

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

Curso 2018 – 2019

Fecha	Versión
19/05/19	1.0

Grupo de prácticas	
Autores	Rol
Guerrero Cuenca, Claudia	Desarrollador
García Borrego, Carlos Jesús	Desarrollador
Valle Zarza, Felix	Desarrollador
Montaño Aguilera, AntonioManuel	Desarrollador

	<p>Complemento de Base de Datos Documentación del proyecto</p>
---	--

Índice de contenido

1. Introducción.....	2
2. Objetivo.....	2
3. Contenido del proyecto “ChatBot mediante árboles de decisión”	3
4. Conclusiones	23
5. Referencias.....	23
Glosario de términos.....	¡Error! Marcador no definido.
Anexos	23
I. Anexo	23



1. Introducción

Nos encontramos ante un proyecto para la asignatura de CBD = Complemento de Base de Datos, del grado de Ingeniería Informática Software.

En esta asignatura hemos visto los distintos tipos de base de datos que existen, y dentro de ellos, nuestro proyecto se centra en la creación de un ChatBot que responda basándose en árboles de decisión.

Para ello hemos decidido centrar el ChatBot sobre las preguntas básicas que suelen hacerse los estudiantes sobre la Beca Erasmus, tanto antes de tenerla como una vez aceptada. Nuestro ChatBot funcionará en Telegram y simplemente con agregar al usuario **@ErasmusETSIIBot** podemos preguntar cualquier tipo de dudas acerca del tema, obteniendo la respuesta a esta.

Para tener todas esas preguntas y respuestas es necesario tener una Base de Datos donde almacenar la información. Para nosotros será DialogFlow, que es la misma plataforma que hemos usado para implementar la funcionalidad.

2. Objetivo

El objetivo de este proyecto, fuera del ámbito colegial, sería la obtención de un Bot (Robot) creado en Telegram que sea capaz de responder a las dudas que los estudiantes tengan acerca de la Beca Erasmus de una forma fácil y cómoda.

Esto produciría una mejora porque se podrían hacer distintas consultas y tener las respuestas de inmediato. Lo único necesario es la agregación del usuario a Telegram (fácil de obtener).

En el caso de que queramos saber toda la información o mayor parte de ella, pero no sabemos qué consultar nuestro proyecto tendrá implementada una función para ello:

Consiste en la sugerencia de preguntas cada vez que se dé una respuesta (que tienen un orden o un enlace con esta), de modo que puedes ir pulsando en las sugerencias que más te importen y obtener su solución.

3. Contenido del proyecto “ChatBot mediante árboles de decisión”

En este apartado se redacta toda la documentación que tenga relación con la implementación de nuestro proyecto “Crear un ChatBot mediante árboles de decisión”.

3.1. Árboles de decisión

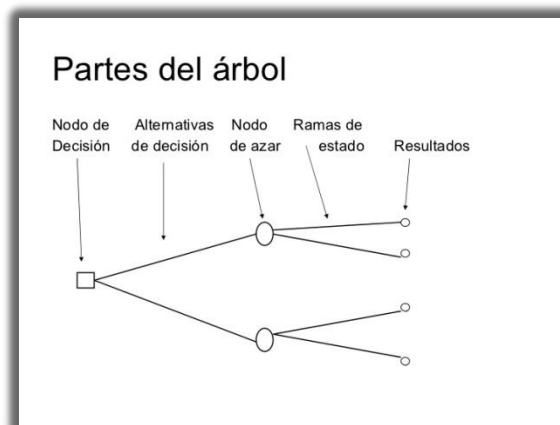
Es necesario conocer en qué consiste un árbol de decisión:

Es un mapa de los posibles resultados que tiene una serie de decisiones relacionadas. Dado un conjunto de datos, se fabrican diagramas de construcciones lógicas que sirven para representar y categorizar las condiciones que ocurren de forma sucesiva en la resolución de un problema.

Parte de un nodo inicial y se va ramificando en los distintos resultados posibles. Cada uno de estos resultados se vuelve a ramificar en otros resultados posibles. Y así sucesivamente.

Distintos elementos que puede tener un árbol de decisión:

- Nodos → Es el momento en el que hay que tomar una decisión entre distintas posibilidades.
- Vectores → Son las distintas soluciones finales a las que se pueden llegar.
- Flechas → Uniones entre nodos que representan la acción de pasar de uno a otro eligiendo la decisión que esa flecha implique.
- Etiquetas → Dan nombre a la acción que se elige.



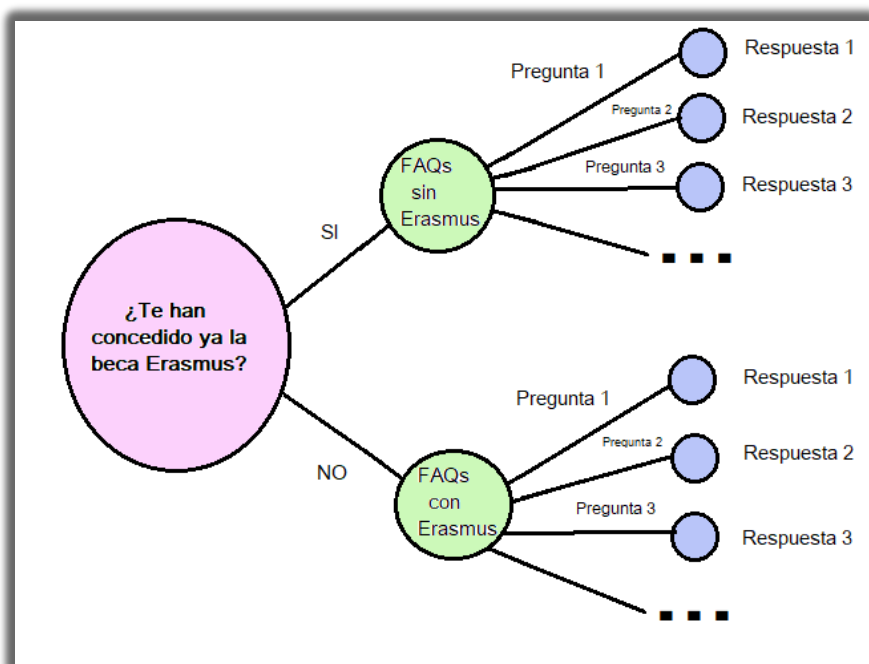
Los árboles decisión cumplen un conjunto de reglas:

- Se empieza con un nodo inicial, que no puede ser apuntado con ninguna flecha.
- El resto de nodos sí son apuntados por una flecha.
- Existe un único camino para llegar del nodo inicial a cada una de las distintas respuestas. Las decisiones son excluyentes, por tanto, no hay distintos caminos para llegar a una misma solución.

