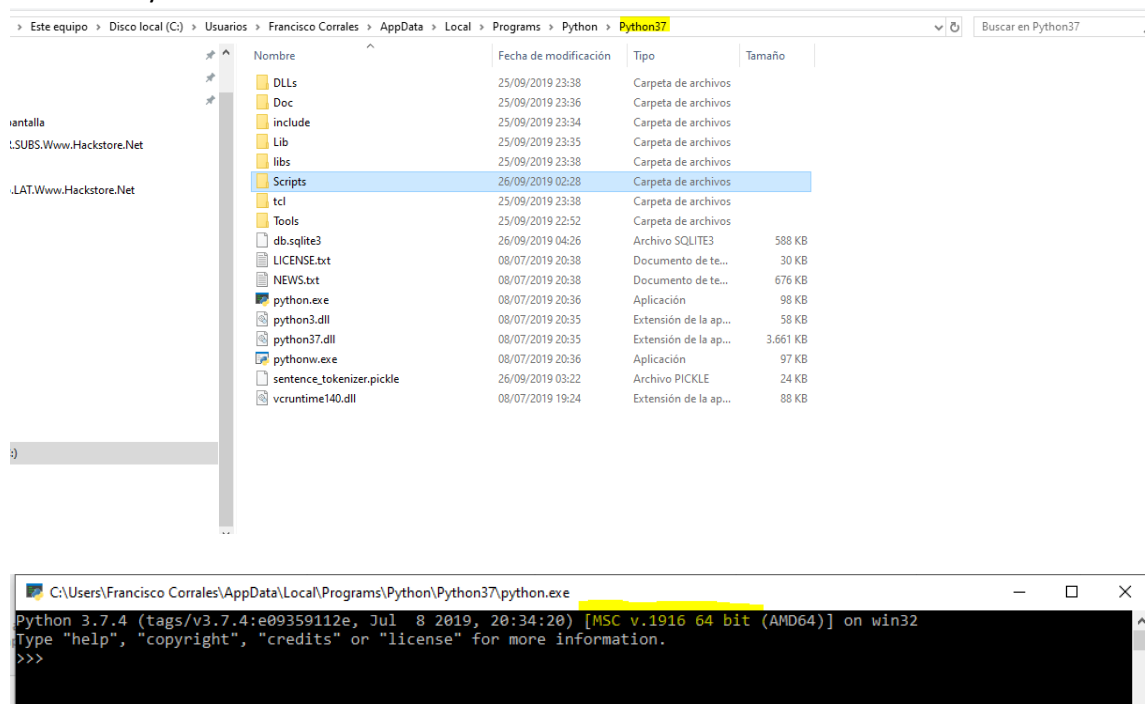


Para la actividad número 5 que comprende la instalación de un chatbot, se tuvo que seguir los siguientes pasos.

Tecnologías Utilizadas

- a) Python
- b) Flask
- c) Flask_socketio
- d) ChatterBot
- e) JQuery
- f) Html
- g) Scc

- 1) En mi caso particular tuve problemas con la versión de Python que tenía instalada ya que era de 32 bits, y esta daba problemas al instalar la herramienta ChatterBot.
- 2) Se instalo Python 3.7



- 3) En mi caso para poder instalar las diferentes herramientas debía moverme hasta la siguiente dirección
C:\Users\Francisco Corrales\AppData\Local\Programs\Python\Python37\Scripts>
De esta manera tenía acceso a "pip" par hacer las instalaciones correspondientes
- 4) Primero se hizo la instalación de flask mediante el siguiente comando
pip3.7.exe install flask

```
C:\Users\Francisco Corrales\AppData\Local\Programs\Python\Python37>cd Scripts
C:\Users\Francisco Corrales\AppData\Local\Programs\Python\Python37\Scripts>pip3.7.exe install flask
Requirement already satisfied: flask in c:\users\francisco corrales\appdata\local\programs\python\python37\lib\site-packages (1.1.1)
Requirement already satisfied: click>=5.1 in c:\users\francisco corrales\appdata\local\programs\python\python37\lib\site-packages (from flask) (7.0)
Requirement already satisfied: itsdangerous>=0.24 in c:\users\francisco corrales\appdata\local\programs\python\python37\lib\site-packages (from flask) (1.1.0)
Requirement already satisfied: Jinja2>=2.10.1 in c:\users\francisco corrales\appdata\local\programs\python\python37\lib\site-packages (from flask) (2.10.1)
Requirement already satisfied: Werkzeug>=0.15 in c:\users\francisco corrales\appdata\local\programs\python\python37\lib\site-packages (from flask) (0.16.0)
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\francisco corrales\appdata\local\programs\python\python37\lib\site-packages (from Jinja2>=2.10.1->flask) (1.1.1)
You are using pip version 19.0.3, however version 19.2.3 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

- 5) Luego se hizo la instalación de una herramienta de tipo socket. Mediante la siguiente instrucción

pip3.7.exe install python-socketio

Socket.IO permite la comunicación en tiempo real, bidireccional y basada en eventos.

Funciona en todas las plataformas, navegadores o dispositivos, centrándose

igualmente en la confiabilidad y la velocidad.

