

**Nombre:**

Ricardo Jara

**Materia:**

Inteligencia Artificial

**Fecha:**

23/05/202

**Tema:**

Examen IA

**~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~**

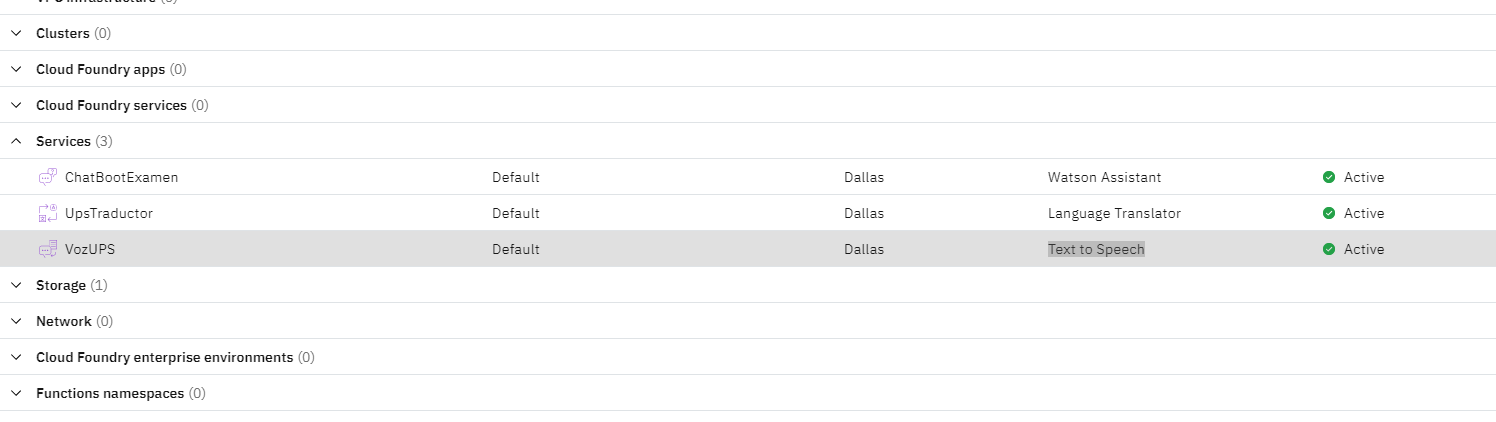
**Chatbot usando Watson Assistant, Language Translator y** **Text to Speech.**

Se desarrolla un sistema en Python el cual se conecta a Messenger y está pendiente de los mensajes enviados a una cuenta, cuando los mensajes son recibidos estos son traducidos utilizando el servicio de , **Language Translator** de IBM.