3.2. Árboles de decisión usados en ChatBot

La forma en la que implementamos los árboles de decisión para que nuestro ChatBot de una respuesta u otra, con respecto a las FAQs de las becas Erasmus, es sencilla:

- El nodo inicial haría la pregunta de si se ha obtenido ya la beca Erasmus, para saber si responder a preguntas de antes o de después de obtener la beca.
- Los siguientes nodos ya contendrán información sobre unas FAQs u otras, y a partir de ahí hay que seleccionar una cierta pregunta de la cual nos mostrará la respuesta.




Ahora mostramos un ejemplo específico de funcionarían los árboles de decisión en nuestro sistema de Telegram:



Problema: Podría darse el caso de que las preguntas que el usuario hace no estén escritas de la misma forma en la que se redactan en las FAQs.

Solución: “Entrenamiento” de las posibles preguntas + definición de sinónimos en DialogFlow (más próximamente).

	<p>Complemento de Base de Datos Documentación del proyecto</p>
---	--

3.3. Explicar tecnologías usadas

Para la creación del ChatBot, en nuestro proyecto decidimos centrarnos en responder a las preguntas más frecuentes que se dan sobre la beca Erasmus con el fin de resolver las dudas que los alumnos futuros tengan sobre el tema.

Para su realización, se han usado las diferentes tecnologías y documentos:

- **FAQs Erasmus** → Diferentes documentos de FAQs que se ofrecen en la página web de la ETSII siendo estos:
 - https://www.informatica.us.es/docs/rrii/soy_titular_erasmus.pdf
 - <https://www.informatica.us.es/docs/rrii/quiero-irme-erasmus-FAQ.pdf>
- **Telegram** → Nuestro objetivo, obviamente, es hacer un ChatBot para Telegram que haga la función de FAQ sobre información de las becas Erasmus. Por tanto, la tecnología usada para realizar las pruebas con el ChatBot ha sido Telegram, donde dicho Bot se puede encontrar como **@ErasmusETSIIBot**
- **DialogFlow** → Para todo el desarrollo del árbol de decisiones de respuestas del ChatBot cuando se le haga una pregunta se ha usado DialogFlow, una herramienta muy fácil de usar y que nos ha servido para llevar a cabo este proyecto de una manera muy eficaz.

3.4. Implementación en DialogFlow

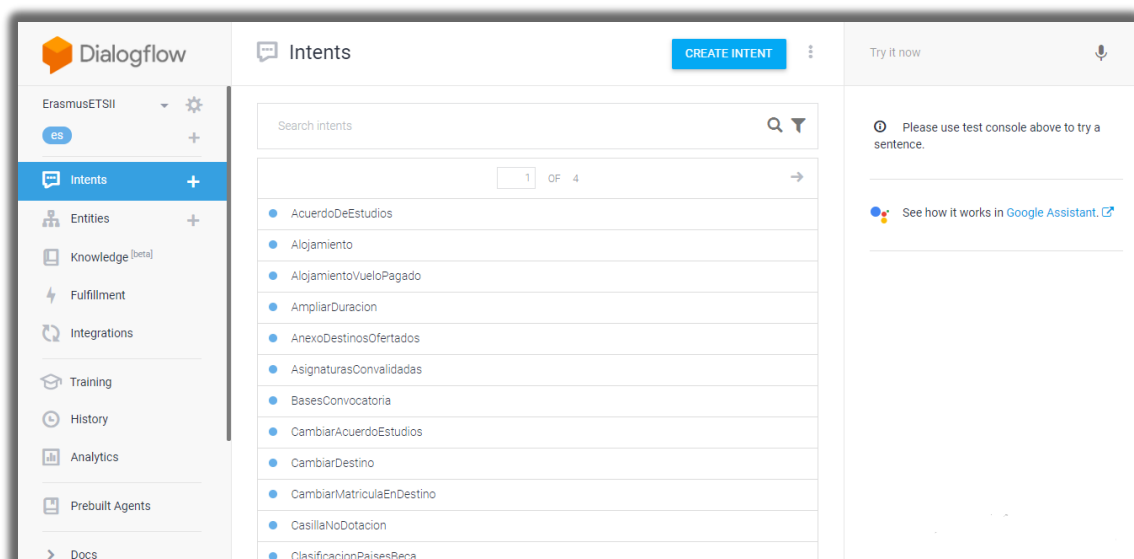
Dialogflow es un desarrollador de tecnologías de interacción persona-computadora, propiedad de Google, basada en conversaciones en lenguaje natural.

La compañía es más conocida por crear el 'Asistente', un compañero virtual para teléfonos inteligentes con Android, iOS y Windows Phone que realiza tareas y responde las preguntas de los usuarios en un lenguaje natural.



Lo primero que debemos hacer, es iniciar sesión con el correo que usamos para crear el agente ErasmusETSII (garguemonval@gmail.com), luego accedemos a <https://dialogflow.com/> y hacemos click en 'Go to Console' situado en la parte superior derecha de la pantalla.

