

Universidad de Sevilla
Escuela Técnica Superior de Ingeniería Informática

CHATBOT POR ÁRBOLES DE DECISIÓN
(DOCUMENTACIÓN DEL PROYECTO)



Grado en Ingeniería Informática – Ingeniería del Software
Complemento de Base de Datos (CBD)
Curso 2019 – 2020

Fecha	Versión
29/04/2020	1.0

Miembros	Roles
Iván López Hernández	Desarrollador y documentador
Joaquín García Macías	Desarrollador y documentador

Contenido

1. INTRODUCCIÓN	2
2. OBJETIVO	2
3. CONTENIDOS DEL PROYECTO	2
3.1. Árboles de decisión en el ChatBot	¡Error! Marcador no definido.
3.2. TECNOLOGÍAS USADAS	4
3.3. IMPLEMENTACIÓN EN DIALOGFLOW	¡Error! Marcador no definido.
4. CONCLUSIONES	11

1. INTRODUCCIÓN

Este trabajo se ha realizado para la asignatura Complemento de Base de Datos (CBD), en la cual hemos visto distintas bases de datos con sus tecnologías.

Para este trabajo se ha utilizado la herramienta Dialogflow, ya que queríamos hacer un ChatBot por árboles de decisión en Telegram y esta herramienta nos permitía llevarlo a cabo con relativa facilidad.

2. OBJETIVO

El objetivo principal de nuestro ChatBot es poder informar fácilmente, a través de la herramienta Telegram, a los usuarios que quieran utilizarlo sobre la situación actual del coronavirus o Covid-19.

Nuestro ChatBot podrá responder fácilmente sobre el Covid-19, ya que es un virus muy nuevo, el cual se está dando a escala mundial, y la gente tiene muchas dudas sobre el virus, la situación en la que estamos, las formas de prever el contagio y más información útil para adaptarse a la situación actual.

3. CONTENIDOS DEL PROYECTO

En este apartado redactaremos toda la información que creamos conveniente para la creación de este proyecto.

3.1. ÁRBOLES DE DECISIÓN EN CHATBOT

Lo primero será dar una pequeña explicación de qué son los árboles de decisión y aplicaremos esta explicación a nuestro ChatBot.

Un árbol de decisión es un modelo de predicción utilizado en diversos ámbitos. Dado un conjunto de datos se fabrican diagramas de construcciones lógicas, muy similares a los sistemas de predicción basados en reglas, que sirven para representar y categorizar una serie de condiciones que ocurren de forma sucesiva, para la resolución de un problema.

Dado un conjunto de datos, se fabrican diagramas de construcciones lógicas que sirven para representar y categorizar una serie de condiciones que ocurren de forma sucesiva, para la resolución de un problema.

Para este proyecto, el ChatBot funcionará de la siguiente forma:

- Nada mas iniciarlo por primera vez, o tras escribir el comando /start, el bot nos mostrará las diferentes categorías disponibles, para que el usuario pueda elegir la principal fuente de información de la cual desea obtener respuestas a través de la conversación con el agente conversacional.
- Una vez elegida una categoría, tras escribirlo en el chat, el usuario podrá realizar preguntas sobre el área seleccionada. Además, el ChatBot, ofrecerá sugerencias al usuario de posibles preguntas que realizar de modo que sea intuitivo y sirva de ayuda para navegar por las diferentes fuentes de contenido que posee.
- En caso de que no se encuentre una respuesta para la pregunta asociada, o de que el ChatBot no sea capaz de entender y/o categorizar lo que el usuario introduzca, se le mostrará un mensaje, donde se indicará que se tendrá en cuenta la pregunta para seguir mejorando, y se le instará al usuario a seguir preguntando otras cosas sobre el tema, o a buscar en una categoría diferente.

Dentro de un árbol de decisión, se encuentran diferentes elementos, que son necesarios para el correcto entendimiento de los mismos, estos son:

- **Nodos:** Momento en el que hay que tomar una decisión entre varias posibilidades. En nuestro bot se podría aplicar a las categorías sobre las que puedes elegir para informarte del coronavirus.
- **Vectores de números:** solución final a la que se llega en función de las diversas posibilidades que se tienen. En el caso de nuestro bot serían las distintas preguntas que puede responder el bot sobre la categoría en la que se entre a partir de la pregunta del nodo inicial.
- **Flechas:** Uniones entre un nodo y otro y representan cada acción distinta que se ha escogido voluntariamente. Para nuestro bot equivaldría a la pregunta que se formula una vez que se llega a un nodo para avanzar al siguiente nodo o a un resultado final.
- **Etiquetas:** Se encuentran en cada nodo y cada flecha y dan nombre a cada acción. De este elemento se podrá ver una representación más visual a continuación.

