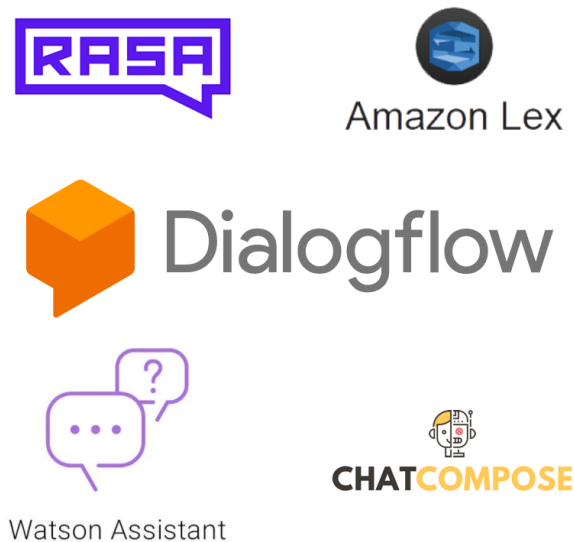


Figura 4.1: Logos plataformas analizadas.

## Rasa [10]

Rasa es un framework de aprendizaje automático y código abierto para la creación de chatbots. Está escrito en Python y sirve para desarrollar los chatbots basados tanto en texto como voz. [11]

Está compuesto por:

- NLU - Comprensión del lenguaje natural. Es el componente que se encarga de procesar la entrada de texto o voz recibida y convertirla en un valor que la máquina pueda entender.
- Core - Es el componente que en base a unos algoritmos y una entrada recibida, decide la acción que el agente debe tomar para continuar la conversación de manera más adecuada. Para ello asigna un valor a cada posible respuesta que se corresponde a la similitud encontrada entre el mensaje que dispara esa respuesta y la entrada recibida.

Por medio de librerías, se puede adaptar el NLU a cualquier idioma.

Las desventajas de esta herramienta son su curva de aprendizaje, la carencia de *hosting* en la nube, no dispone de integraciones *out of box* (que no requieren de instalación) y no dispone en su versión de gratuita de interfaz gráfica para la construcción del chatbot.



## Amazon Lex [12]

Amazon Lex es un servicio de Amazon para la creación de chatbots, basados tanto en texto como voz. Utiliza el mismo motor que Amazon Alexa, un potente asistente virtual lanzado por Amazon en 2013, y actualmente el más popular del mundo con un 62% de la cuota de mercado en 2017.[13]

Ofrece NLU para la comprensión del lenguaje natural y ASR para el reconocimiento de voz.

Sus principales características son:

- Dispone de una interfaz gráfica web para el desarrollo, lo que reduce la curva de aprendizaje.
- Permite integraciones con servicios de la plataforma AWS.
- Soporte para diez idiomas, castellano incluido.
- Integraciones *out of box* con Slack y Facebook, entre otras.

Como principal desventaja, a partir del segundo año pasa a ser de pago, por lo que no cumple la principal restricción del proyecto.

## Watson Assistant [14]

Watson Assistant es un producto de IBM para el desarrollo de chatbots conversacionales. Permite el salto de un nodo de la conversación a otro, lo cual en nuestra herramienta elegida (Dialogflow) no es posible, teniendo que duplicar diálogos en ocasiones.

Sus principales características son:

- Interfaz gráfica web para el desarrollo, sencilla e intuitiva.
- Soporte para trece idiomas, castellano incluido.
- Genera un *script* para una sencilla integración web.
- Fácil integración con Slack, Whatsapp y Facebook, entre otras.

Dispone de una versión gratuita, la cual podría cubrir todas las necesidades de nuestro proyecto. Esta versión limita a mil usuarios al mes, lo cual no supone un problema para el espectro de este proyecto; ante una posible futura expansión del asistente a un mayor número de asignaturas o incluso la página oficial de la UBU, podría llegar a suponer un problema.

Siendo esta una herramienta que cumple todos los requisitos de nuestro proyecto, se ha realizado una comparativa con Dialogflow para determinar qué plataforma era más recomendable utilizar. A favor de Watson se ha detectado la ventaja del salto de nodos comentada anteriormente, pero en su contra, todas estas desventajas:

- Menos idiomas e integraciones que Dialogflow.
- No permite diálogos multidioma, lo cual en una futura actualización inclusiva para los estudiantes Erasmus podría ser beneficioso; ya que junto al punto anterior se podría dar respuesta en mayor número de idiomas.
- Está enfocado para desarrollos más complejos que el de nuestro Q/A, para lo cual Dialogflow está mejor preparado.

## Chatcompose [15]

Chatcompose es una plataforma web para el desarrollo de chatbots. Es una herramienta muy potente por su rápida personalización, implementación y despliegue en múltiples plataformas. Permite añadir *rich responses* de manera automática, sin necesidad de ningún tipo de código o lenguaje especial, simplemente con las herramientas que nos proporciona su interfaz gráfica.

A diferencia del resto de herramientas, proporciona *live chat*, en el que un agente humano puede tomar parte de la conversación. Esta funcionalidad no resulta interesante para nuestra aplicación, ya que no se pretende que ningún miembro de la UBU responda las preguntas, sino que se delega todo en el chatbot.