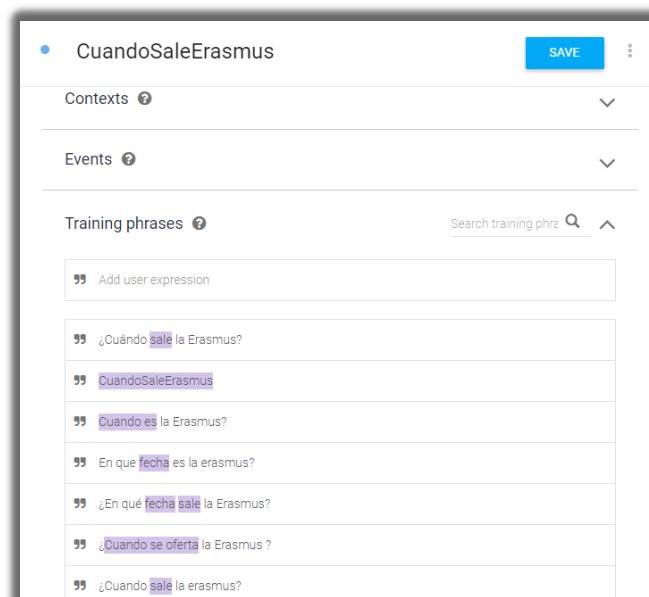
Y nos aparece la vista principal:



3.4.1 Intents

Esta pestaña se utiliza para poner las frases de entrada de los usuarios y las respuestas a dichas preguntas. Un intent se compone de:

- **Nombre de la intención:** en nuestro ejemplo es 'CuandoSaleErasmus'.
- **Contexts:** Podemos usarlo para hacer que un intent solo pueda aparecer si antes se produce la interacción con otro intent.
- **Events:** Camino opcional de trigger un intent sin necesidad de escribirlo explícitamente. Esta pestaña no la usaremos durante el proyecto.
- **Training phrases:** Aquí escribiremos las frases que pensamos que el usuario usará para obtener una determinada respuesta.



• CuandoSaleErasmus SAVE

Contexts ?

Events ?

Training phrases ? Search training phrs

” Add user expression

” ¿Cuándo **sale** la Erasmus?

” **QuandoSaleErasmus**

” **Quando es** la Erasmus?

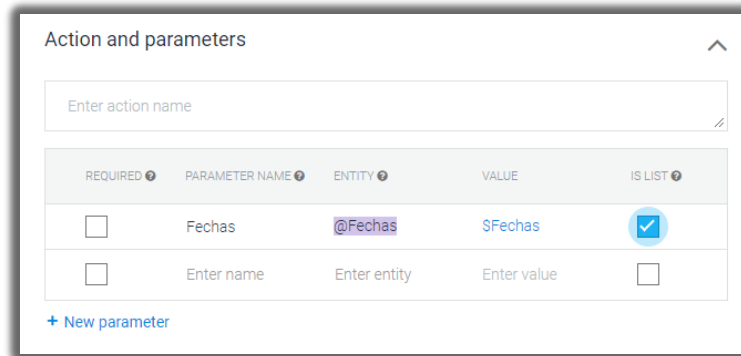
” En que **fecha** es la erasmus?

” ¿En qué **fecha sale** la Erasmus?

” ¿Quando se **oferta** la Erasmus ?

” ¿Quando **sale** la erasmus?

- **Action and parameters:** Aquí podemos añadir, modificar o eliminar las entidades para las palabras seleccionadas.

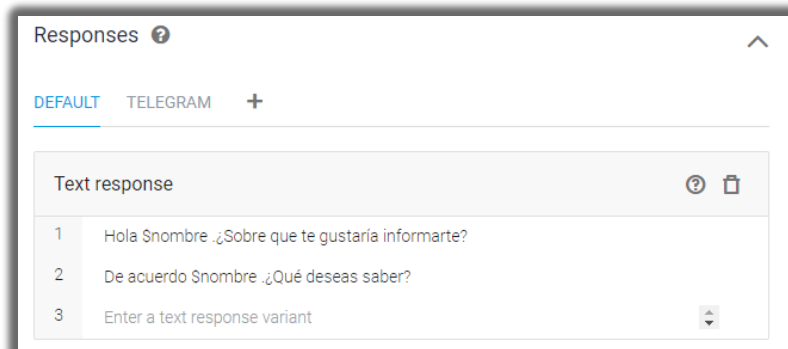


REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST
<input type="checkbox"/>	Fechas	@Fechas	\$Fechas	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>

+ New parameter

- **Responses:** En este apartado podemos devolver respuestas simples. De manera aleatoria se mostrará una de las respuestas que hayamos escrito para la pregunta que nos hizo el usuario.

Por ejemplo, si el usuario hubiese puesto que se llama Juan, nuestro agente respondería utilizando una de estas dos respuestas, sustituyendo la variable \$nombre (que sería una variable como \$Fechas, que usamos en la imagen anterior), por el nombre que nos dijo el usuario en la frase de entrada.



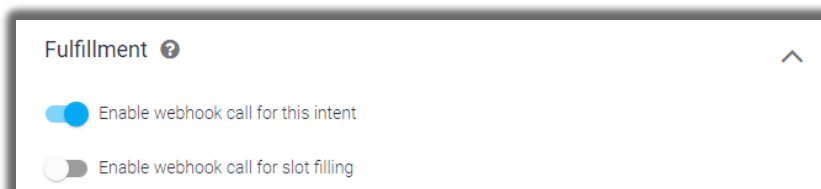
Responses ?

DEFAULT TELEGRAM +

Text response ?

- 1 Hola \$nombre ¿Sobre que te gustaría informarte?
- 2 De acuerdo \$nombre ¿Qué deseas saber?
- 3 Enter a text response variant

Por último, si prefieres responder con un mensaje más elaborado (más adelante indicaremos cómo hacerlo), activaremos la siguiente opción:




Fulfillment ?