En la siguiente figura, se ilustra la estructura general del Árbol de decisión implementado para el ChatBot:

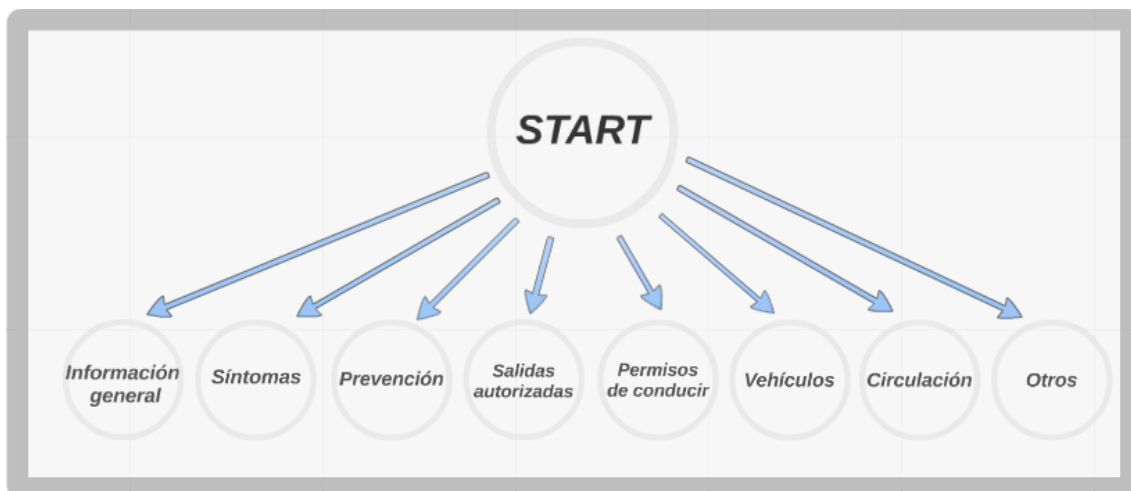


Fig 1. Ilustración del árbol de decisión del chatbot

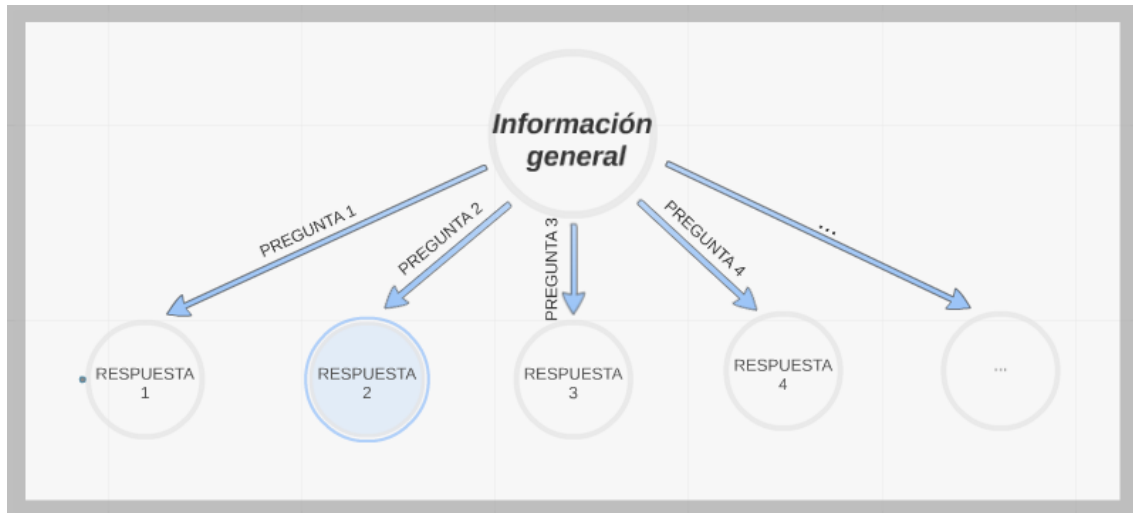


Fig 2. Ilustración del árbol de decisión del chatbot y sus preguntas y respuestas

Dentro de los árboles de decisión, se dan ciertas reglas que se deben cumplir siempre:

- Existe un único nodo inicial que no es apuntado por ninguna flecha.
- El resto de los nodos si son apuntados por una única flecha.
- Hay un único camino para llegar del nodo inicial a cada uno de los distintos nodos. No hay varias formas de llegar a la misma solución final, las decisiones son excluyentes.

Se podría dar el caso en que una persona haga una pregunta y el bot no sepa responderla, debido a que esa forma de preguntarla o ese tema puede que no esté recogido en la herramienta Dialogflow.

Para dar solución a esto, aparte de mostrar un pequeño mensaje al usuario que ha realizado la pregunta indicándole que esa pregunta podrá ser añadida en futuras versiones, definiremos distintas versiones de las preguntas en Dialogflow, así como sinónimos en distintas palabras que veamos que sean necesarias.

3.2. TECNOLOGÍAS USADAS

Para la realización de este trabajo, se procedió a la investigación de diferentes tecnologías de mercado y cual tendría un mayor impacto en los usuarios a la hora del despliegue y uso.

Tras investigar tecnologías como Watson, para la realización de los árboles de decisión, y otras interfaces, como slack, nos decantamos por el uso de Telegram como forma de interacción y Dialogflow, como maquinaria, por su versatilidad y compatibilidad.

