

The background is a dark blue gradient with a subtle pattern of white dots. Overlaid on this are several concentric circles and arcs in a lighter blue color. Some of these arcs have degree markings, such as 40, 150, 160, 170, 180, 190, 200, 210, 220, 230, 240, 250, and 260. There are also small white arrows pointing in various directions, suggesting a sense of rotation or movement.

CHAT BOT

VELAZQUEZ CHAVEZ GLORIA

LUNA ROJO ALEXANDER

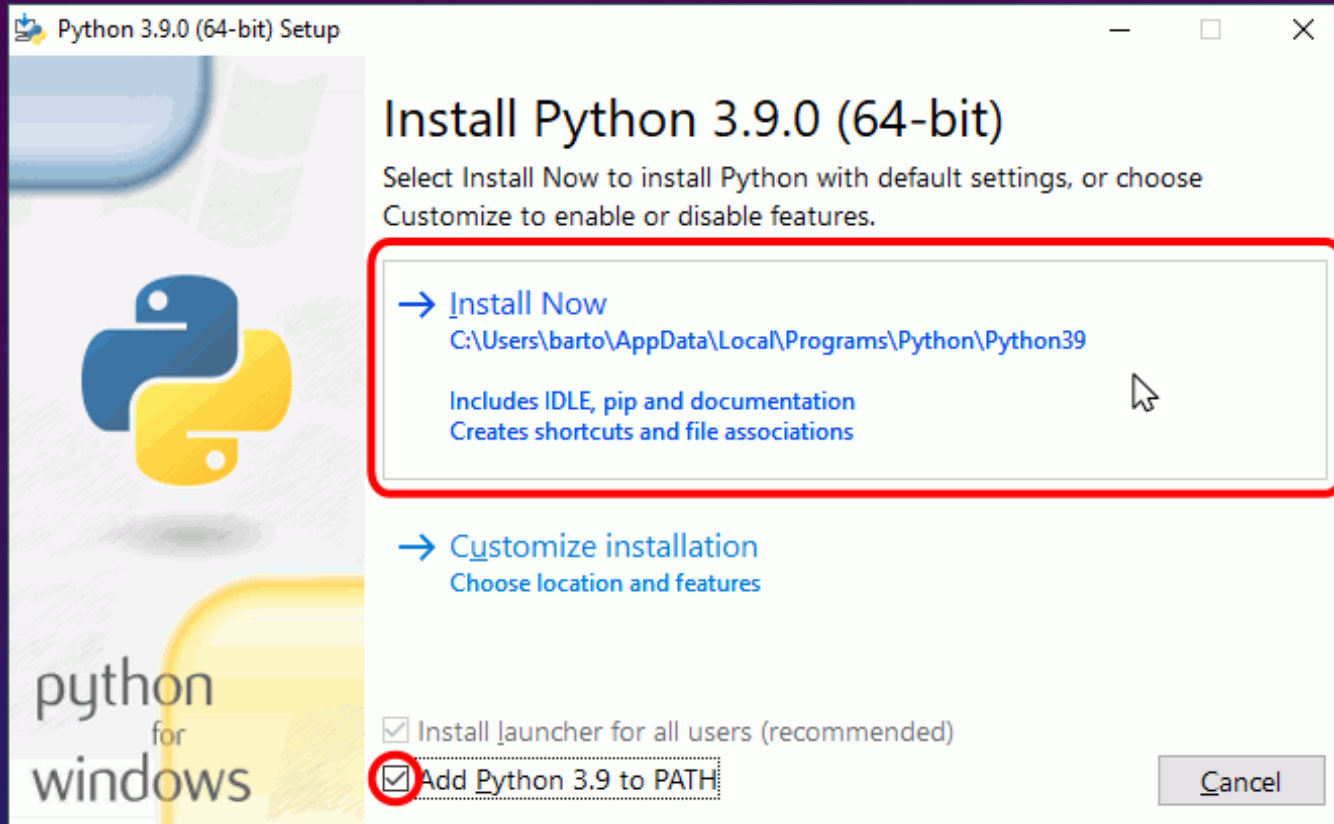
VALTIERRA CERVANTES JOEL

SANCHEZ VALDEZ DANIEL

HERRAMIENTAS:

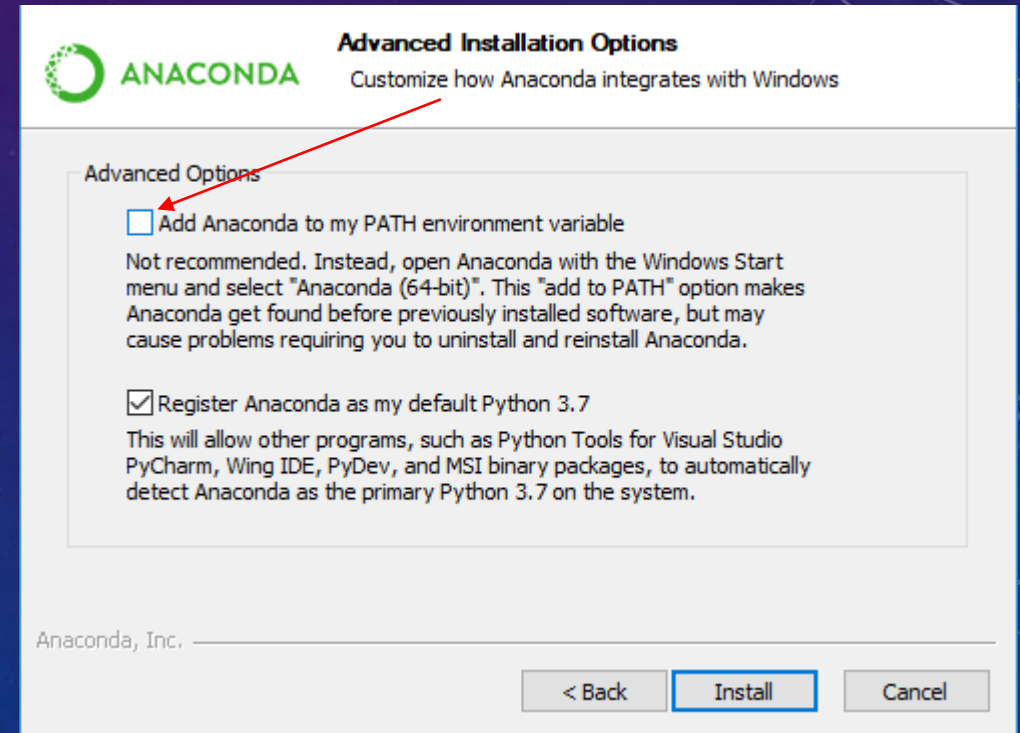
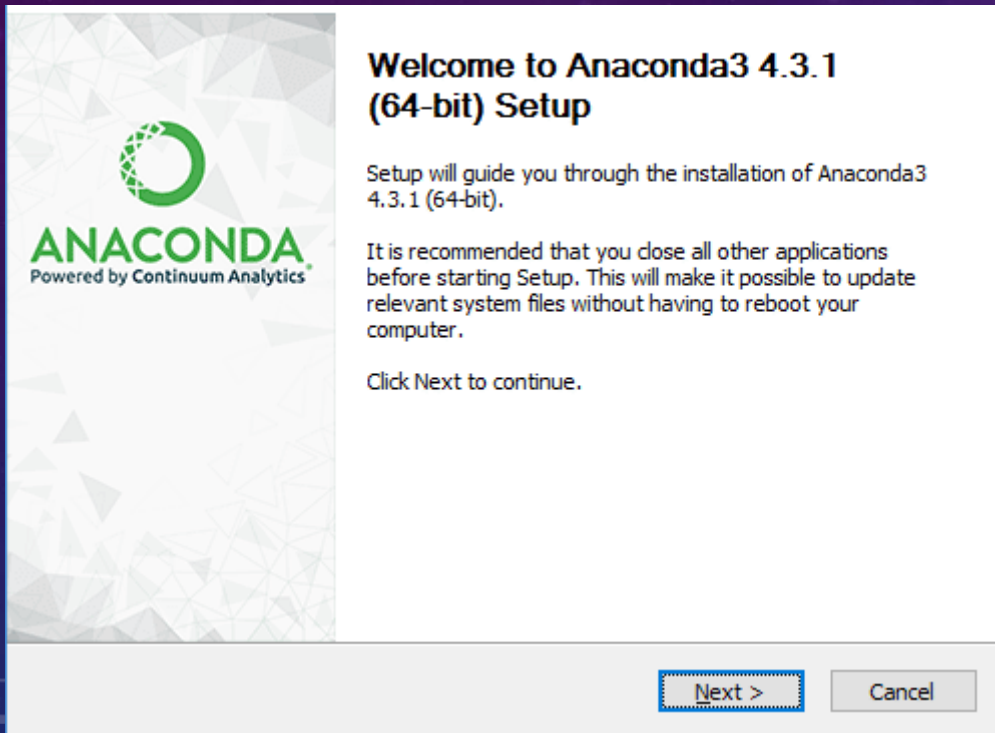
ANACONDA3