☒ Enable webhook call for this intent

☐ Enable webhook call for slot filling

3.4.2 Entities

Se utilizan para agrupar conjuntos de palabras. Si tenemos una entidad seguramente tengamos varias entradas que pueden ser un conjunto de palabras o frases. Observe el siguiente ejemplo, para una mejor comprensión:



Tenemos una entidad que llamaremos 'Vegetales', luego tenemos dos entradas de entidad que son 'Zanahoria' y 'Cebolla'. Además tenemos, varios sinónimos para la entrada 'Cebolla' como son 'cebolla verde' o 'cebolleta'.

Por lo que en caso de que el usuario escriba la palabra Cebolleta en un input de entrada, la palabra cebolleta será seleccionada como entidad @Vegetales, y sinónimo de la palabra Cebolla.

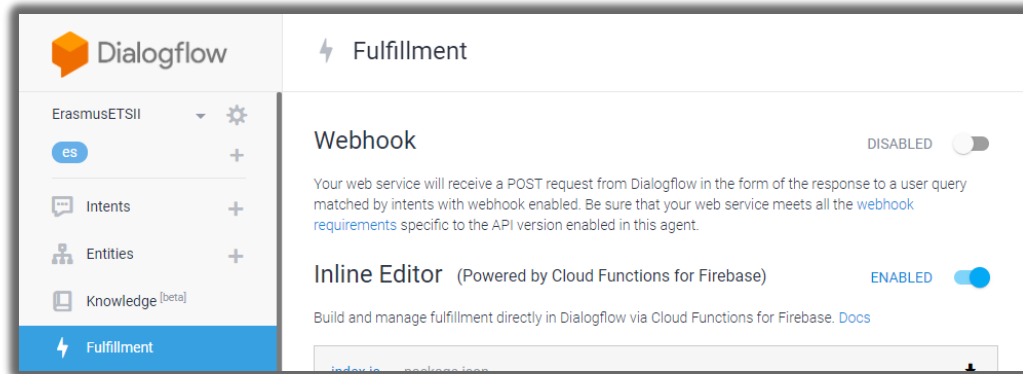
3.4.3 Fullfilment

Fulfillment es un código que se implementa como un webhook que le permite a su agente de Dialogflow llamar a la lógica empresarial en base a cada intento.

Durante una conversación, permite utilizar la información extraída por el procesamiento en lenguaje natural de Dialogflow para generar respuestas dinámicas o desencadenar acciones en su back-end.

Cómo dijimos anteriormente, para las respuestas elaboradas, escribiremos aquí el código para llevarlo a cabo.
(Si activamos en 'Intents' el fulfillment, saldrá las respuestas de esta pestaña y no las respuestas predeterminadas que pusistes anteriormente).

Primero habilitamos el editor online(esto solo se hace una vez durante la creación del proyecto):



En primer lugar vamos a declarar algunas variables, ya que algunas palabras clave se repiten y nos será más cómodo si queremos modificarlo en un futuro.

```
const fechasErasmus = {
  inicio: 'Octubre',
  fin: 'Diciembre'
};

const fechaContactoUniversidadDestino = '1 de Mayo';

const cambioAcuerdoEstudios = {
  inicio1C: '1 de octubre',
  fin1C: '1 de noviembre',
  inicio2C: '15 de febrero',
  fin2C: '15 de marzo',
  alemanes: '15 de abril'
};
```

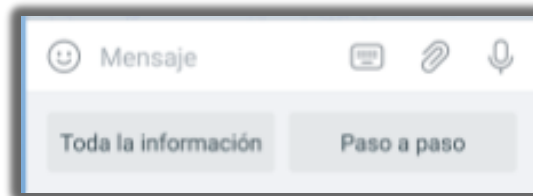
Escribimos aquí las funciones que deseemos hacer:

```
function general(agent){
  agent.add(`Aquí te dejo un enlace donde podrás ver toda la información disponible de la convocatoria`);
  agent.add(`${enlaces.programaErasmus}`); valor del parámetro 'programaErasmus' de la clase 'enlaces' declarada anteriormente
  agent.add(`Además puedes consultar:`); una clase 'enlaces' declarada anteriormente
  basesConvocatoria(agent);
  anexoDestinosOfertados(agent);
  agent.add(`Sino sabes por donde empezar, te iré sugiriendo las preguntas más frecuentes.`);
  agent.add(new Suggestion('Paso a paso'));
}

Las sugerencias pueden ser cualquier frase de entrenamiento de un intents
```

Para el mensaje de salida usamos “ `` ”(botón a la derecha de la P).

Las sugerencias aparecerán de la siguiente manera en nuestro dispositivo:



Y por último hacemos esto para poder invocar a la función:

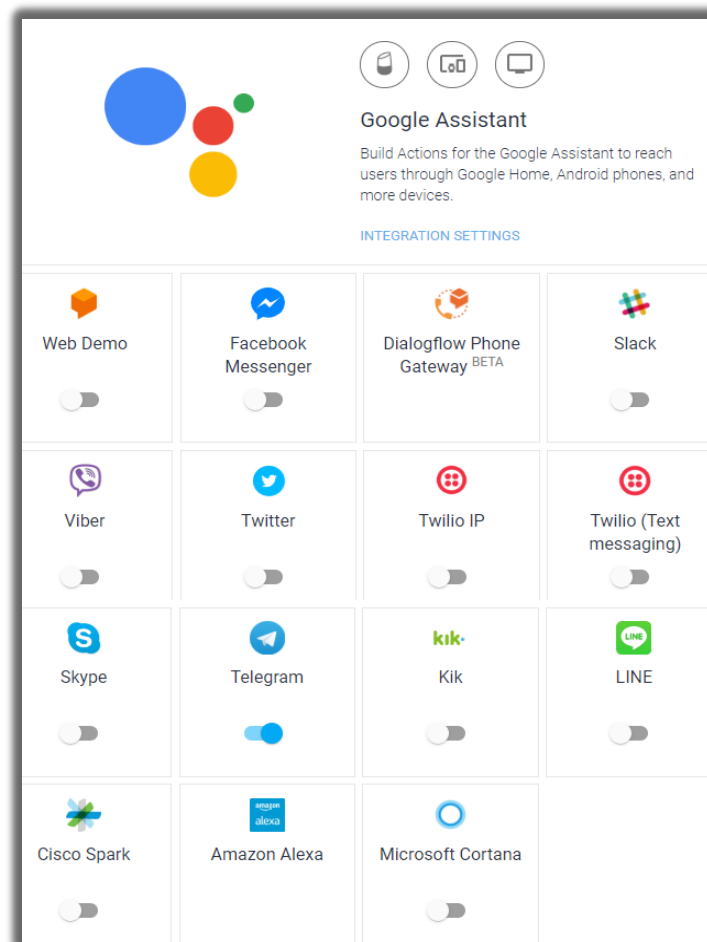
El 'General' corresponder con el Intent, y `general` corresponde con el nombre de la función.

```
let intentMap = new Map();
intentMap.set('Default Welcome Intent', welcome);
intentMap.set('Default Fallback Intent', fallback);
intentMap.set('EstudianteConDestino', estudianteConDestino);
intentMap.set('EstudianteSinDestino', estudianteSinDestino);
intentMap.set('BasesConvocatoria', basesConvocatoria);
intentMap.set('AnexoDestinosOfertados', anexoDestinosOfertados);
intentMap.set('General', general);
intentMap.set('Guia', guia);
```