3.2.1 DIALOGFLOW

Se trata de una herramienta de creación de ChatBots de Google capaz de entender el lenguaje natural y proveer infraestructura para recrear conversaciones y construir diálogos con el fin de interactuar con el usuario de manera fluida.



Dentro de DialogFlow, es necesario definir ciertos aspectos para entender el funcionamiento de los Árboles de Decisión:

INTENTS

Un intent clasifica la intención del usuario final para un turno de conversación. Se pueden definir varios intents para cada agente, de manera que, en conjunto, estos intents permitan manejar una conversación completa. Cuando un usuario final escribe o dice algo (lo que se denomina *expresión de usuario final*), Dialogflow trata de identificar el intent más adecuado de los agentes para esa expresión del usuario final.

La búsqueda de intents coincidentes también se conoce como *clasificación de intent*.

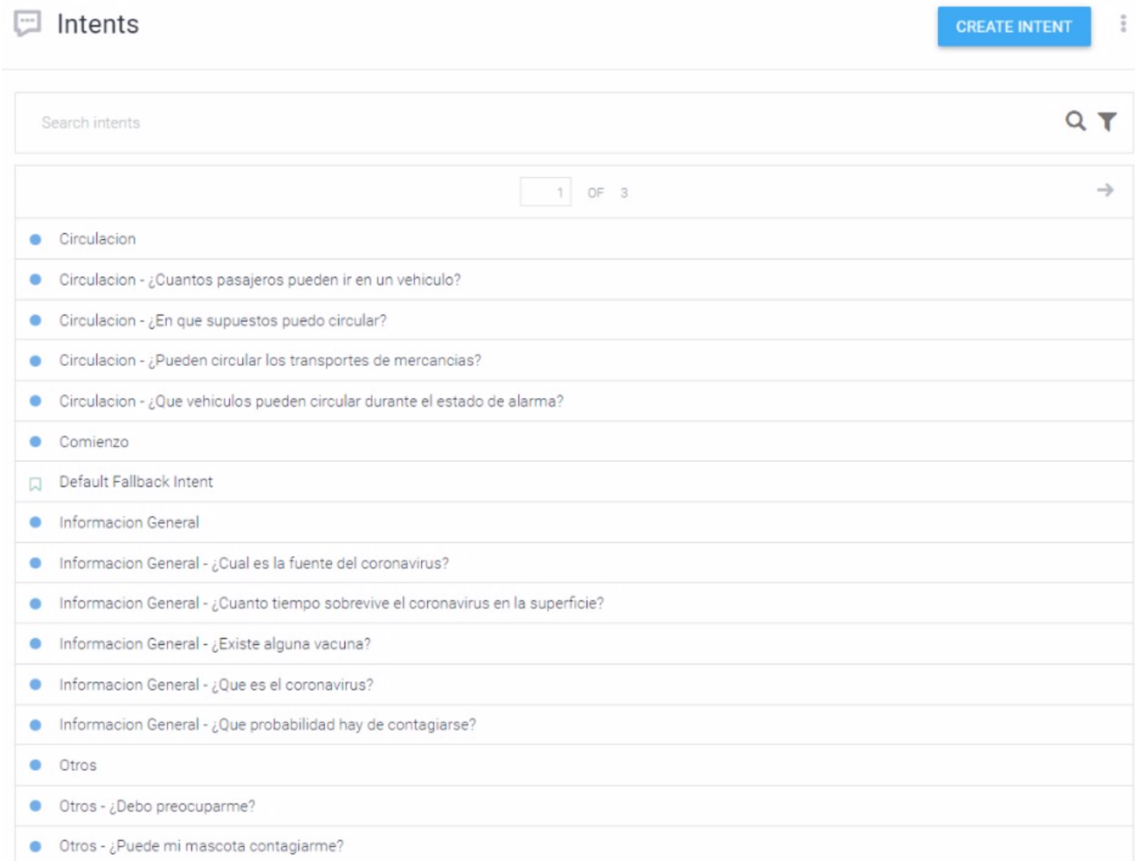


Fig 3. Intents en DialogFlow

Por ejemplo, podrías crear un agente meteorológico que reconozca preguntas de los usuarios finales sobre el clima y las responda. Podrías definir un intent para las preguntas sobre el pronóstico del tiempo. Si un usuario final dice “¿Cuál es el pronóstico?”, Dialogflow determinaría que esa expresión del usuario final corresponde al intent del pronóstico. También puedes definir tu intent con el fin de extraer información útil de la expresión del usuario final, como una hora o ubicación para el pronóstico del tiempo deseado. Estos datos extraídos son importantes a fin de que tu sistema realice una consulta sobre el clima para el usuario final.