PYTHON 3.8



PARA INSTALAR
PYTHON SE
DESCARGA DESDE
LA PAGINA OFICIAL
Y AGREGAMOS AL
PATH.

DESCARGAR ANACONDA Y AGREGAR AL PATH



PARA LA CREACIÓN DEL ENTORNO DE DESARROLLO SE USA LA LÍNEA CONDA CRÉATE -N CHAT PYTHON 3.8

```
C:\WINDOWS\system32\cmd.exe - conda create -n chat python=3.8
Microsoft Windows [Versión 10.0.19041.746]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\windows 10pro>conda create -n chat python=3.8
WARNING: A space was detected in your requested environment path
'C:\Users\windows 10pro\.conda\envs\chat'
Spaces in paths can sometimes be problematic.
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

environment location: C:\Users\windows 10pro\.conda\envs\chat

added / updated specs:
- python=3.8

The following NEW packages will be INSTALLED:

ca-certificates pkgs/main/win-64::ca-certificates-2021.1.19-haa95532_0
certifi pkgs/main/win-64::certifi-2020.12.5-py38haa95532_0
openssl pkgs/main/win-64::openssl-1.1.1i-h2bbff1b_0
pip pkgs/main/win-64::pip-20.3.3-py38haa95532_0
python pkgs/main/win-64::python-3.8.5-h5fd99cc_1
setuptools pkgs/main/win-64::setuptools-51.3.3-py38haa95532_4
sqlite pkgs/main/win-64::sqlite-3.33.0-h2a8f88b_0
vc pkgs/main/win-64::vc-14.2-h21ff451_1
vs2015_runtime pkgs/main/win-64::vs2015_runtime-14.27.29016-h5e58377_2
wheel pkgs/main/noarch::wheel-0.36.2-pyhd3eb1b0_0
wincertstore pkgs/main/win-64::wincertstore-0.2-py38_0
zlib pkgs/main/win-64::zlib-1.2.11-h62dcd97_4

Proceed ([y]/n)? y
```

SE INSTALARAN LAS LIBRERÍAS TENSERFLOW, NAMPY, TFLEARN Y NLTK.

PARA HACERLO SE NECESITA ENTRAR AL ENTORNO CON LA LÍNEA:

ACTIVATE CHAT

```
C:\Users\windows 10pro>pip install tensor flow
Collecting flow
  Downloading flow-0.0.1.tar.gz (5.2 kB)
Collecting tensor
  Downloading tensor-0.3.6.tar.gz (50 kB)
    |████████████████████| 50 kB 380 kB/s
Requirement already satisfied: PyYaml in c:\programdata\anaconda3\lib\site-packages (from tensor) (5.3.1)
Requirement already satisfied: protobuf in c:\programdata\anaconda3\lib\site-packages (from tensor) (3.14.0)
Collecting construct
  Downloading construct-2.10.56.tar.gz (54 kB)
    |████████████████████| 54 kB 197 kB/s
Requirement already satisfied: six>=1.9 in c:\programdata\anaconda3\lib\site-packages (from protobuf->tensor) (1.15.0)
Collecting pysnmp
  Downloading pysnmp-4.4.12-py2.py3-none-any.whl (296 kB)
    |████████████████████| 296 kB 3.3 MB/s
Requirement already satisfied: pyasn1>=0.2.3 in c:\programdata\anaconda3\lib\site-packages (from pysnmp->tensor) (0.4.8)
Collecting pycryptodomex
  Downloading pycryptodomex-3.9.9-cp38-cp38-win_amd64.whl (14.1 MB)
    |████████████████████| 12.4 MB 3.3 MB/s eta 0:00:01
```

IMPORTACION DE LIBRERIAS

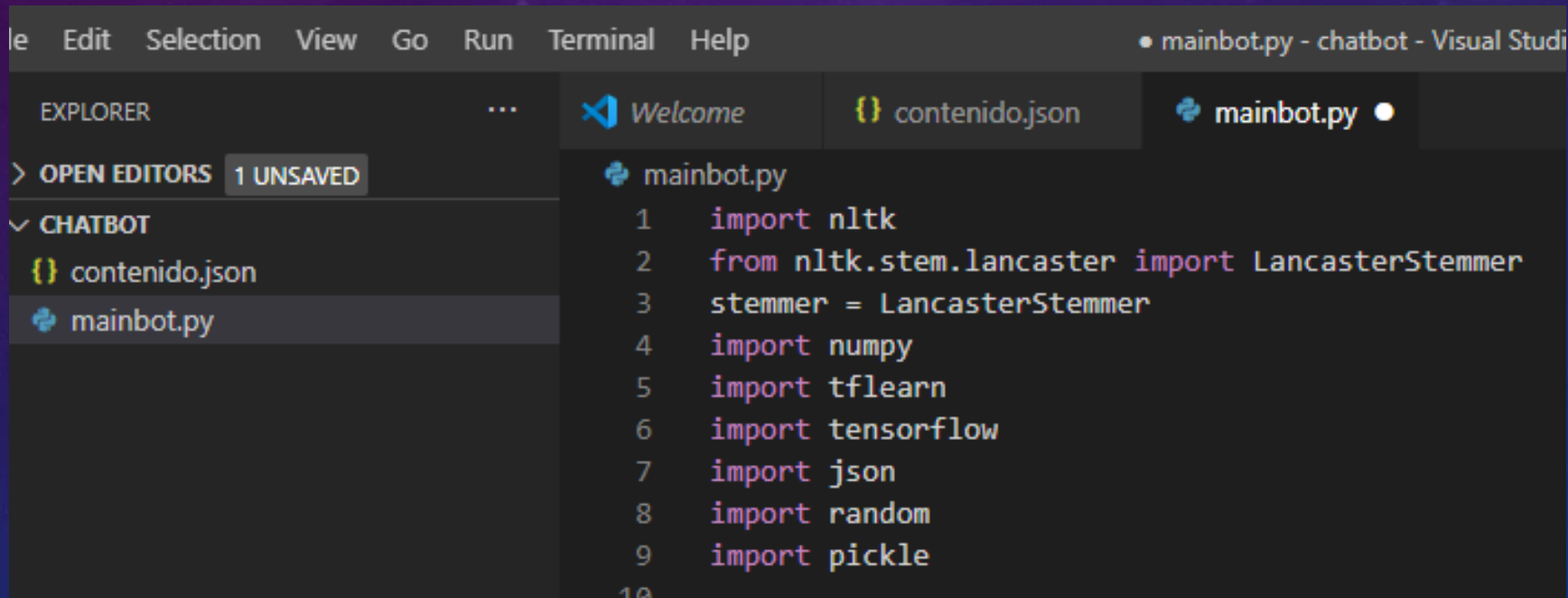
LA LIBRERÍA NLTK PERMITIRÁ EL PROCESAMIENTO DE LENGUAJE NATURAL.