Una vez terminemos, le damos a DEPLOY (si cambiamos de pestaña perdemos lo escrito).

3.4.4 Integración

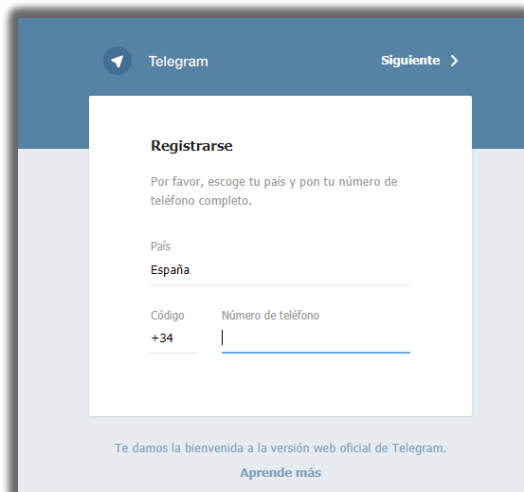
Una vez construimos nuestro agente, Dialogflow nos permite usar distintas herramientas de integración para así tener nuestro agente en múltiples plataformas, con tan solo un click.
En nuestro caso, integraremos Telegram.



Para usar nuestro agente que acabamos de crear, tenemos dos opciones, instalamos Telegram en nuestro dispositivo móvil o usamos Telegram Web(accediendo al siguiente enlace: <https://web.telegram.org/>).

	<p>Complemento de Base de Datos Documentación del proyecto</p>
---	--

Donde nos aparecerá la siguiente pantalla:



Telegram Siguiente >

Registrarse

Por favor, escoge tu país y pon tu número de teléfono completo.

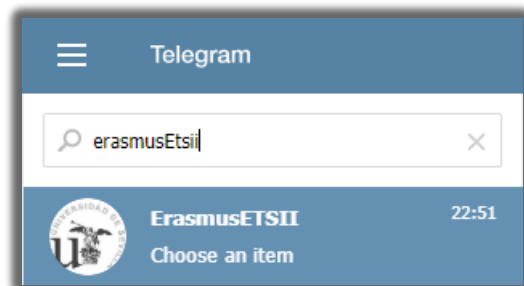
País
España

Código Número de teléfono
+34 |

Te damos la bienvenida a la versión web oficial de Telegram.
[Aprende más](#)


Una vez introducimos nuestro teléfono móvil, nos mandarán un código que deberemos introducir para acreditar que somos la persona poseedora de dicho teléfono.

Luego buscaremos: @ErasmusETSIIBot o simplemente ErasmusETSII.

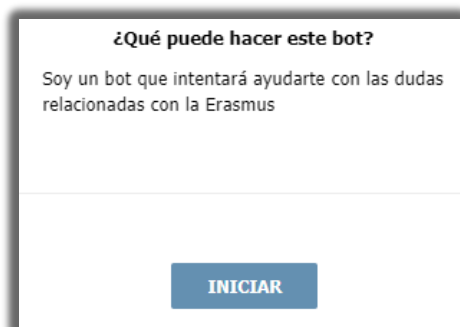


Telegram

erasmusEtsii

 ErasmusETSII 22:51
Choose an item

Y por último iniciaremos la interacción con el bot. Siempre que queramos iniciar una conversación desde cero, tenemos varias alternativas: borrar todos los mensajes, hacer click en 'Vaciar chat' o escribir /start.



¿Qué puede hacer este bot?

Soy un bot que intentará ayudarte con las dudas relacionadas con la Erasmus

INICIAR



3.4.5 Training

La página de entrenamiento muestra una lista de conversaciones. Los registros de las conversaciones de su agente aparecen automáticamente en la lista de conversaciones cuando los clientes conversan con su agente.

En esta pestaña podemos ver :

Conversation	Requests	No match	Date	
TELEGRAM_WELCOME	6	3	May 12	>
¿Puedo cursar asignaturas en vuestro destino y también otras asignaturas en la US?	2	0	May 11	>
¿Puedo cursar asignaturas en vuestro destino y también otras asignaturas en la US?	3	0	May 9	>
¿Qué necesito antes de irme?	1	0	May 9	>
donde puedo encontrar vuelo	13	1	May 9	>
TELEGRAM_WELCOME	6	1	Today	>
¿De qué depende que me vaya 1 semestre o un año?	3	0	Today	>
cuando me pagan el erasmus?	12	1	May 7	>
quiero contactar con alumnos de otros años	2	0	May 7	>
destinos ofertados	2	0	May 7	>

- **Conversation:** El primer enunciado en la conversación.
- **Request:** El número de peticiones en la conversación.
- **No Match:** Solicitudes sin coincidencia.
- **Date:** La fecha en que la conversación fue registrada o cargada.