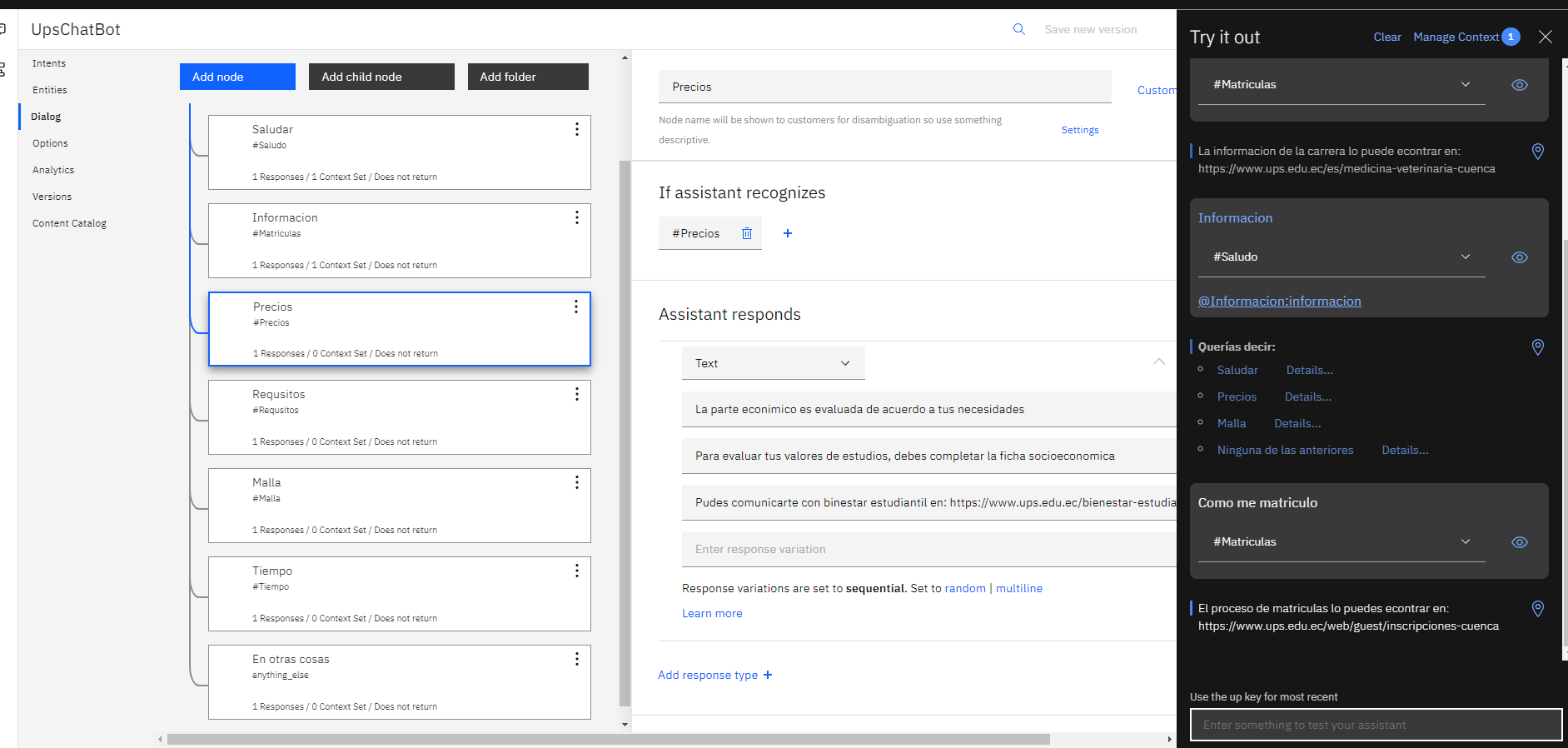
Una vez los mensajes son traducidos estos son enviados como mensaje al Chatbot de **Watson Assistant** éste nos devuelve una respuesta de cuerdo a la configuración del Chatbot en la página de IBM.

Con la respuesta obtenida del Chatbot utilizamos el servicio de **Text to Speech** para convertir la respuesta de Chatbot en audio y así responder al mensaje de Messenger con un audio y texto a la vez.

* **Desarrollo.**
* **Creamos nuestro servicio en https://cloud.ibm.com.**

****

* **Administramos el dialogo de nuestro Chatbot en Watson Assistant.**

****

* **En Python importamos las siguientes librerías.**

from fbchat import log, Client, Message  
from os.path import join, dirname  
from ibm\_watson import AssistantV2, LanguageTranslatorV3, TextToSpeechV1  
from ibm\_cloud\_sdk\_core.authenticators import IAMAuthenticator

* **Conectamos nuestro servicio de Watson Assistant**

authenticator = IAMAuthenticator('U4IKxuhQ4XBIrskFYSLLqrB29b\_A2fch9uL4gWQUZ-f4')  
assistant = AssistantV2(  
 version='2018-09-20',  
 authenticator=authenticator)  
assistant.set\_service\_url('https://api.us-south.assistant.watson.cloud.ibm.com/instances/20d0e70b-3f11-4c02-9137-205d38b948a9')  
assistant.set\_disable\_ssl\_verification(False)  
session = assistant.create\_session("633359aa-4a7e-4cfa-8ebe-78113a86ad21").get\_result()

* **Conectamos al servicio de Language Translator**

authenticatorT = IAMAuthenticator('0Iobj63HD2qz6ChKXOCkc1kARMJ8E9-Gkq1045FQIGf7')  
language\_translator = LanguageTranslatorV3(  
 version='2018-05-01',  
 authenticator=authenticatorT)  
language\_translator.set\_service\_url('https://gateway.watsonplatform.net/language-translator/api')

* **Conectamos al servicio de Text to Speech**

authenticatorV = IAMAuthenticator('rQkJz0iTpTyboZqk6SymQ2hh6zfG7sfmxdZBD9V9qQIV')  
service = TextToSpeechV1(authenticator=authenticatorV)  
service.set\_service\_url('https://stream.watsonplatform.net/text-to-speech/api')