NLTK.STEM.LANCASTER ESTO PERMITIRÁ TRANSFORMAR LAS PALABRAS, PARA QUE SEAN MAS ENTENDIBLES PARA EL CHATBOT

JSON QUE AHÍ SE TENDRÁ LA INFORMACIÓN

RANDOM PARA LAS RESPUESTAS ALEATORIAS

PICKLE PARA GUARDAR EL PROYECTO



The screenshot shows the Visual Studio Code interface. The Explorer panel on the left shows a project named 'CHATBOT' with two files: 'contenido.json' and 'mainbot.py'. The 'mainbot.py' file is open in the editor, showing the following Python code:

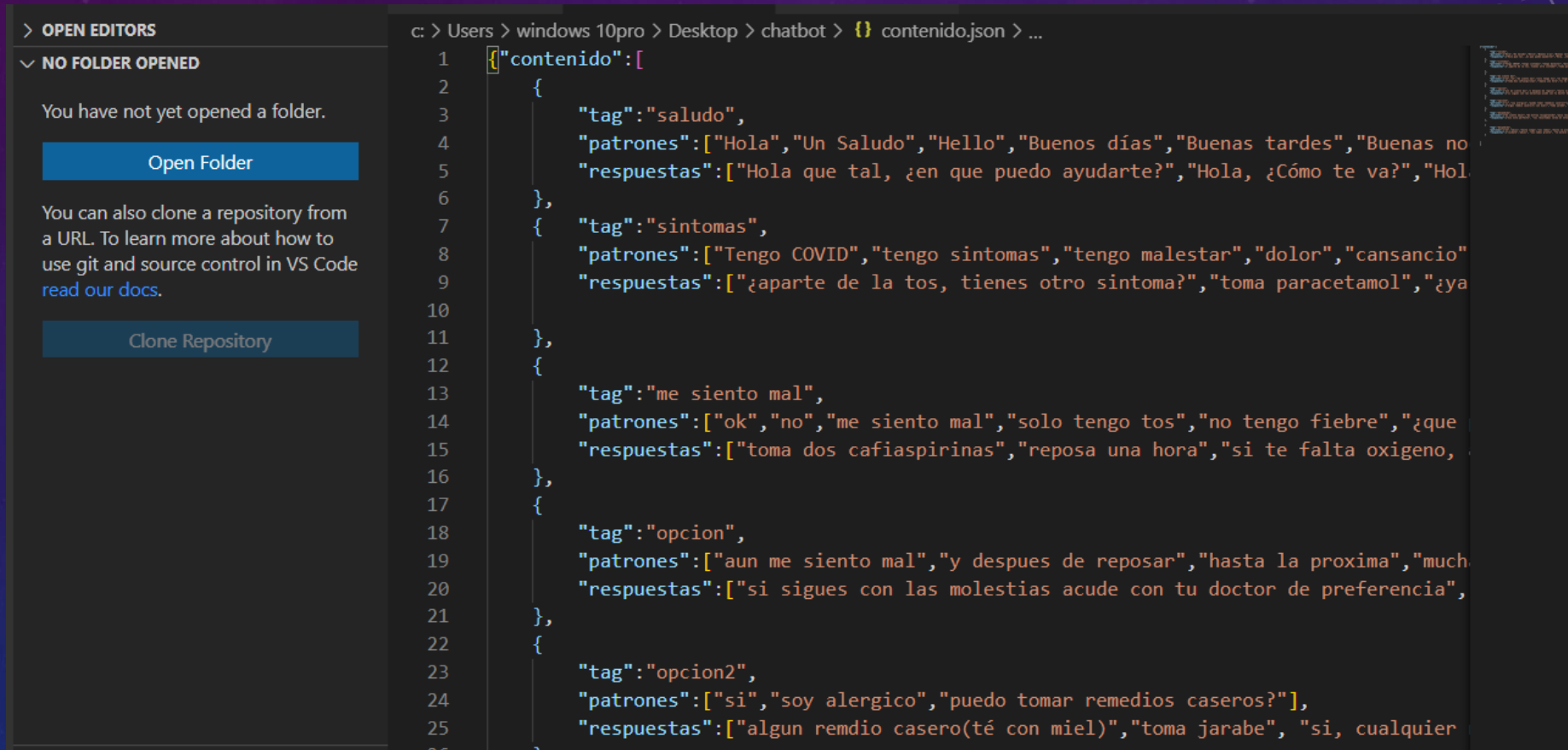
```
1 import nltk
2 from nltk.stem.lancaster import LancasterStemmer
3 stemmer = LancasterStemmer
4 import numpy
5 import tflearn
6 import tensorflow
7 import json
8 import random
9 import pickle
10
```