Además, tenemos la opción de importar datos de entrenamiento pinchando en 'Upload' siempre que no supere los 3MB.

También nos ofrece la posibilidad de hacer click en alguna conversación y ver dijo el usuario, así como las entidades que se usan y si tuvo coincidencias con algún intent.



Complemento de Base de Datos Documentación del proyecto

donde puedo encontrar vuelo

May 9 13 REQUESTS 1 NO MATCH APPROVE

USER SAYS cuando tengo que llegar al destino

PARAMETER NAME	ENTITY	RESOLVED VALUE	
Tiempo	@Tiempo	cuando	✕
Lugares	@Lugares	destino	✕

INTENT CuandoLlegarDestino

USER SAYS que documento tengo que recoger al llegar al destino?

INTENT RecogerLegadaDestino

EVENT TELEGRAM_WELCOME

⚠ INTENT No intent invoked

3.4.6 History


En esta pestaña podemos ver todas las interacciones que se han producido entre un usuario y el bot. Pudiendo nosotros filtrar por plataformas (Telegram, Facebook, etc), por todas las conversaciones o conversaciones sin coincidencias o por fechas.

History

All platforms All conversations May 1, 2019 May 6, 2019

Conversation Date

USER	Mayte	May 2, 4:09 PM
AGENT		May 2, 4:09 PM
USER	Me han dado una erasmus, qué hago ahora?	May 2, 4:10 PM
AGENT		May 2, 4:10 PM
USER	Si no encuentro rvui skencia para una asignatura, qué puedo hacer?	May 2, 4:14 PM
AGENT	No matched intent	May 2, 4:14 PM
USER	Puedo convalidar las asignaturas?	May 2, 4:15 PM
AGENT		May 2, 4:15 PM
TELEGRAM_WELCOME (8)		May 2
¿Las asignaturas que haga en el extranjero se me van a reconocer a mi vuelta? (2)		May 2

	Complemento de Base de Datos Documentación del proyecto
---	--

En esta imagen, podemos observar los datos que introduce el usuario y como nuestro agente le devuelve una respuesta para su pregunta, a excepción de una, que al no encontrar coincidencias nos lo indica con un símbolo de alerta en amarillo.

En caso de ser rojo, seguramente nos indica que no hemos habilitado correctamente el fulfillment y no puede dar ninguna respuesta.

3.4.6 Analytics

La página de análisis le da una idea de qué tan bien se está desempeñando su agente, para que pueda trabajar para mejorar aún más la experiencia del usuario que está proporcionando.

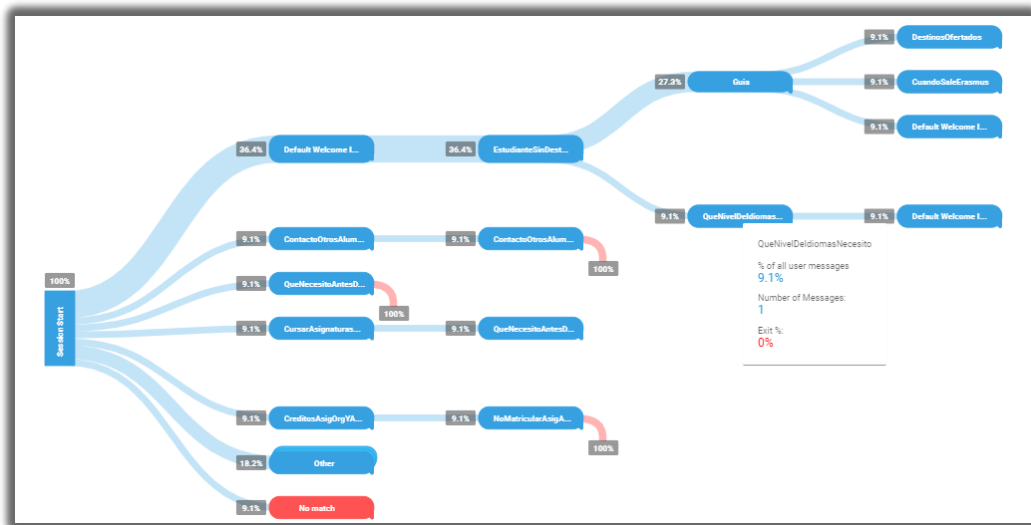
Mostramos dos tipos de datos relacionados con el agente y las conversaciones de las que forma parte:


- Datos de uso: Número de sesiones y consultas por sesión.
- Datos de NLU: intentos más utilizados y porcentajes de salida.

Intents ?				
Intent	Sessions	Count	Exit %	Agent response time
Default Welcome Intent	3	10	10%	
EstudianteConDestino	1	6	0%	
TengoAsignadaUnaPlaza	1	4	0%	
Sevius	1	4	0%	
UniversidadDeDestino	1	4	0%	
AcuerdoDeEstudios	1	4	0%	
QueNecesitoAntesDelrme	2	3	66.67%	
PasoAcuerdoEstudios	1	3	0%	
ContactoOtrosAlumnos	1	3	33.33%	

En esta imagen podemos ver el número de sesiones, las veces que son llamados y el porcentaje que tuvimos de 'sin coincidencia' en los últimos 7 días.

Además contamos con un mapa interactivo, que nos despliega las acciones para saber de que intent a que otro se realizan las interacciones, así como las veces que ha usado los intents respecto a los otros y mucha más información.