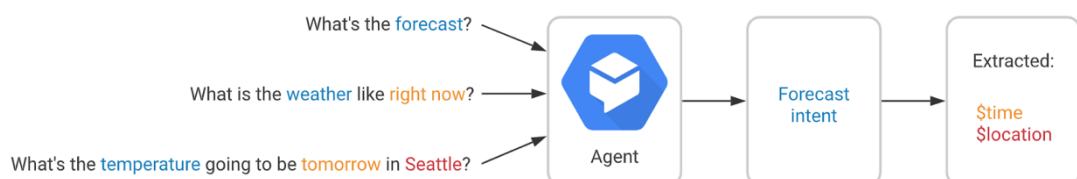


Fig 4. Funcionamiento de un agente

En un intent básico, existen los siguientes elementos:

- **Frases de Entrenamiento:** Estas son frases de ejemplo de algo que podrían decir los usuarios finales. Cuando una expresión de usuario final se parece a una de estas frases, Dialogflow identifica la coincidencia con el intent.
- **Acción:** Cuando se identifica una coincidencia con un intent, Dialogflow proporciona la acción al sistema, lo que se puede usar para lanzar diferentes acciones en el sistema.

• **Información General - ¿Cual es la fuente del coronavirus?** ⓘ SAVE ⋮

Contexts ⓘ ⌵

Events ⓘ ⌵

Training phrases ⓘ Search training phrases 🔍 ⌵

” Add user expression

” ¿De donde sale el coronavirus?

” Origen del coronavirus

” ¿De donde viene el coronavirus?

” ¿Cuál es la fuente del coronavirus?

” ¿Dónde nació el coronavirus?

Action and parameters ⌵

Enter action name

REQUIRED ⓘ	PARAMETER NAME ⓘ	ENTITY ⓘ	VALUE	IS LIST ⓘ
<input type="checkbox"/>	coronavirus	@Coronavirus	\$coronavirus	<input type="checkbox"/>

Fig 5. Frases de entrenamiento y Acciones

- **Parámetros:** Cuando se identifica una coincidencia con un intent en el entorno de ejecución, Dialogflow proporciona los valores extraídos de la expresión del usuario final en forma de *parámetros*.
- **Respuestas:** Define las respuestas de texto, de voz o visuales que se le mostrarán al usuario final. Estas pueden responder una pregunta del usuario final, solicitarle más información o finalizar la conversación.

Nosotros, realizaremos las respuestas a través de código, para poder tener una mayor complejidad a la hora de responder, en lo que veremos más adelante como “fullfilment”, para ello, es importante activar dentro del intent la opción “Enable webhook call for this intent”, que nos permitirá realizar las respuestas a través del fullfillment, ignorando las respuestas incluidas dentro del intent.

• Informacion General - ¿Cual es la fuente del coronavirus? ⓘ SAVE ⋮

REQUIRED	PARAMETER NAME	ENTITY	VALUE	
<input type="checkbox"/>	coronavirus	@Coronavirus	\$coronavirus	<input type="checkbox"/>
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>

[+ New parameter](#)

Responses ? ^

DEFAULT TELEGRAM +

Text Response 🗑

1 Enter a text response ⌵

ADD RESPONSES

☐ Set this intent as end of conversation ?

Fulfillment ? ^

☒ Enable webhook call for this intent

☐ Enable webhook call for slot filling

Fig 6. Fullfilment de un intent

FULLFILMENT

La consola de Dialogflow tiene un editor de código integrado llamado fullfilment, que se puede usar para crear un código de cumplimiento e implementarlo en Cloud Functions.

El fullfilment solo es compatible con Node.js y usa la biblioteca de entregas de DialogFlow.

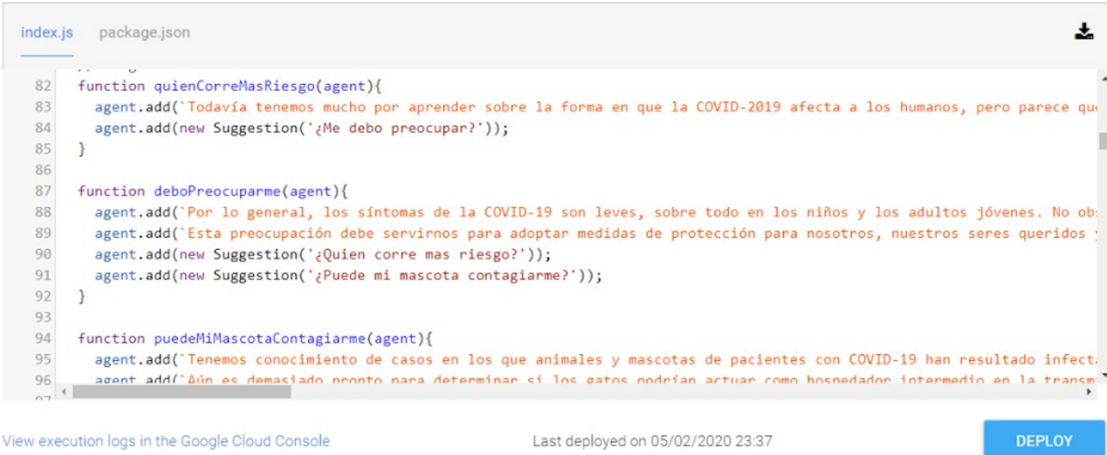
Utilizando este editor de texto, somos capaces de crear acciones más complejas, añadiendo funciones como las que vemos a continuación para poder realizar un ChatBot de precisión y con cierta complejidad.