Sus principales ventajas son:

- Soporte para 32 idiomas, castellano incluido.
- Única herramienta estudiada con soporte propio para catalán.
- Integración con Slack, Telegram, Facebook, Whatsapp, SMS y Line entre otras.
- Sencilla y ampliamente personalizable.

Es una herramienta muy interesante, pero en su contra tiene que la versión gratuita tiene muchísimas limitaciones, la principal es que únicamente permite desarrollar un chatbot. Para poder hacer un uso efectivo de la plataforma deberíamos recurrir a la versión de pago.

Además, Dialogflow está mejor preparado para conversaciones abiertas, por su fácil y rápida implementación de preguntas y respuestas. Chatcompose sería más eficaz si se guiase al usuario a través de un flujo de conversación, ya que permite implementar las *suggestion chips* sin necesidad de enviar la información en formato JSON como requiere Dialogflow.

## Otras plataformas

Las plataformas analizadas representan un pequeño número de la inmensa cantidad de plataformas para el desarrollo de chatbots que existen en la actualidad. Se han seleccionado esas plataformas debido a que en la fecha de inicio del proyecto contaban con soporte nativo para castellano y eran algunas de las soluciones más populares. No obstante, cabe aclarar que existen diversas alternativas que también cumplen dicho criterio y han quedado fuera porque el objetivo de este trabajo no es el de analizar todas las posibles soluciones -que serían muchas-, sino realizar un estudio más exhaustivo de cinco de ellas.

A continuación se resumen otras alternativas:

- **Microsoft Bot Framework:** también llamado Azure Bot Service, es una solución de Microsoft. Su principal fortaleza es la integración en el ecosistema tecnológico de Microsoft, incluyendo Office y Teams como principales integraciones de las que carecen otras plataformas. Cuenta con soporte para castellano, pero la versión gratuita está limitada a 10000 mensajes mensuales. [16]
- **Chatfuel:** Es una herramienta muy popular, pero únicamente dispone de integración con Facebook Messenger e Instagram. [17]
- **Wit.Ai:** Cuenta con soporte para castellano y es gratuita, aunque con limitaciones. Su punto flojo es la poca documentación e información de la que se dispone. [18]
- **Flow XO:** Ofrece más de 100 integraciones, incluyendo YouTube, GitHub, LinkedIn, Office y MySQL entre otras. No dispone de versión gratuita. [19]

Plataforma	Castellano	UI Based	Slack Integration	Hosting	Gratuita
Dialogflow	X	X	X	X	X
Rasa	X				X
Amazon Lex	X	X	X	X	
IBM Watson	X	X	X	X	X
Chatcompose	X	X	X	X	

Tabla 4.1: Plataformas de desarrollo de Chatbots.

### Tabla comparativa

En la tabla 4.1 se muestra la comparativa entre las distintas plataformas estudiadas en los aspectos que resultan más relevantes para el proyecto.

## 4.2. Documentación

- Herramientas consideradas: [TeXstudio](#) y [Microsoft Word](#).
- Herramienta elegida: [TeXstudio](#).

Office Word fue la primera opción para la documentación debido a la familiaridad con su uso. Pero al partir de la plantilla LaTeX suministrada por la Universidad resultó una opción mucho más cómoda utilizar la aplicación de escritorio TeXstudio. Se trata de una editor de LaTeX de código abierto. El principal motivo fue su interfaz y la opción de 'Compilar y ver' que permite visualizar al mismo tiempo el código y el resultado en PDF.

## 4.3. Control de versiones

- Herramientas consideradas: [GitHub](#), [GitLab](#) y [Cloud Source Repositories](#).
- Herramienta elegida: [GitHub](#).

Las tres herramientas consideradas para el control de versiones me resultaban familiares al haber trabajado con ellas previamente. Me decanté por GitHub al disponer ya de una cuenta de la Universidad y ser la más utilizada a lo largo del grado.



---

# Aspectos relevantes del desarrollo del proyecto

---

En este apartado se va a realizar un análisis del camino seguido en el desarrollo de este proyecto, haciendo hincapié en los aspectos más relevantes. El apartado sigue un orden cronológico en el que se empieza por la fase inicial de estudio del problema y planteamiento de la solución, y se acaba por el análisis final de resultados obtenidos y su documentación.

## 5.1. Inicio y análisis de posibles soluciones

Tras analizar las propuestas de trabajos publicadas por los profesores me resultó interesante el desarrollo de un *chatbot* para la plataforma de UBUVirtual, dado que es un trabajo que considero que va a aportar valor a la Universidad y será de utilidad en el futuro. Esto era algo importante para mí, ya que he observado que muchos de los proyectos de fin de grado acaban quedando obsoletos y no reciben ningún uso una vez presentados.

Por otra parte, en mi trabajo desempeñaba funciones relacionadas con *chatbots*, por lo que partía de unos conocimientos previos de algunas tecnologías. Como primer paso del proyecto, hube de realizar un estudio exhaustivo de todas las herramientas que podrían ayudar a realizar el cometido, así como de las limitaciones que cada una de ellas tenía.