	Complemento de Base de Datos Documentación del proyecto
---	--

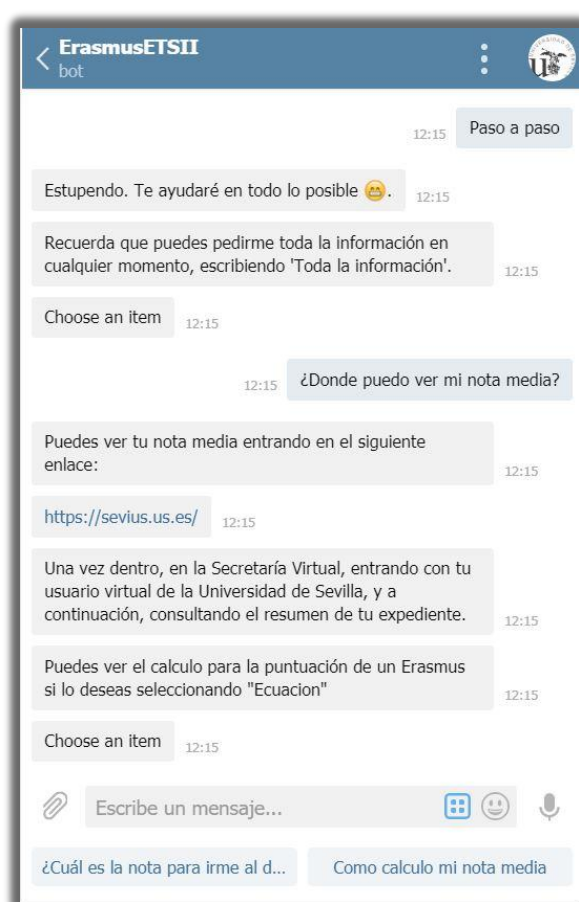
3.5. Comprobación las distintas maneras de llamar a una respuesta

La parte verdaderamente interesante de nuestro ChatBot es que haciendo muchas preguntas que se parezcan, nuestro Bot sea capaz de identificarlas y ofrecer la misma respuesta (la correcta).

A continuación, se muestra un ejemplo del proceso que se sigue para identificar una pregunta dando lugar a su respuesta:

1 - Se comenzará haciéndole la pregunta a nuestro Bot, mediante la cual trataremos de obtener información sobre la calificación media de los alumnos.

Una vez que se le hace la pregunta, el bot buscará en la frase las posibles “Entities” que contenga la frase, analizando la frase palabra a palabra para acortar la búsqueda.





2 - Una vez analizadas las palabras y encontradas las “Entities” en la frase, buscará en los “Intents” la frase más parecida gracias a su entrenamiento con frases insertadas por nosotros mismos.

En este caso, el “Intent” que busca es el siguiente:

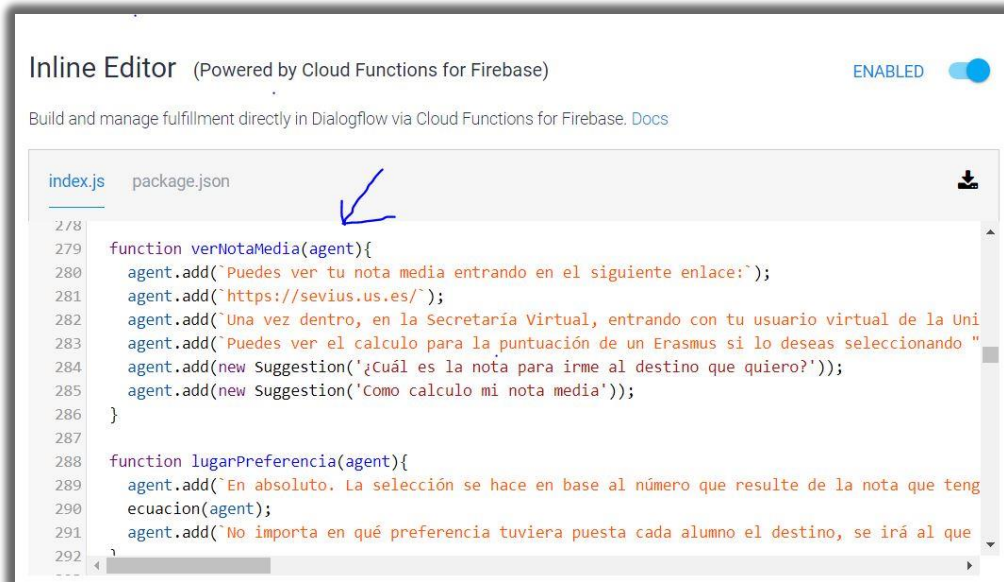
The screenshot shows the Dialogflow console for the intent 'SaberNotaPonderada'. The 'Training phrases' section contains the following phrases:

- Donde ver mi media
- Donde esta mi nota media
- Como se mi calificacion
- Como puedo saber mi media
- Ver nota media
- Como se mi nota media
- Donde puedo ver mi media
- Como puedo saber mi nota ponderada

3 - Una vez encontrado el “Intent” adecuado, se buscará el mapeo correspondiente en nuestro código, tal y como muestra la siguiente imagen:

```
index.js  package.json
680 intentMap.set('DestinosMasSolicitados', destinosMasSolicitados);
681 intentMap.set('PlanesEstudioDestinos', planesEstudioDestinos);
682 intentMap.set('NotaAlumnoParaDestino', notaAlumnoParaDestino);
683
684 intentMap.set('SaberNotaPonderada', verNotaMedia);
685 intentMap.set('EcuacionNotaMedia', ecuacion);
686 intentMap.set('LugarPreferencia', lugarPreferencia);
687 intentMap.set('CambiarDestino', cambiarDestino);
688 intentMap.set('PlazasLibresSinAcreditar', plazasLibresSinAcreditar);
689 intentMap.set('DineroBeca', dineroBeca);
690 intentMap.set('ClasificacionPaísesBeca', clasificacionPaíses);
691 intentMap.set('LatinoamericaDestino', latinoamericaDestino);
692 intentMap.set('CasillaNoDotacion', casillaNoDotacion);
693
694
```

4 - El Bot buscará la función “verNotaMedia” en el código y tras eso, tal y como se muestra en la imagen del chat, mostrará la respuesta y las sugerencias correspondientes que se encuentran dentro de la función.