Para poder hacer un correcto uso del fullfilment, deberemos activar el toggle “Enabled”, como se muestra a continuación.

Inline Editor (Powered by Google Cloud Functions)

ENABLED 

Build and manage fulfillment directly in Dialogflow via Cloud Functions. [Docs](#)



```

index.js  package.json
82  function quienCorreMasRiesgo(agent){
83      agent.add('Todavía tenemos mucho por aprender sobre la forma en que la COVID-2019 afecta a los humanos, pero parece que
84      agent.add(new Suggestion('¿Me debo preocupar?'));
85  }
86
87  function deboPreocuparme(agent){
88      agent.add('Por lo general, los síntomas de la COVID-19 son leves, sobre todo en los niños y los adultos jóvenes. No ob
89      agent.add('Esta preocupación debe servirnos para adoptar medidas de protección para nosotros, nuestros seres queridos :
90      agent.add(new Suggestion('¿Quién corre mas riesgo?'));
91      agent.add(new Suggestion('¿Puede mi mascota contagiarme?'));
92  }
93
94  function puedeMiMascotaContagiarme(agent){
95      agent.add('Tenemos conocimiento de casos en los que animales y mascotas de pacientes con COVID-19 han resultado infect
96      agent.add('¿Aún es demasiado pronto para determinar si los gatos podrían actuar como hospedador intermedio en la transm
97
View execution logs in the Google Cloud Console
Last deployed on 05/02/2020 23:37
DEPLOY

```

Fig 7. Ilustración de las funciones en el fulfillment

Con esto, podremos realizar funciones más complejas mediante diferentes métodos, como “add”, utilizado para añadir las respuestas del agente, o “Suggestions”, que nos permitirá proporcionar sugerencias al usuario.

Finalmente, podemos enlazar todas las funciones desarrolladas a los intents ya creados, de modo que cuando el ChatBot lance uno de ellos, la función definida se ejecutará.

Inline Editor (Powered by Google Cloud Functions)

ENABLED 

Build and manage fulfillment directly in Dialogflow via Cloud Functions. [Docs](#)



```

index.js  package.json
410
411  //Categorías
412  intentMap.set('Salidas Autorizadas', salidas);
413  intentMap.set('Otros', otros);
414  intentMap.set('Prevencion', prevencion);
415  intentMap.set('Síntomas', síntomas);
416  intentMap.set('Informacion General', general);
417  intentMap.set('Permisos de conducir', permisosDeConducir);
418  intentMap.set('Vehiculos', vehiculos);
419  intentMap.set('Circulacion', circulacion);
420
421  //Categoría Otros
422  intentMap.set('Otros - ¿Quién corre mas riesgo?', quienCorreMasRiesgo);
423  intentMap.set('Otros - ¿Debo preocuparme?', deboPreocuparme);
424  intentMap.set('Otros - ¿Puede mi mascota contagiarme?', puedeMiMascotaContagiarme);
425
View execution logs in the Google Cloud Console
Last deployed on 05/02/2020 23:37
DEPLOY

```

Fig 8. Ilustración de la relación de las funciones y sus intents

ENTITIES

Cada parámetro de intent pertenece a un tipo, el cual se denomina *tipo de entidad*. Esto determina de forma exacta cómo se extraen los datos de una expresión de usuario final.

Dialogflow proporciona entidades del sistema predefinidas que pueden detectar coincidencias con muchos tipos comunes de datos. Por ejemplo, hay entidades del sistema que detectan coincidencias con fechas, horas, colores, direcciones de correo electrónico, etcétera. También puedes crear tus propias entidades personalizadas para detectar coincidencias en datos personalizados.

En el caso de la aplicación, por ejemplo, se ha utilizado para definir las diferentes formas para llamar al Covid-19, por parte del usuario, así como de referirse a las mascotas, de modo que al realizar una conversación con el agente en Telegram, este pueda detectar los diferentes sinónimos y formas de expresión para referirse a los términos especificados, como se muestra a continuación:

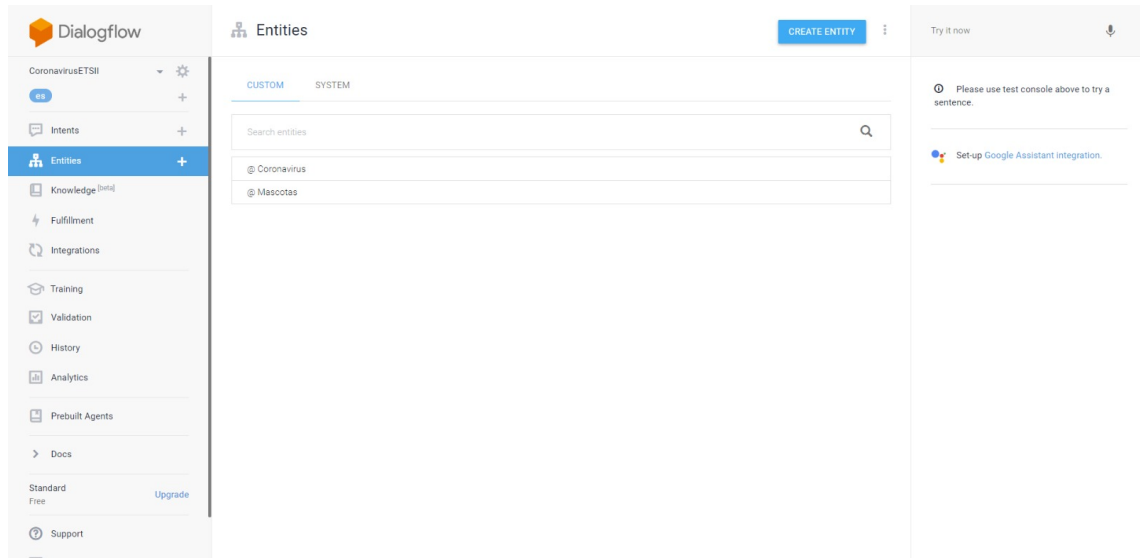


Fig 9. Entities en Dialogflow

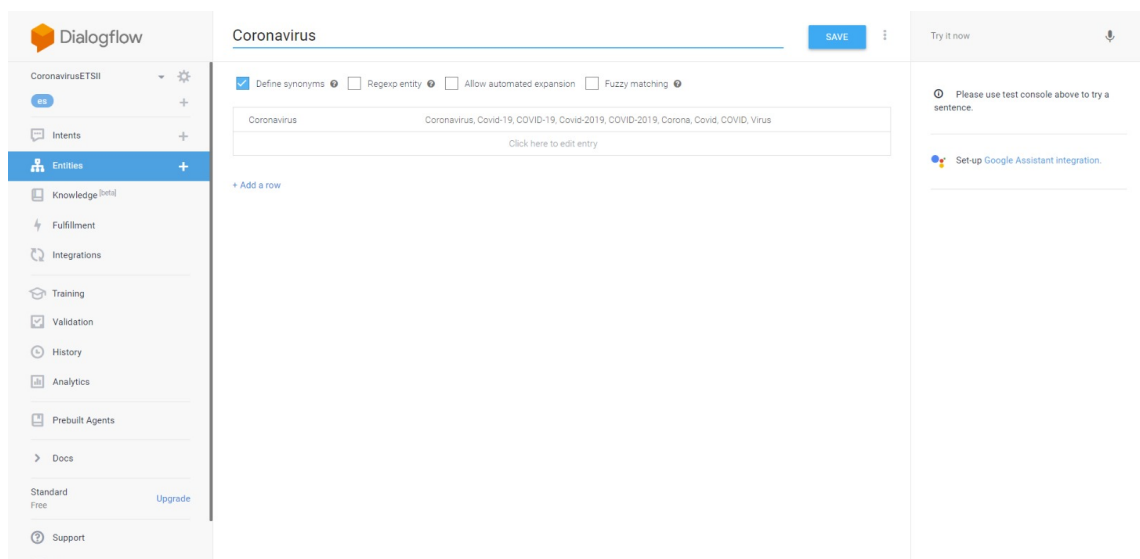


Fig 10. Definición de una entidad

3.2.2 TELEGRAM

Telegram es una aplicación de mensajería que servirá como interfaz de cara al usuario final, donde podrá realizar las diferentes interacciones con el ChatBot.

Para la integración del ChatBot en Telegram, deberemos realizar los diferentes pasos en Telegram:

- Acceder a Telegram e ir a <https://telegram.me/botfather>, ya que lo necesitaremos para la creación de un nuevo bot.

- Hacer clic en el botón Start (Iniciar) en la interfaz web o bien, escribir /start.
- Hacer clic en **/newbot** o escribirlo y, luego, ingresar un nombre.
- Ingresar un nombre de usuario para el bot, que termine en "bot" (p. ej., garthsweatherbot), y no esté previamente escogido.
- Copiar el token de acceso generado, que necesitaremos para utilizar en DialogFlow y comunicarlo con la API .