Tras este análisis -en el que creé pequeños proyectos en varias soluciones de diferentes proveedores- determiné que la solución óptima era utilizar la versión gratuita que ofrece Dialogflow (*Trial Edition*) [20].

Los motivos principales fueron los siguientes:

- La limitación de la cuenta gratuita no nos suponía ningún problema ahora ni de cara al futuro.
- Fácil integración en la web y en Slack.
- Personalización del CSS.
- Soporte para castellano.
- *Rich responses*.
- Monitorización de las conversaciones mediante logs.
- Potente NLP de Google.

Teniendo todo esto en consideración comencé a realizar el proyecto bajo el nombre de UBUChatbot y cuyo logo se puede ver en la imagen 5.1. El símbolo utilizado en el logo es un oso blanco con un icono de conversación sobre un fondo granate. Decidí utilizar un oso ya que era algo diferente a las siempre utilizadas imágenes de robots o teleoperadoras. Además, este animal dota de un aspecto simpático a nuestro *chatbot*, que puede resultar agradable a los alumnos e incentivarles a interactuar con él. Los colores se han seleccionado en base a los que utiliza la plataforma UBUVirtual para así preservar la identidad corporativa de la Universidad.



Figura 5.1: Logo del proyecto.



## 5.2. Formación y metodologías

Antes de comenzar con la implementación seguí un proceso de formación para entender bien el funcionamiento y posibilidades de nuestras herramientas: Dialogflow y Moodle.

En el caso de Dialogflow consulté la documentación oficial suministrada por Google [8]. Para conocer el funcionamiento interno de Moodle y las posibilidades de introducción/modificación de código HTML, JavaScript y CSS opté por utilizar una versión local de Moodle en la que, por medio de prueba y error, adecuar la implementación a esta plataforma.

Por otra parte, era vital entender también el funcionamiento de la asignatura de Trabajo de Fin de Grado, con el fin de poder implementar correctamente las preguntas y respuestas. Esto precisó un trabajo colaborativo con mi tutor y personal docente de la Universidad para establecer un consenso en la información a introducir en el chatbot. Por ello aplicamos una metodología ágil: Scrum. Cabe matizar que no se aplicó estrictamente esta metodología al estar sujetos a varias restricciones: la mayor parte del trabajo era individual, las fechas estaban muy determinadas por temas externos al proyecto (por lo que no se podían fijar unos plazos estrictos) y tampoco podíamos mantener reuniones diarias.

Pese a todas estas limitaciones, en líneas generales sí que se aplicó esta filosofía en aspectos como los siguientes:

- Desarrollo incremental y revisiones durante y al final de cada desarrollo.
- En cada iteración se entregaron e implementaron nuevas versiones del producto.
- Planificación de tareas a realizar en el sprint.
- Sprints de 2 semanas. En épocas de menor disponibilidad por trabajo se aumentaron los plazos a 3 o 4 semanas.
- Control de versiones para todos los elementos del proyecto: base de datos de preguntas y respuestas, código software del programa y documentación de la memoria y anexos.

### 5.3. Implementación básica e integración en UBUVirtual

Comencé estudiando la documentación de FAQs sobre la asignatura disponible en UBUVirtual. A partir de ella, fui implementando y probando al mismo tiempo los correspondientes *intents* en Dialogflow. Una vez implementada toda la información disponible en el FAQ, continué añadiendo nuevas preguntas que pudieran no haber salido hasta la fecha. Para ello conté con la colaboración de amigos y compañeros que cursaban o habían cursado un Trabajo de Fin de Grado para que pensasen en dudas que en su momento les habían surgido.

Dado que una vez se integrase iba a estar disponible para el resto de alumnos, fueron necesarios varios meses de trabajo en la implementación de preguntas y respuestas; así se evita que las primeras impresiones de los alumnos fueran negativas y esto les produjese un rechazo a la hora de volver a utilizar el *chatbot*. En el momento de su integración en UBUVirtual el 12 de enero de 2021 contaba con más de 50 respuestas (*intents*) para más de 280 preguntas (*training phrases*).

La integración la realizamos de manera colaborativa entre tutor y alumno, siendo el primero quien tenía los permisos para añadir contenido HTML a la web. En este HTML se incluyó el código CSS para utilizar la misma combinación de colores granate y blanco que se utiliza en la plataforma. También se añadió el código JavaScript propio de Dialogflow junto con alguna personalización para adecuarlo a nuestro propósito. En la Figura 5.2 se muestra el resultado de dicha integración.