* **Definimos un método que envía una mensaje y recibe la respuesta del Chatbot**

def mensaje(text, session):  
 message = assistant.message("633359aa-4a7e-4cfa-8ebe-78113a86ad21",  
 session["session\_id"],  
 input={'message\_type': 'text','text': text}).get\_result()  
 return (message['output']['generic'][0]['text'])

* **Definimos un método que realiza la traducción de un texto que recibe como parámetro.**

def traducir(text,language\_translator):  
 translation = language\_translator.translate(  
 text=text, model\_id='en-es').get\_result()  
 return translation['translations'][0]['translation']

* **Definimos un método que pasa de texto a voz guardando un .mp3 en la capeta loca.**

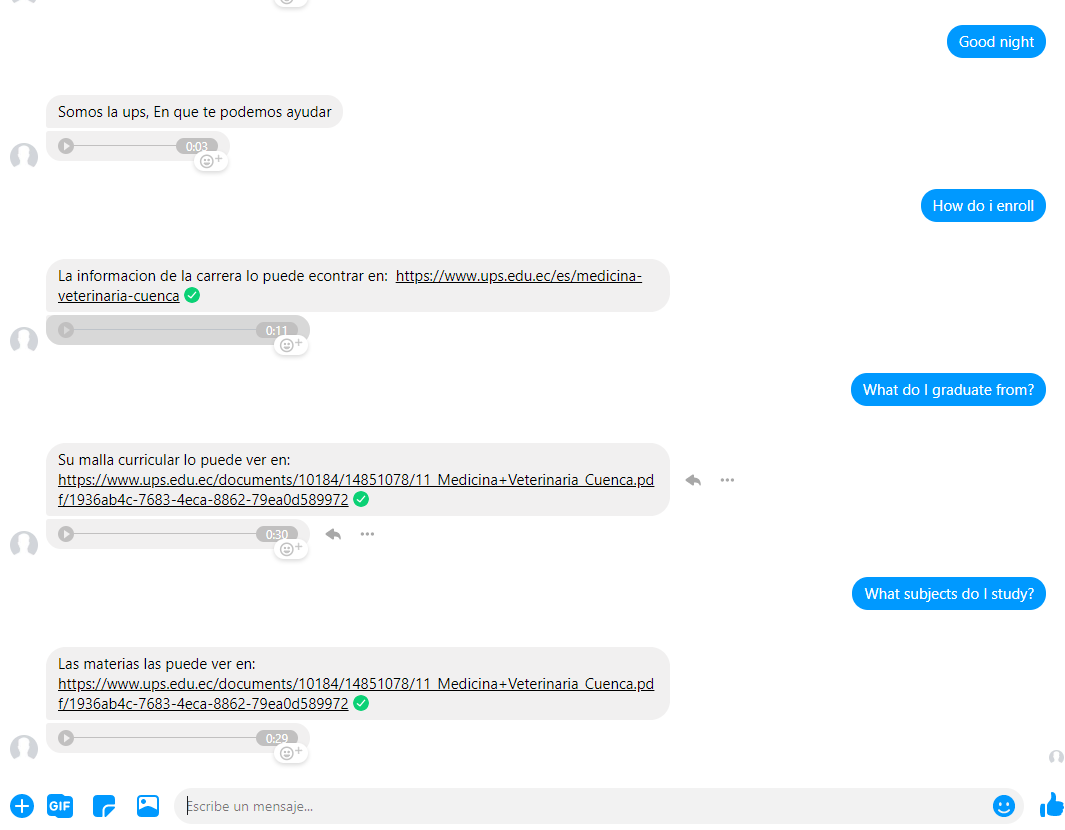
def voz(text, service):  
 with open(join(dirname(\_\_file\_\_), 'output.mp3'),  
 'wb') as audio\_file:  
 response = service.synthesize(  
 text, accept='audio/mp3',  
 voice="es-LA\_SofiaV3Voice").get\_result()  
 audio\_file.write(response.content)