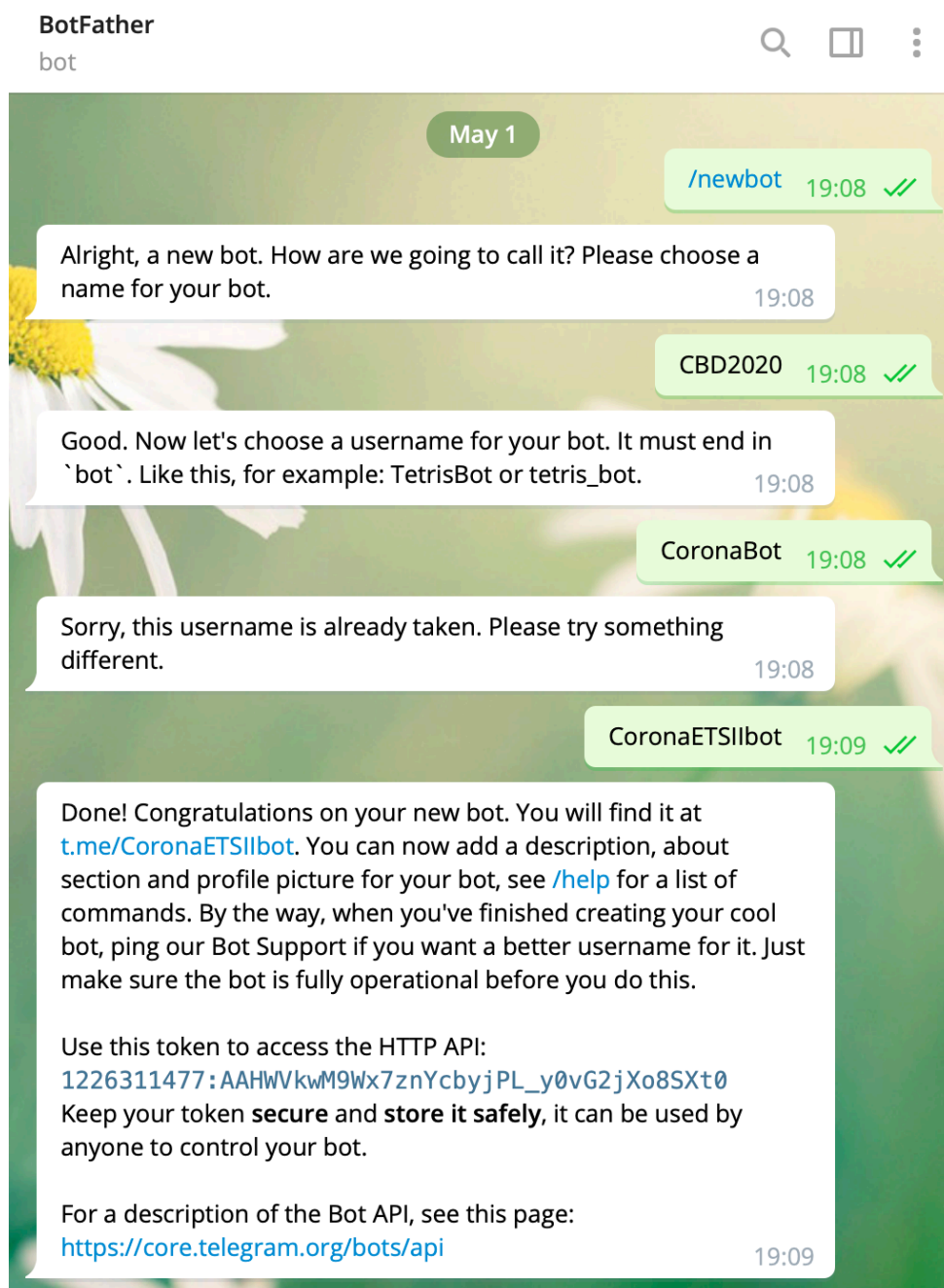


Fig 11. Configuración en Telegram

Una vez realizados estos pasos, deberemos dirigirnos a DialogFlow, donde deberemos llevar a cabo los siguientes pasos, dentro del apartado “Integrations”:

- Hacer clic en el mosaico Telegram.
- Pegar el Token de acceso en el campo relacionado.
- Haz clic en el botón Start (Iniciar).