### 5.3. IMPLEMENTACIÓN BÁSICA E INTEGRACIÓN EN UBUVIRTUAL

31

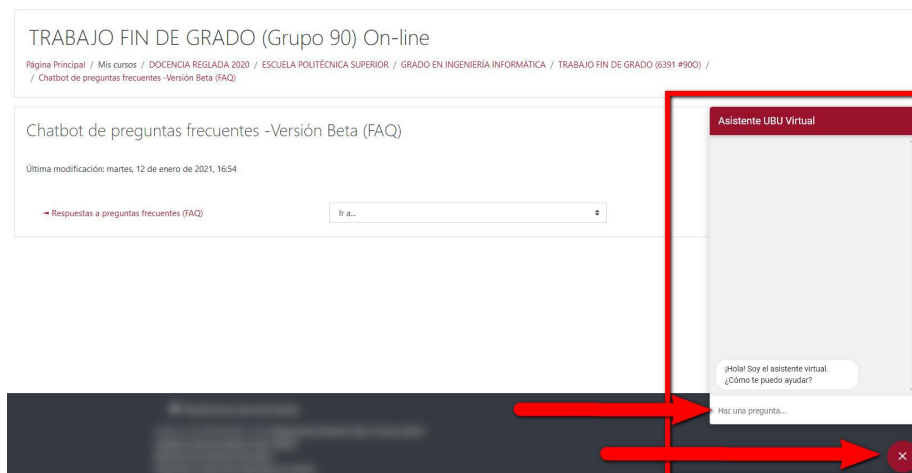


Figura 5.2: Integración en Moodle.

Una vez integrado (enero de 2021) el tutor informó mediante un mensaje en el foro a todos los alumnos y profesores con acceso a la asignatura de la incorporación de esta característica, con el fin de que lo utilizarasen y poder comenzar a recopilar información y entrenar al *chatbot* con la información obtenida de los registros de estas conversaciones.

Finalmente, en junio de 2021 modificamos el botón para abrir o cerrar el chatbot añadiéndole el icono del proyecto: el oso blanco sobre el fondo granate.

## 5.4. Estudio de características avanzadas e implementación en Slack

Una vez en funcionamiento la versión básica y en paralelo al entrenamiento manual del chatbot por medio de las nuevas preguntas que iban surgiendo en los logs, estudié e implemente nuevas características para dotarlo de resultados mas visuales.

Por medio de las *Rich Responses* de Dialogflow añadí formato a varias de las respuestas, fundamentalmente a aquellas que devolvían un enlace o lista; la implementación que se tenía únicamente devolvía texto plano y de esta manera se añaden hipervínculos en barras, descripciones y otras características. Estas respuestas se implementan en formato JSON indicando el tipo de respuesta, sus parámetros y valores.

De una manera parecida realicé la implementación de respuestas para la integración de Slack, aunque en este caso los JSON utilizan el lenguaje Markdown. En cada *intent* se asocia una respuesta para la versión web (texto plano o JSON) y otra para la versión de Slack.

Realicé la integración dentro del espacio de trabajo de Slack creado para este fin. Dialogflow dispone de integración nativa con Slack, por lo que fue un proceso rápido en el que únicamente seguí una serie de pasos: configurar Slack, crear una aplicación, agregar el usuario bot y finalmente habilitar y configurar la integración desde Dialogflow.

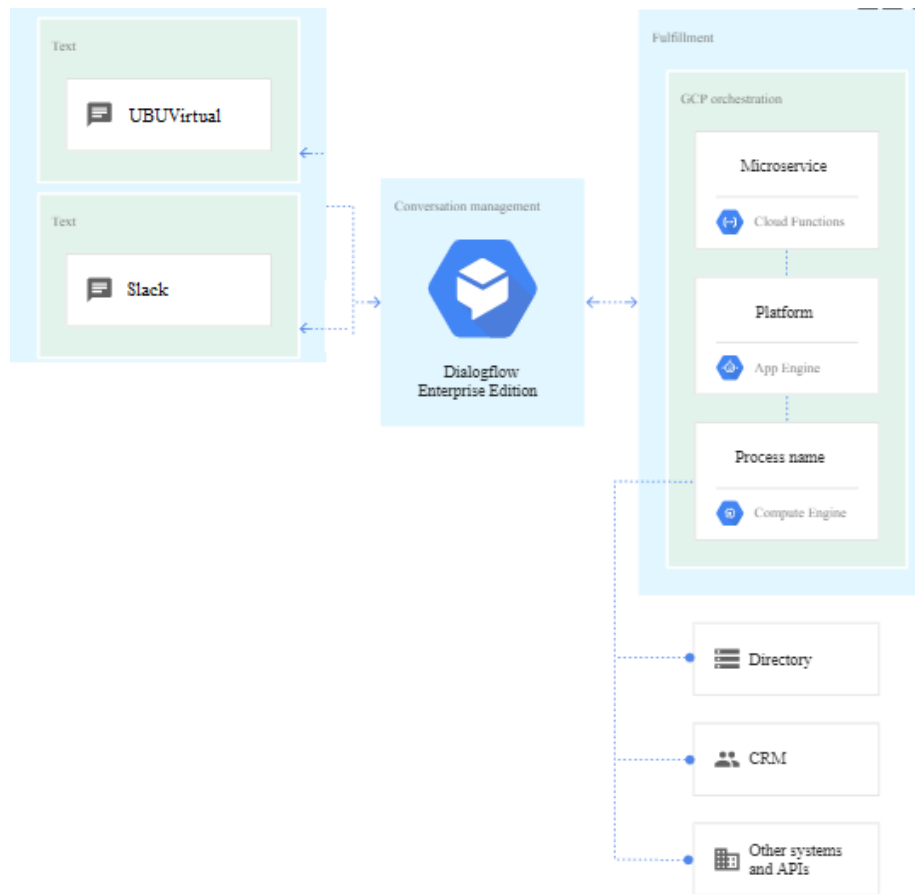


Figura 5.3: Modelo arquitectónico [21]

En la Figura 5.3 se muestra el modelo arquitectónico utilizado. Se muestra también la estructura que seguirían los *fulfillment* -característica analizada anteriormente- que no están integrados en el proyecto debido a que es necesario tener una cuenta de facturación asociada a la cuenta para poder utilizarlos.