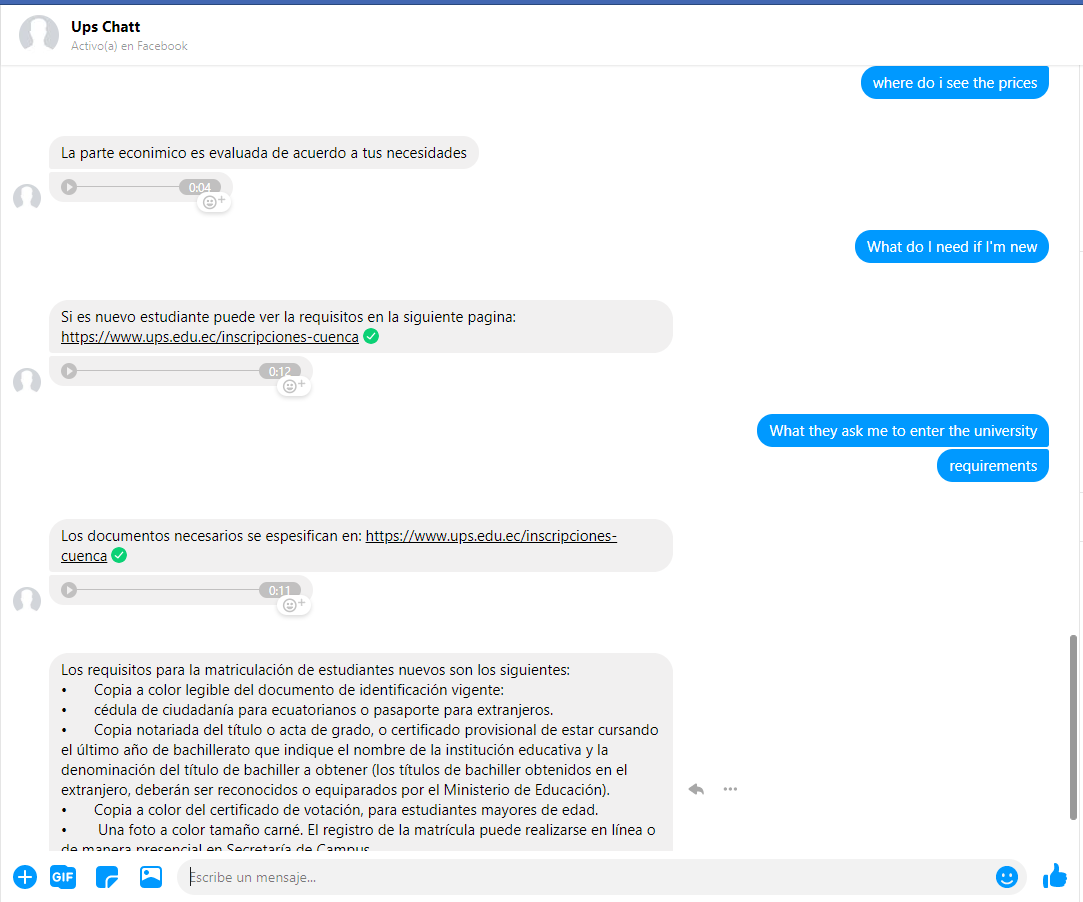
* **Por último, definimos el método que estará a la escucha de los mensaje que llegue a la cuenta de Messenger.**