```
C:\Users\Francisco Corrales\AppData\Local\Programs\Python\Python37\Scripts>pip3.7.exe install python-socketio
Requirement already satisfied: python-socketio in c:\users\francisco corrales\appdata\local\programs\python\python37\lib\site-packages (4.3.1)
Requirement already satisfied: six>=1.9.0 in c:\users\francisco corrales\appdata\local\programs\python\python37\lib\site-packages (from python-socketio) (1.12.0)
Requirement already satisfied: python-engineio>=3.9.0 in c:\users\francisco corrales\appdata\local\programs\python\python37\lib\site-packages (from python-socketio) (3.9.3)
You are using pip version 19.0.3, however version 19.2.3 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

- 6) Luego se instaló Chatterbot mediante el siguiente comando

pip3.7 install chatterbot-corpus

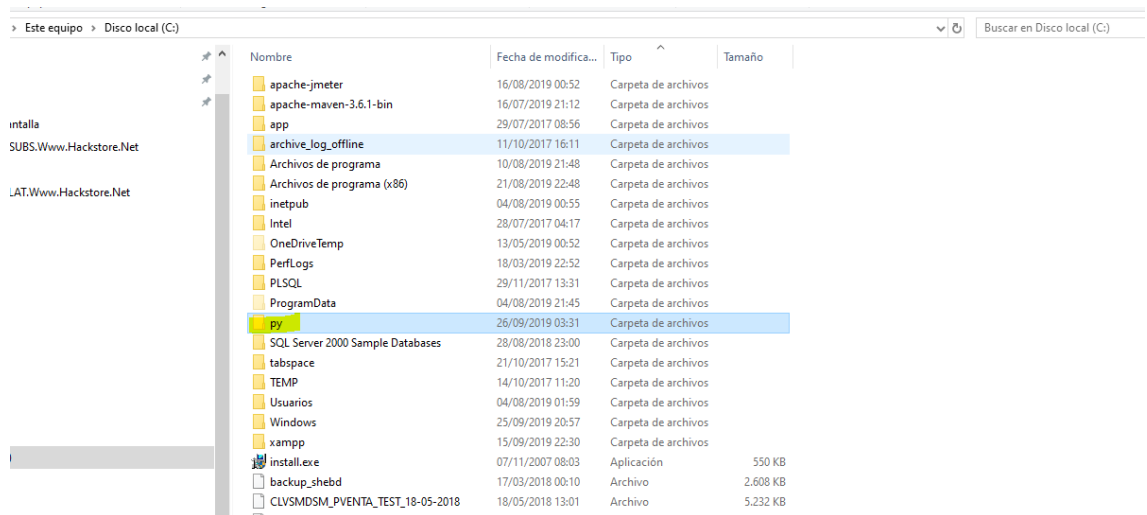
En mi caso fue la única manera de instalarlo

```
C:\Users\Francisco Corrales\AppData\Local\Programs\Python\Python37\Scripts>pip3.7 install chatterbot-corpus
Requirement already satisfied: chatterbot-corpus in c:\users\francisco corrales\appdata\local\programs\python\python37\lib\site-packages (1.2.0)
Requirement already satisfied: PyYAML<4.0,>=3.12 in c:\users\francisco corrales\appdata\local\programs\python\python37\lib\site-packages (from chatterbot-corpus) (3.13)
You are using pip version 19.0.3, however version 19.2.3 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

- 7) Ya tenemos todas la dependencia instaladas

rios > Francisco Corrales > AppData > Local > Programs > Python > Python37 > Scripts					Buscar en Scripts	
Nombre	Fecha de modificación	Tipo	Tamaño			
__pycache__	26/09/2019 02:22	Carpeta de archivos				
chardetect.exe	26/09/2019 02:22	Aplicación	101 KB			
easy_install.exe	25/09/2019 22:53	Aplicación	101 KB			
easy_install-3.7.exe	25/09/2019 22:53	Aplicación	101 KB			
f2py.exe	26/09/2019 02:23	Aplicación	101 KB			
flask.exe	26/09/2019 02:28	Aplicación	101 KB			
pip.exe	25/09/2019 22:53	Aplicación	101 KB			
pip3.7.exe	25/09/2019 22:53	Aplicación	101 KB			
pip3.exe	25/09/2019 22:53	Aplicación	101 KB			
plac_runner.py	26/09/2019 02:22	JetBrains PyChar...	3 KB			
spacy	26/09/2019 02:22	Archivo	1 KB			
tqdm.exe	26/09/2019 02:23	Aplicación	101 KB			

- 8) En la carpeta (C) creamos un directorio con el nombre de py



- 9) Creamos un archivo con el nombre app.py donde se va a escribir el código del chatbot, su entrenamiento, además de los métodos de respuesta mediante flask
- 10) En el app.py hacemos los import