## 5.5. Proceso de entrenamiento

El entrenamiento del *chatbot* fue un proceso continuo que llevé a cabo desde el inicio del proyecto hasta su entrega, siendo la principal tarea y fuerte del proyecto y en la que más horas de trabajo invertí.

Por una parte, Dialogflow realiza aprendizaje automático a partir de los logs y nuevas implementaciones, por medio del cual afina el algoritmo de asignación de *intent* a una entrada. El algoritmo utilizado tanto para el entrenamiento como para la asignación son propios de Google y no son de código abierto, por lo que no hay disponible información de su implementación.

Por otra parte, el entrenamiento manual consistió en el estudio de todos los logs para analizar en qué ocasiones el algoritmo de asignación fallaba, cuándo se respondían realmente las dudas y cuándo no era capaz de entender lo que le estaban diciendo, para posteriormente arreglarlo o implementarlo.

Por ejemplo, si un usuario preguntaba las fechas de entrega y nuestro *chatbot* detectaba el *intent* en el que se enlaza al calendario de entregas lo contabilizamos como un acierto. Si en cambio hubiese detectado un *intent* relacionado con otro aspecto de la entrega (por ejemplo el material a entregar) lo contabilizamos como incorrecto. Finalmente, si por ejemplo el usuario decide preguntarnos por el tiempo que hace ese día lo contabilizamos como una pregunta inválida.

## 5.6. Análisis de resultados

Para comprobar la evolución del *chatbot* y comprobar si su entrenamiento estaba surgiendo efecto era necesario realizar un estudio estadístico del éxito de las conversaciones con personas ajenas al proyecto desde que resultase accesible por medio de la integración en UBUVirtual.

Dialogflow en su sección de *Analytics* ofrece una serie de análisis de las conversaciones, no obstante estas no sirven para el análisis de resultados por los siguientes motivos:

- Rango máximo permitido de un mes; es decir, no muestra las estadísticas conjuntas de todo este periodo.
- Al no distinguir los usuarios que interactúan, toma también las conversaciones llevadas a cabo por el desarrollador, que en muchas ocasiones son solo conversaciones de prueba y desvían el resultado real.
- Solo es capaz de distinguir si el algoritmo ha sabido asignar un *intent* a la pregunta. No es capaz de discernir si la respuesta que ha dado realmente responde a la pregunta al no saber analizar el lenguaje natural como solo puede hacer un humano.

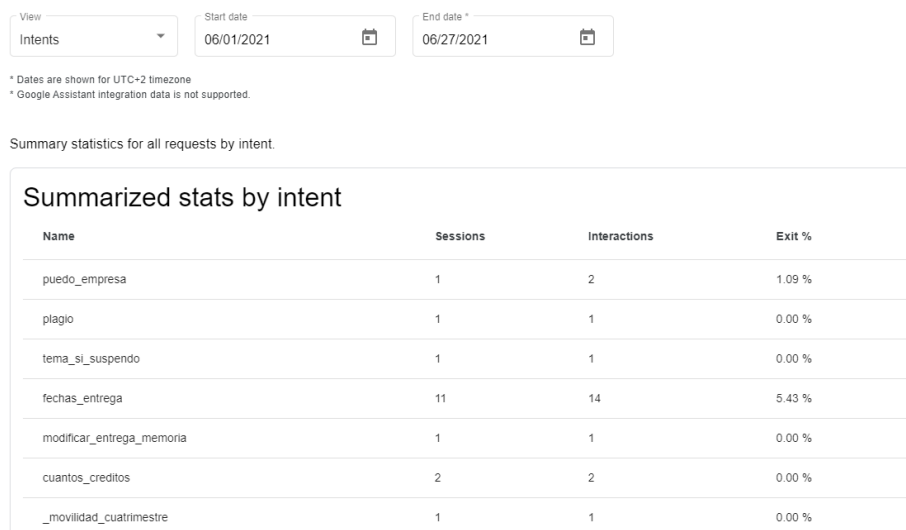


Figura 5.4: Dialogflow Analytics: detección de intents.

Por todos esos motivos realicé un estudio numérico de forma manual en el que analicé una por una cada conversación, determinando en qué ocasiones la respuesta que se daba realmente respondía a la pregunta y descartando también aquellas preguntas mal formuladas, incoherentes o realizadas por personas implicadas en el proyecto.

Realicé este estudio tanto para la versión online como la presencial, combinándolos finalmente en la tabla de totales mostrada en la Figura 5.5.