ARCHIVO J.SON

TAG: ES LA ACCIÓN

PATRONES: PALABRAS QUE FORMAN LA ORACIÓN

RESPUESTAS: SE LE DA RESPUESTA A TODOS LOS PATRONES



```
c: > Users > windows 10pro > Desktop > chatbot > {} contenido.json > ...
1  {"contenido": [
2    {
3      "tag": "saludo",
4      "patrones": ["Hola", "Un Saludo", "Hello", "Buenos días", "Buenas tardes", "Buenas no",
5      "respuestas": ["Hola que tal, ¿en que puedo ayudarte?", "Hola, ¿Cómo te va?", "Hol
6    },
7    {
8      "tag": "síntomas",
9      "patrones": ["Tengo COVID", "tengo síntomas", "tengo malestar", "dolor", "cansancio"
10     "respuestas": ["¿aparte de la tos, tienes otro síntoma?", "toma paracetamol", "¿ya
11   },
12   {
13     "tag": "me siento mal",
14     "patrones": ["ok", "no", "me siento mal", "solo tengo tos", "no tengo fiebre", "¿que
15     "respuestas": ["toma dos cafiaspirinas", "reposa una hora", "si te falta oxígeno,
16   },
17   {
18     "tag": "opcion",
19     "patrones": ["aun me siento mal", "y despues de reposar", "hasta la proxima", "much
20     "respuestas": ["si sigues con las molestias acude con tu doctor de preferencia",
21   },
22   {
23     "tag": "opcion2",
24     "patrones": ["si", "soy alergico", "puedo tomar remedios caseros?"],
25     "respuestas": ["algun remdio casero(té con miel)", "toma jarabe", "si, cualquier
26   ]
27 }
```


MANDAMOS A LLAMAR EL ARCHIVO J.SON

```
mainbot.py > ...  
1  import nltk  
2  from nltk.stem.lancaster import LancasterStemmer  
3  stemmer = LancasterStemmer()  
4  import numpy  
5  import tflearn  
6  import tensorflow  
7  import json  
8  import random  
9  import pickle  
10  
11  #nltk.download('punkt')  
12  with open("contenido.json") as archivo:  
13  |     datos = json.load(archivo)  
14  print(datos)
```

Se usara la línea:

```
palabras = [stemmer.stem(w.lower()) for w in palabras if w!="?"]
```

Para pasar la palabra en minúscula siempre y cuando la palabra sea diferente a ?

Se ordenara las palabras y los tags

```
palabras = sorted(list(set(palabras)))
```

```
tags = sorted(tags)
```

```
palabras = [stemmer.stem(w.lower()) for w in palabras if w!="?"]  
palabras = sorted(list(set(palabras)))  
tags = sorted(tags)
```