```

1
2 from flask import Flask, render_template, request #
3 from flask_socketio import SocketIO, send #
4 from chatterbot import ChatBot #
5 from chatterbot.trainers import ChatterBotCorpusTrainer#
6 from chatterbot.trainers import ListTrainer#
7 import os
8

```

Flask nos servirá para crear un servidor web, de manera muy sencilla.
 Render_tempalte para enviar html al navegador
 Flask_socketio nos permite conexión en tiempo real dentro de Flask.

- 11) Para la inicialización del servidor web es necesario la siguiente línea de código

```

app = Flask(__name__)
app.config['SECRET_KEY'] = 'secret'
socketio = SocketIO(app)

```

Dentro de socketio está la conexión almacenada.

- 12) Con Flask podemos tener el primer método que devuelve algo mediante una función

```

@app.route('/')
def home():
    return render_template('index.html')

```

En este caso devuelve el “index.html” para este caso esta va hace la primera pagina que se va a observar cuando se entre la dirección http://127.0.0.1:5000/ por defecto

- 13) Para iniciar el chatbot utilizamos el inimport de chatterbot

```
bot = ChatBot("Walter Mitty")
trainer = ChatterBotCorpusTrainer(bot)
trainer.train("chatterbot.corpus.spanish")
trainer.train("chatterbot.corpus.spanish.greetings")
trainer.train("chatterbot.corpus.spanish.conversations")
```

En este caso se inicia en bot se le pasa el nombre de “Walter mitty”
Además, se le indica el entrenamiento mediante los trainer. El cual se le induce el español como lenguaje para hacer las respuestas.

- 14) Para obtener las respuestas del chatbot se configura otro método por flask

```
@app.route("/get")
def get_bot_response():
    userText = request.args.get('msg')
    return str(bot.get_response(userText))
```

Este tiene una función para obtener la respuesta, además, desde un cliente se obtiene por el método “get”, el userText toma el mensaje que se ha enviado desde el cliente y de hace un retorno mediante la función del chatbot de get_response usando como parámetro el “userText” que emos tomado desde el cliente.

- 15) Para correr el servidor web, en nuestra variable socketio tenemos el método run para inicializar el servidor llamado app de tipo flask.

```
if __name__ == '__main__':
    socketio.run(app)
```

- 16) A nivel del cliente se utiliza jquery para hacer lo llamadas

Esta es la función que hace un llamado get con el parámetro que se ha abtenido desde un input, se agrega el mensaje al hilo de la conversación y se manda a traer la respuesta del chatbot que también se agrega al hilo de conversación. Además se van bajando el escroll de manera automática mediante la misma función.

```
function getBotResponse() {
    var rawText = $("#textInput").val();
    var userHtml = '<p class="userText"><span>' + rawText + '</span></p>';
    $("#textInput").val("");
    $("#chatbox").append(userHtml);
    document.getElementById('userInput').scrollIntoView({block: 'start', behavior: 'smooth'});
    $.get("/get", { msg: rawText }).done(function(data) {
        var botHtml = '<p class="botText"><span>' + data + '</span></p>';
        $("#chatbox").append(botHtml);
        document.getElementById('userInput').scrollIntoView({block: 'start', behavior: 'smooth'});
    });
}
```

- 17) Se utilizan dos eventos para ejecutar la función

```

    }
    $("#textInput").keypress(function(e) {
        if(e.which == 13) {
            getBotResponse();
        }
    });
    $("#buttonInput").click(function() {
        getBotResponse();
    });
}

```

Tanto el enter como el click en un botón junto al input de donde se toma el mensaje para que le chatbot responda. Estas funciones son mediante jquery.

Ejecución de la aplicación:

Debe ir la siguiente ruta (Depende de la instalación que haya hecho de python)

..\AppData\Local\Programs\Python\Python37>

Debe correr el siguiente comando (Esto va a depender de donde haya descargado el proyecto)

python.exe C:\py\app.py

Así debe verse al iniciar el servidor:

```

C:\Users\Francisco Corrales\AppData\Local\Programs\Python\Python37>python.exe C:\py\app.py
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data] C:\Users\Francisco
[nltk_data] Corrales\AppData\Roaming\nltk_data...
[nltk_data] Package averaged_perceptron_tagger is already up-to-
[nltk_data] date!
[nltk_data] Downloading package stopwords to C:\Users\Francisco
[nltk_data] Corrales\AppData\Roaming\nltk_data...
[nltk_data] Package stopwords is already up-to-date!
Training conversations.yml: [#####] 100%
Training greetings.yml: [#####] 100%
Training trivia.yml: [#####] 100%
Training greetings.yml: [#####] 100%
Training conversations.yml: [#####] 100%
WebSocket transport not available. Install eventlet or gevent and gevent-websocket for improved performance.
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)

```

De manera seguida vamos al navegador y vamos la url: <http://127.0.0.1:5000/>

Este debe ser el resultado



Ya se puede hacer la interacción con el ChatBot