Mes	Correctas	Incorrectas	Inválidas	Total	Tasa acierto
Enero	29	72	16	101	28,71%
Febrero	3	6	2	9	33,33%
Marzo	19	20	5	39	48,72%
Abril	2	2	0	4	50,00%
Mayo	11	9	3	20	55,00%
Junio	31	25	8	56	55,36%

Figura 5.5: Totales extraídos del análisis numérico.

En la tabla de totales se puede ver como la actividad varió mucho en los distintos meses. Esto es debido a que los meses con mayor actividad son aquellos en los que se incentivó por medio del foro o por otros canales de comunicación a personas ajenas al proyecto a que lo utilizasen, con el fin de aumentar el número de *logs* disponibles para poder continuar entrenándolo.

En la imagen 5.6 se muestra la evolución temporal de la tasa de éxito de respuesta.



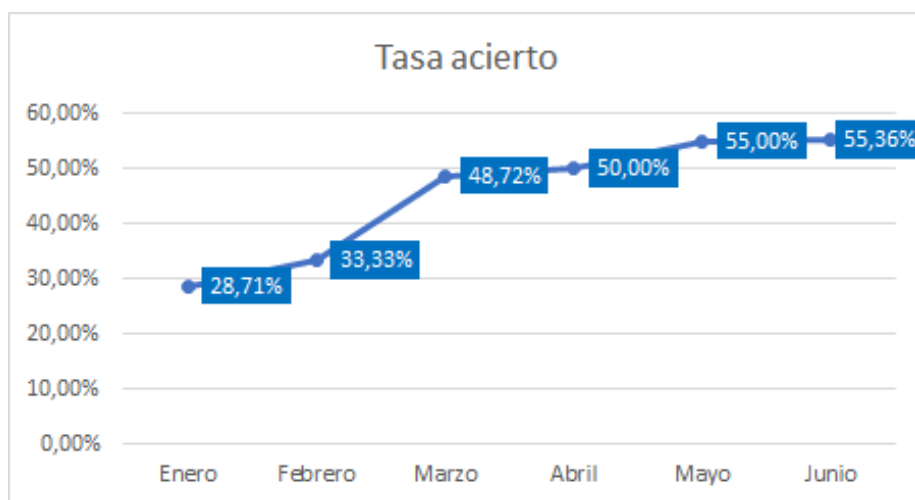


Figura 5.6: Evolución temporal de la tasa de acierto.

Como podemos ver, el proyecto ha evolucionado correctamente, aumentando su tasa de éxito de un 28,7 % a 55,3 %, casi el doble. Para seguir esta línea ascendente serán necesarios nuevos logs a partir de los cuales seguir aumentando nuestra base de datos de preguntas y frases de entrenamiento. Ayudaría darle mayor visibilidad al *chabot*, colocándolo en la página principal de la asignatura en lugar de la subsección en la que está habilitado.

Podemos concluir que el proyecto ha sido un éxito, dado que obtener una tasa de acierto de más del 50 % -pese a haber dispuesto de una cantidad de *logs* bastante reducida- es más de lo que se podía esperar en un principio. Además, hemos aportado valor a la Universidad resolviendo muchas de las preguntas que les han surgido a los alumnos.



---

## Trabajos relacionados

---

El primer bot conversacional fue creado entre 1964 y 1966 por Joseph Weizenbaum en el Laboratorio de Ciencias de la Computación e Inteligencia Artificial del Instituto de Tecnología de Massachusetts [22]. Simulaba ser un psicólogo y se creó con el propósito de demostrar la superficialidad de las conversaciones entre humanos y máquinas.

Desde entonces han evolucionado, especialmente en los últimos años y aún más tras la pandemia de Covid-19, reflejo de ello es un estudio de *Markets And Markets* [23] que apunta un incremento previsto en el volumen de negocio de los chatbots de 4.8 billones de dólares en 2020 a 13.9 billones en 2025.

Las finalidades de los chatbots son muy variadas: responder preguntas frecuentes, ofrecer soporte al cliente, encuestas, reservas, etc. En este apartado vamos a analizar algunos trabajos relacionados. Al tratarse de un campo tan amplio vamos a focalizar la búsqueda con una serie de criterios. El primer criterio es que únicamente vamos a tener en cuenta los chatbots programados en castellano, ya que el idioma es un aspecto relevante al ser en torno a lo que giran los NLP. El segundo criterio va a ser el del fin; solo vamos a buscar chatbots implementados en universidades españolas con el fin de responder dudas frecuentes de estudiantes, siendo este el mismo propósito que el de nuestro proyecto.

No obstante, en primer lugar vamos a analizar los proyectos de la Universidad de Burgos que tengan alguna relación con los chatbots, para tener una visión de la situación actual de la UBU en esta tecnología.