- Para entrenar al bot, se hace uso de una cubeta, por esto se usara un for, el cual enumerará las palabras por un índice guardado en x
- En el for w, si esta en palabras se añadirá en cubeta, 1 si esta 0 si no esta.
- La fila salida ayudara para obtener el índice de y para asignarle el valor de 1
- En entrenamiento se agregara la cubeta

```
for x, documento in enumerate(auxX):
    cubeta=[]
    auxPalabra= [stemmer.stem(w.lower()) for w in documento]
    for w in palabras:
        if w in auxPalabra:
            cubeta.append(1)
        else:
            cubeta.append(0)
    filaSalida = salidaVacía[:]
    filaSalida[tags.index(auxY[x])]=1
    entrenamiento.append(cubeta)
    salida.append(filaSalida)
print(entrenamiento)
print(salida)
```

CON TENSORFLOW AREMOS QUE SE REINICIE TODO

ENTRENAMIENTO DE LA RED NEURONAL
RED CON UNA FORMA NO DEFINIDA

```
entrenamiento = numpy.array(entrenamiento)
salida = numpy.array(salida)
tensorflow.compat.v1.reset_default_graph()

red = tflearn.input_data(shape=[None, len(entrenamiento[0])])
red = tflearn.fully_connected(red, 80)
red = tflearn.fully_connected(red, 80)
red = tflearn.fully_connected(red, 80)
red = tflearn.fully_connected(red, 80)
red = tflearn.fully_connected(red, 80)
red = tflearn.fully_connected(red, len(salida[0]), activation="softmax")
red = tflearn.regression(red)
```


ABRIMOS NUESTRO PROYECTO UBICANDO LA CARPETA DONDE SE ENCUENTRA

```
C:\WINDOWS\system32\cmd.exe - python mainbot.py
Microsoft Windows [Versión 10.0.19042.746]
(c) 2020 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Joe Valtierra C>activate chatbot

chatbot) C:\Users\Joe Valtierra C>cd desktop

chatbot) C:\Users\Joe Valtierra C\Desktop>cd proyect

chatbot) C:\Users\Joe Valtierra C\Desktop\proyect>python mainbot.py
2021-01-24 19:57:39.553575: W tensorflow/stream_executor/platform/default/dso_loader.cc:60] Could not load dynamic library 'cudart64_110.dll'; dlopen: cudart64_110.dll not found
2021-01-24 19:57:39.553707: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
```

PODREMOS TENER UNA CONVERSACION Y NUESTRO CHAT BOT CONTESTARA UTILIZANDO INTELIGENCIA, PATRONES.

```
--  
Training Step: 9995 | total loss: +-[1m-[32m0.07462+-[0m-[0m  
| Adam | epoch: 9995 | loss: 0.07462 - acc: 0.9533 -- iter: 43/43  
--  
Training Step: 9996 | total loss: +-[1m-[32m0.07418+-[0m-[0m  
| Adam | epoch: 9996 | loss: 0.07418 - acc: 0.9533 -- iter: 43/43  
--  
Training Step: 9997 | total loss: +-[1m-[32m0.07377+-[0m-[0m | time: 0.016s  
| Adam | epoch: 9997 | loss: 0.07377 - acc: 0.9533 -- iter: 43/43  
--  
Training Step: 9998 | total loss: +-[1m-[32m0.07339+-[0m-[0m  
| Adam | epoch: 9998 | loss: 0.07339 - acc: 0.9533 -- iter: 43/43  
--  
Training Step: 9999 | total loss: +-[1m-[32m0.07302+-[0m-[0m  
| Adam | epoch: 9999 | loss: 0.07302 - acc: 0.9534 -- iter: 43/43  
--  
Training Step: 10000 | total loss: +-[1m-[32m0.07268+-[0m-[0m  
| Adam | epoch: 10000 | loss: 0.07268 - acc: 0.9534 -- iter: 43/43  
--  
Tu: Hola  
BOTSITO: Hola, un gusto de verte, en que puedo ayudarte?  
Tu: tengo malestar  
BOTSITO: ¿aparte de la tos, tienes otro sintoma?  
Tu: solo tengo malestar  
BOTSITO: reposa una hora  
Tu: muchas gracias  
BOTSITO:  
Tu: gracias  
BOTSITO: cuidate y recuerda 'sana distancia'  
Tu: _
```