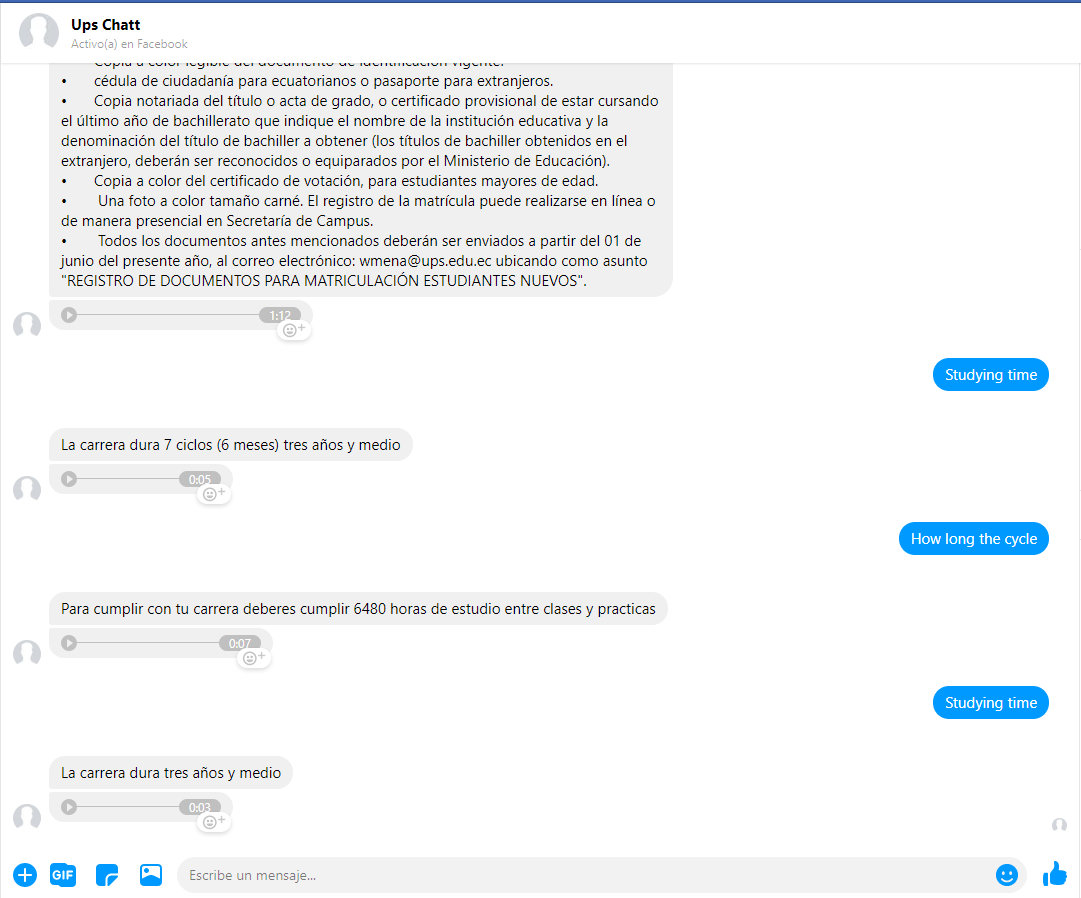
class EchoBot(Client):  
 def onMessage(self, author\_id, message\_object, thread\_id, thread\_type, \*\*kwargs):  
 self.markAsDelivered(thread\_id, message\_object.uid)  
 self.markAsRead(thread\_id)  
 if author\_id != self.uid:  
 messenger = message\_object.text  
 print(messenger)  
 traduccion = traducir(messenger, language\_translator)  
 print(traduccion)  
 respuesta = mensaje(traduccion,session)  
 print(respuesta)  
 voz(respuesta, service)  
 #self.send(Message(text=respuesta), thread\_id=thread\_id, thread\_type=thread\_type)  
 self.sendLocalVoiceClips('output.mp3',Message(text=respuesta),thread\_id=thread\_id, thread\_type=thread\_type)

Este método ejecuta a los demás métodos ya que cuando esté recibe un mensaje inmediatamente manda a llamar al método de **traducir** pasándola el mensaje y obteniendo el mensaje traducir los mensajes son recibidos en inglés y el método o el servicio los traduce al español (Este metodo usa el servicio de **Language Translator)**. Una vez con el mensaje traducido llamamos al método mensaje qué es lo que hace es consumir el servicio de **Watson assistant** y me retorna un mensaje de parte del Chatbot. Ya con la respuesta del Chatbot lo que se hace es llamar al método **voz** que lo que hace es recibir la respuesta del Chatbot y la convierte en un archivo punto MP3 el cual se le guarda en la carpeta local. Por último, el método de enviar o **self.sendLocalVoiceClips** lo que hace es recibir un archivo en audio y un mensaje, le pasamos el ID del cliente que queremos que reciba el mensaje y **así emos consumido los tres servicios de IMB, haciendo que un usuario de Messenger pregunte por informacion de la carrera de Medicina veterinaria. Y nuestro Chatbot le responda con audios.**

* **Ejecutamos el programa, y presentamos capturas de su funcionamiento.**

****

****

****

* **Código**

<https://github.com/RicardoVinicioJara/fbchat>

* **Examen IA**

<https://github.com/RicardoVinicioJara/FbChat/blob/master/Informe_IA.pdf>

* **Examen SE**

<https://github.com/RicardoVinicioJara/FbChat/blob/master/Informe_SE.pdf>