## 6.1. Proyectos de la Universidad de Burgos

### *UBUAssistant*



Figura 6.1: UBUAssistant logo

Proyecto de Daniel Santidrián Alonso continuado por Carlos González Calatrava en 2018 tutorizado por Pedro Renedo Fernández. Se trata de un asistente virtual para Android que cuenta con asistente de voz. El objetivo es facilitar las búsquedas en la web de la UBU al realizarlas desde un smartphone. La aplicación está alojada en un servidor de Azure que ofrece una suscripción gratuita para estudiantes, aunque con ciertas limitaciones. Utiliza el algoritmo de Razonamiento Basado en Casos. [24]

### *Chatbot for Tourist Recommendations*



Figura 6.2: Tourist Bot logo

Proyecto de Jasmin Wellnitz en 2017 tutorizado por el Dr. Bruno Baroque Zanon. Chatbot implementado en Telegram actualmente fuera de servicio cuya finalidad era dar recomendaciones turísticas en inglés. Utilizó la misma tecnología que este proyecto, en aquel entonces se llamaba API.AI y era el mismo NLP que es ahora Dialogflow tras ser comprado en 2016 por Google que optó al año siguiente por cambiarle el nombre. El proyecto esta alojado en la *Platform as a Service* Heroku. Utiliza la versión de prueba gratuita pero tiene limitaciones en cuanto al uso mensual. [25]

***UBUVoiceAssistant***

Figura 6.3: UBUVoiceAssistant logo

Proyecto de Álvaro Delgado Pascual tutorizado por el Dr. Raúl Marticorena Sanchez. Se trata de una aplicación que por medio de un asistente de voz permite al usuario obtener información sobre una plataforma de Moodle. Está desarrollado en Python y utiliza el asistente de voz Mycroft. [26]

## 6.2. Proyectos externos

### *Chatbot Lola*

*Lola* es un chatbot de la Universidad de Murcia creado en 2018 cuya finalidad es la de resolver dudas a nuevos estudiantes acerca del proceso de admisión y matrícula.

Lola es un proyecto del Servicio de Información Universitario (SIU) que se ha desarrollado con el Área de Tecnologías de la Información y de las Comunicaciones Aplicadas de la Universidad de Murcia (ATICA) y la empresa 1 million bot. [27]



Figura 6.4: Logo y funcionalidades de Lola

Este mismo chatbot está implementado por la misma empresa en otras universidades: Universidad de Zaragoza, Complutense de Madrid y Universidad Politécnica de Valencia entre otras. Todos tienen el mismo funcionamiento aunque utilizan nombres comerciales distintos, siendo siempre un nombre de mujer.

Proyecto	UBUChatbot	UBUAssistant	Tourist Chatbot	UBUVoice	Lola
Gratuito	X	X	X	X	X
Chatbot QA	X				X
Integración Web	X	X		X	X
Integración Slack	X				
Castellano	X	X		X	X

Tabla 6.1: Plataformas de desarrollo de Chatbots.

### 6.3. Tabla comparativa del proyecto

En la tabla 6.1 se muestra una comparativa de las características necesarias para este proyecto que aportan cada uno de los proyectos analizados.





---

# Conclusiones y Líneas de trabajo futuras

---

En este apartado se reflejan las conclusiones sacadas de la realización del proyecto así como una serie de líneas de trabajo extraídas a partir de ellas y que pueden servir como directrices para futuras mejoras y continuidad de la explotación del chatbot.

## 7.1. Conclusiones

- El objetivo del proyecto se ha cumplido, el chatbot resuelve con éxito más del 50 % de las preguntas planteadas por los alumnos, reduciendo con esto de una manera notable el número de consultas realizadas por otros medios al personal docente de la asignatura. Además, no se ha incurrido en ningún coste ni introducido ninguna cuenta de facturación en ninguna de las herramientas ni plataformas utilizadas.
- La integración con una plataforma distinta al portal de UBUVirtual se ha realizado correctamente. La plataforma elegida ha sido Slack, aunque también se podía haber realizado con Whatsapp (aunque no de manera gratuita), Facebook Messenger o Telegram, entre otras. Esta integración por si misma no aporta valor al proyecto por el momento, pero en futuras versiones del chatbot que implementen nuevas funcionalidades sí podría ser de utilidad.
- Dialogflow ha resultado ser una solución óptima para este proyecto: un NLP potente, gratuita y con una curva de aprendizaje muy pequeña que facilitará la continuidad del chatbot en manos de futuros

estudiantes.

- El éxito de la conversación depende en gran medida del correcto uso por parte del estudiante, si sabe formular las preguntas de manera precisa y concisa va a obtener muchos mejores resultados que alguien que formula largas preguntas con un lenguaje similar al que utilizaría si estuviese hablando con un humano. Por esta parte, al tratarse los usuarios de estudiantes de Ingeniería Informática con ciertos conocimientos sobre la limitación de la Inteligencia Artificial, en la mayoría de casos no se da este problema.  
Además, con la adición de indicaciones de cómo formular las preguntas por parte de nuestro chatbot en caso de mensajes no reconocidos, se ha mejorado en este aspecto.
- Los chatbots son una herramienta en auge, los NLP son cada vez más potentes y sin duda es una tecnología que ha venido para quedarse. Cada vez son más las webs que cuentan con un chatbot y la Universidad de Burgos no puede quedarse atrás en este aspecto.