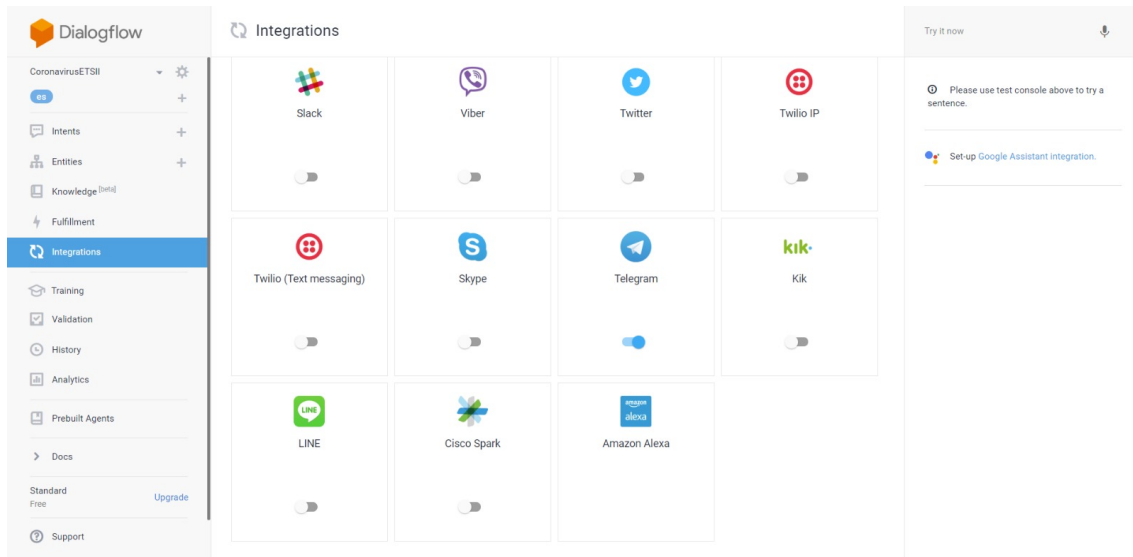


Fig 12. Configuración en DialogFlow dentro de Integrations

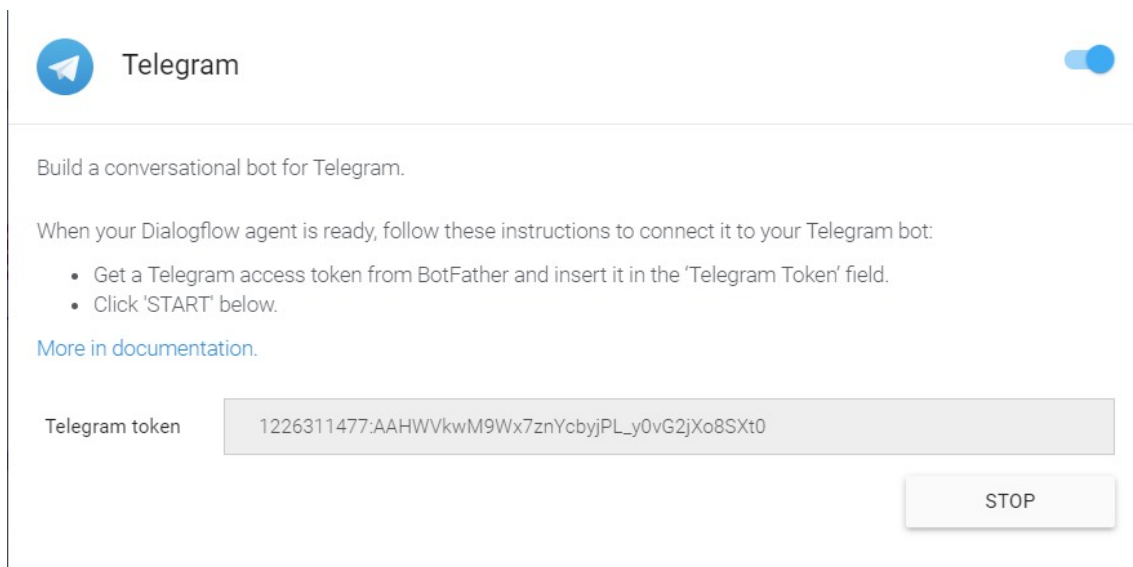


Fig 13. Configuración de Telegram con el token de la API

Una vez finalizado, podremos empezar a utilizar el ChatBot en Telegram, buscándolo bajo el siguiente nombre: @CoronaETSIIbot.

4. CONCLUSIONES

Uno de los principales objetivos en la realización de este trabajo, era el de acercarse al tratamiento de árboles de decisión desde la perspectiva de un proyecto real.

El principal aspecto para destacar en este proyecto ha sido la fase de diseño, pues es donde se aplica la mayor fase del conocimiento adquirido sobre los árboles de Diseño. Para ello, hemos estudiado previamente la tecnología para saber como funcionaba y hasta donde podíamos llegar con ella, y, tras eso, comenzamos con el diseño de los árboles de decisión.

En este proceso, recabamos toda la información necesaria de diferentes fuentes, destacando la página oficial de la Organización Mundial de la Salud.

Una vez obtenidos todos los contenidos necesarios, procedimos a categorizarlos de la manera que consideramos más objetiva y intuitiva dadas las informaciones que se presentaban, a su vez, teniendo en cuenta las decisiones que un usuario final podría tomar de cara a entregar el mejor producto.

Es por ello que ofrecimos la solución previamente detallada, con un resultado, que consideramos, muy notable teniendo en cuenta la cantidad de trabajo e integrantes del grupo que conformaban el proyecto.

Hemos podido ver en profundidad la herramienta de DialogFlow, del mismo modo, y con ella, explorar una tecnología útil y vanguardista con múltiples formas de utilizarse, siendo, así mismo, una herramienta que abarca un gran mercado en el mundo.

5. BIBLIOGRAFÍA

<https://cloud.google.com/dialogflow/docs?hl=es-419>

<https://cloud.google.com/dialogflow/docs/integrations/telegram>

<https://www.who.int/es/emergencies/diseases/novel-coronavirus-2019/advice-for-public/q-a-coronaviruses#:~:text=sintomas>

https://es.wikipedia.org/wiki/%C3%81rbol_de_decisi%C3%B3n

<https://www.redaccionmedica.com/secciones/sanidad-hoy/coronavirus-desescalada-salidas-franjas-de-paseo-y-deporte-al-boe-7341>

<https://www.elnortedecastilla.es/sociedad/salud/coronavirus-graficos/fases-desescalada-nueva-20200430174924-ntrc.html?ref=https:%2F%2Fwww.google.com%2F>

<http://www.dgt.es/es/covid-19/>