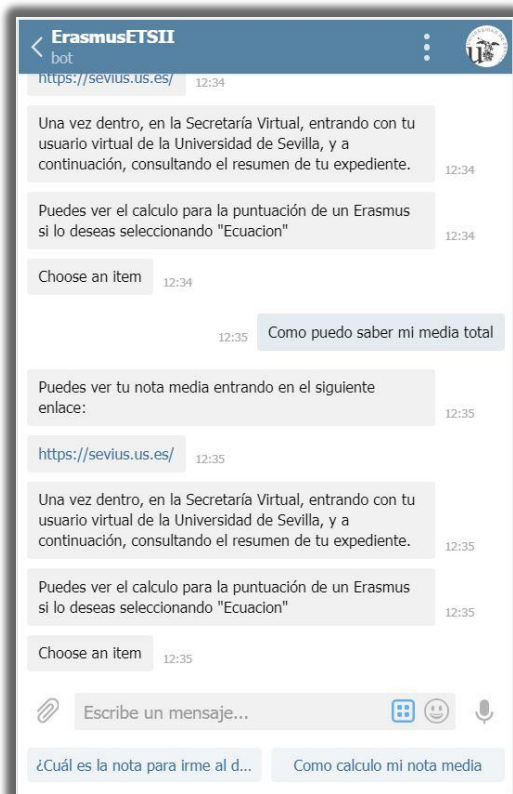
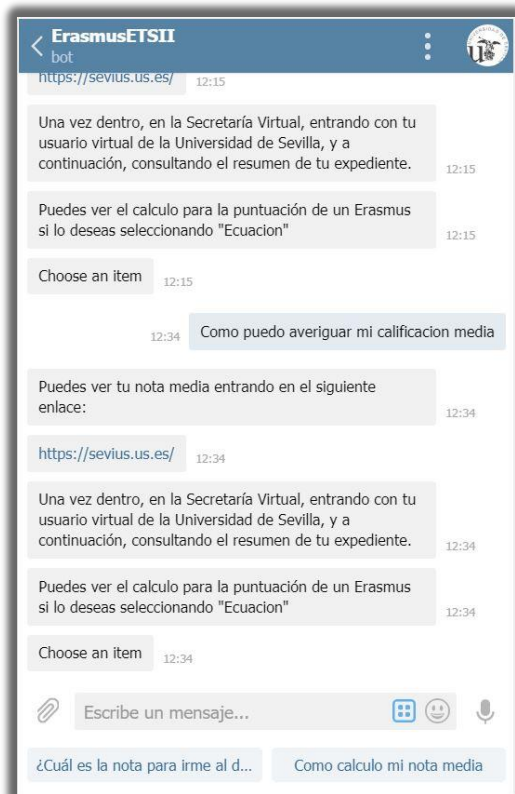
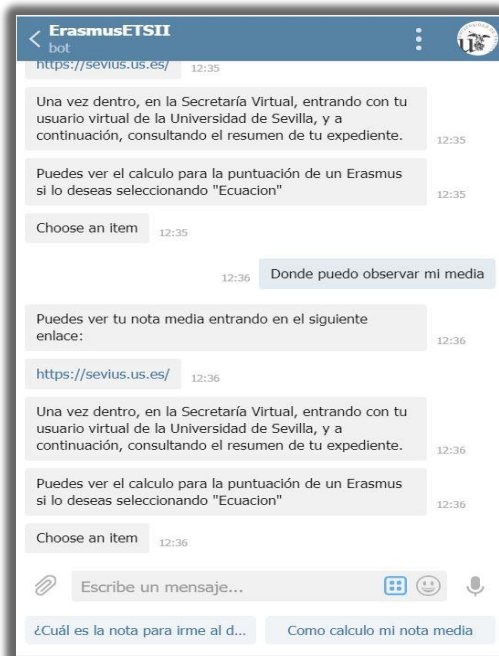
The screenshot shows the 'Inline Editor' interface, powered by Cloud Functions for Firebase. It displays a code editor with two files: 'index.js' and 'package.json'. The 'index.js' file is open, showing a JavaScript function named 'verNotaMedia' starting at line 279. A blue checkmark is drawn above the function name. The function contains several 'agent.add' calls and 'Suggestion' objects. The second function, 'lugarPreferencia', starts at line 288.

```
278
279 function verNotaMedia(agent){
280   agent.add(`Puedes ver tu nota media entrando en el siguiente enlace:`);
281   agent.add(`https://sevius.us.es/`);
282   agent.add(`Una vez dentro, en la Secretaría Virtual, entrando con tu usuario virtual de la Uni
283   agent.add(`Puedes ver el calculo para la puntuación de un Erasmus si lo deseas seleccionando "
284   agent.add(new Suggestion('{Cuál es la nota para irme al destino que quiero?}'));
285   agent.add(new Suggestion('Como calculo mi nota media'));
286 }
287
288 function lugarPreferencia(agent){
289   agent.add(`En absoluto. La selección se hace en base al número que resulte de la nota que teng
290   ecuacion(agent);
291   agent.add(`No importa en qué preferencia tuviera puesta cada alumno el destino, se irá al que
292
```



Complemento de Base de Datos Documentación del proyecto

Ahora vamos a buscar la misma respuesta con preguntas distintas para demostrar que el Bot ha aprendido correctamente con las frases insertadas en su aprendizaje.





4. Conclusiones

En conclusión, mediante la implementación del ChatBot para Telegram, se ahorra una enorme cantidad de tiempo y facilita la información necesaria a los alumnos que la soliciten de forma más inmediata, así como elimina una gran carga de trabajo a los coordinadores de Erasmus.

5. Referencias

- [1]. Wikipedia/Árboles de decisión
- [2]. Wikipedia/DialogFlow
- [3]. Tutoriales DialogFlow
- [4]. Wikipedia/ChatBot
- [5]. Teoría Bot de Telegram

Anexos

I. Anexo

No procede