## 7.2. Líneas de trabajo futuras

- Integración del chatbot en la página principal de la asignatura. Con esto se obtendrá una mayor visibilidad por parte de los potenciales usuarios, lo que elevará el número de interacciones, reduciendo el número de preguntas que realicen a los profesores. Además, esto nos proporcionará mayor número de *logs*.
- Actualización y mejora de nuestro NLP con nuevas preguntas y frases de entrenamiento obtenidas a partir de los nuevos *logs* recibidos. El objetivo debe ser continuar mejorando la tasa de éxito de respuesta.
- Añadir una encuesta al finalizar la conversación en la que se reciba feedback por parte del usuario. Se implementará en forma de preguntas sobre la satisfacción y sugerencias de mejora. Con esto se obtendrá información más objetiva sobre el grado de satisfacción y mejoras deseadas por parte de nuestros usuarios.
- Actualización a proyecto con cuenta de pago asociada que permita la implementación de código para nuestros *webhooks*. Implementar por código un control de *fallbacks* que permita redirigir la conversación a otro medio una vez se produzcan errores como tres mensajes seguidos sin contestación o la entrada reiterada a un mismo *intent*.
- Estudiar el posible interés en realizar consultas por código a fuentes externas.
- Analizador léxico para extraer información que permita reconocer patrones de comportamiento.
- Agente multilingüe para responder preguntas a estudiantes Erasmus o de otros programas de intercambio.
- Implementación de chatbot para la página web principal de la Universidad de Burgos y otras asignaturas.
- Estudiar los posibles beneficios de continuar la implementación en Slack y/o otras plataformas distintas a UBUVirtual.



---

## Bibliografía

---

- [1] P. Teesh, “Family’s of natural language,” 2019. [Online; Accedido 20-Mayo-2021].
- [2] S. Intents, “Using dialogflow rich responses,” 2020. [Online; Accedido 15-Junio-2021].
- [3] Dialogflow, “Fulfillment,” 2021. [Online; Accedido 15-Junio-2021].
- [4] A. A. Vázquez, “Repositorio ubuchatbot,” 2021. [Online; Accedido 23-junio-2021].
- [5] Wikipedia, “Eugene goostman — Wikipedia, The Free Encyclopedia,” 2021. [Online; Accedido 06-Mayo-2021].
- [6] M. G. Institute, “Eugene goostman — Wikipedia, The Free Encyclopedia,” 2017. [Online; Accedido 20-Mayo-2021].
- [7] H. Shift, “What is intent in a chatbot?,” 2020. [Online; Accedido 15-Junio-2021].
- [8] Dialogflow, “Dialogflow es documentation,” 2021. [Online; Accedido 15-Junio-2021].
- [9] S. Pulse, “How to send webhooks via chatbots,” 2021. [Online; Accedido 15-Junio-2021].
- [10] Rasa, “Rasa página oficial,” 2021. [Online; Accedido 07-Mayo-2021].
- [11] J. Tenorio, “Construyendo un chatbot en español usando rasa,” 2021. [Internet; descargado 2-abril-2021].

- [12] A. Lex, “Amazo lex página oficial,” 2021. [Online; Accedido 07-Mayo-2021].
- [13] S. Smiljanic, “Amazon alexa statistics, facts, and trends,” 2021. [Internet; descargado 2-abril-2021].
- [14] W. Assistant, “Watson assistant página oficial,” 2021. [Online; Accedido 07-Mayo-2021].
- [15] C. Compose, “Chat compose página oficial,” 2021. [Online; Accedido 07-Mayo-2021].
- [16] M. B. Framework, “Microsoft bot framework página oficial,” 2021. [Online; Accedido 07-Mayo-2021].
- [17] C. Fuel, “Chat fuel página oficial,” 2021. [Online; Accedido 07-Mayo-2021].
- [18] Wit.Ai, “Wit.ai página oficial,” 2021. [Online; Accedido 07-Mayo-2021].
- [19] F. XO, “Flow xo página oficial,” 2021. [Online; Accedido 07-Mayo-2021].
- [20] D. Docs, “Dialogflow editions,” 2021. [Online; Accedido 27-junio-2021].
- [21] P. Vergadia, “Deconstructing chatbots: Getting started with dialogflow,” 2010. [Online; Accedido 27-junio-2021].
- [22] Wikipedia, “Eliza — Wikipedia, The Free Encyclopedia,” 2021. [Online; Accedido 20-Mayo-2021].
- [23] Markets and Markets, “Conversational ai market,” 2021. [Online; Accedido 17-junio-2021].
- [24] D. S. Alonso, “Ubuassistant,” 2018. [Online; Accedido 19-junio-2021].
- [25] J. Wellnitz, “Development of a chatbot for tourist recommendations,” 2017. [Online; Accedido 19-junio-2021].
- [26] Álvaro Delgado Pascual, “Ubuvoiceassistant,” 2020. [Online; Accedido 19-junio-2021].
- [27] U. de Murcia, “La universidad de murcia presenta a lola, un asistente de inteligencia artificial para ayudar a los nuevos alumnos,” 2018. [Online; Accedido 14-Junio-2021].