

## Charlando con la Inteligencia Artificial

---

En este primer laboratorio crearemos una Inteligencia Artificial donde el input será nuestra voz y la respuesta por parte de la I.A. será por sonido.

La IA seleccionada es Groq, debido a la facilidad que tiene usar su API gratuita!

---

## ¡Antes de empezar repasemos algunos conceptos!

---

Python tiene una palabra reservada para usar librerías que no trae por defecto, esta palabra es *import*, seguido del nombre nos dejará usar los métodos, funciones y variables que está traiga definidas en su interior. Un ejemplo ya visto es *import random* librería que nos permite generar números aleatorios.

Ahora empiezan los problemas, ya que no todas las librerías están por defecto en la instalación de python, hay algunas que debemos instalarlas nosotros y esto se hace poniendo el siguiente comando en la terminal (*Es muy importante que recuerden ponerlo en la terminal, ya que sino python no hará nada*)

### *pip install*

Seguido del nombre de la librería a instalar, nos descargará todos los módulos y paquetes que son necesarios para usarla!



Al usar *pip install* solo se nos descargará de forma local la librería, si compartimos el archivo de *python.py*, el mismo no se ejecutará ya que no está instalado en la computadora del cliente! La solución es al momento de compartir el ejecutable es instalarle la librería (más adelante lo veremos más a fondo)

---

## ¿Qué es una API?

---

Las APIs (Application Programming Interfaces, o Interfaces de Programación de Aplicaciones) son un conjunto de definiciones y protocolos que permiten que las diferentes aplicaciones de software se comuniquen entre sí.

---

API (Interface de Programación de Aplicaciones): Es una interfaz que define las interacciones entre diferentes piezas de software. Una API especifica cómo deben interactuar los componentes del software entre sí. Esto incluye qué funciones están disponibles, cómo se llaman esas funciones, qué parámetros se deben proporcionar y qué resultados se pueden esperar.

## Componentes de una API:

---

No se preocupen, las veremos más a fondo en próximos módulos! 🙌

- Endpoints
- Métodos HTTP
- Headers
- Body

---

## Ejemplo de uso de una API:

Imagina que tienes una aplicación de clima que necesita obtener los datos meteorológicos de diferentes ciudades. En lugar de tener una base de datos propia con esta información, tu aplicación puede usar una API pública de un servicio meteorológico. El proceso sería algo así:

- Tu aplicación envía una solicitud a la API del servicio meteorológico en un endpoint como <https://api.weather.com/v3/weather/forecast?city=Barcelona>.
- La API del servicio meteorológico procesa la solicitud, recupera los datos del clima actual de su base de datos y envía una respuesta con esta información.
- Tu aplicación recibe la respuesta y presenta los datos al usuario, como la temperatura actual, la previsión del tiempo, etc.

## Beneficios de las APIs

---

- **Modularidad:** Las APIs permiten dividir aplicaciones grandes en módulos más pequeños y manejables, que pueden ser desarrollados y mantenidos de forma independiente.
- **Reusabilidad:** El código que expone una API puede ser reutilizado por múltiples aplicaciones, reduciendo la duplicación de esfuerzos.
- **Interoperabilidad:** Las APIs permiten que diferentes sistemas y tecnologías se comuniquen entre sí, independientemente del lenguaje de programación en el que estén escritos.
- **Escalabilidad:** Facilitan el escalado de aplicaciones, ya que los servicios pueden ser distribuidos en diferentes servidores.

---

# Usando pip install en el Laboratorio

---

Para este caso necesitaremos instalar las siguientes librerías:

- pyttsx3
  - SpeechRecognition
  - pyaudio
  - groq
- 

Nuestros pasos para realizar el programa completo serán los siguientes:

**1- Realizar un programa que haga hablar a nuestra computadora utilizando la librería pyttsx3.**

---

```
#Un pequeño resumen del uso que le dimos!  
#Recorda tambien que podemos cambiar las voces y la velocidad, esta todo en el archivo que hic  
import pyttsx3  
engine = pyttsx3.init()  
engine.say("Hola ¿como estás?")  
engine.runAndWait()
```

---

**2- Realizar un programa que escuche nuestra voz y la transcriba a texto utilizando la librería SpeechRecognition**

---

```
#¿Como se usa SpeechRecognition?  
import speech_recognition as sr  
r =sr.Recognizer()  
  
with sr.Microphone() as source:  
    print("Hable: ")  
    audio = r.listen(source)  
    try:  
        text = r.recognize_google(audio, language='es-ES')  
        print(f"Dijiste: {text}")  
    except sr.UnknownValueError:  
        print("Lo siento, no he podido entender lo que has dicho.")
```

```
except sr.RequestError:
    print("Lo siento, ha habido un problema al intentar comunicarme con el servicio de Goo
```

### 3- Realiza un programa que permita consultar a la API de groq, mandandole un mensaje con un input y recibiendo por terminal la respuesta.

```
#Estructura Basica de Groq, esta se obtiene
#directamente de la documentacion oficial
#en https://console.groq.com/docs/quickstart
from groq import Groq

client = Groq(
    api_key="TU LLAVE API",#Modificamos esta linea para no usar os
)

chat_completion = client.chat.completions.create(
    messages=[
        {
            "role": "user",
            "content": "Explain the importance of fast language models",
        }
    ],
    model="llama3-8b-8192",
)

print(chat_completion.choices[0].message.content)
```

### Pasos para obtener tu API KEY

- 1: Ingresar a <https://console.groq.com/> e iniciar sesion con google.
- 2: Dirigirte en el panel izquierdo a "API Keys"
- 3: Utilizar el boton que dice "Create API Key", colocar un nombre identificativo para esta llave API y darle siguiente.
- 4: Muy importante **Copiar la API Key generada** y pegarla en un archivo .txt o en un comentario de python, ya que una vez generada no la podremos volver a copiar.
- 5: Modificar el codigo que copiamos de Quickstart para usar nuestra API sin necesidad de utilizar os

### 4- Realiza un programa que permita consultar la API de groq siendo el input nuestra voz y que nos de la respuesta por medio de sonido.

## Repasando algunas lineas de codigo:

---

La palabra reservada *as* sirve para asignarle el nombre a una biblioteca importada, en este caso le asignamos *sr* a *speech\_recognition* para que nos sea mas comodo trabajar.

```
import speech_recognition as sr
```

La palabra reservada *with* en Python se usa para simplificar el manejo de recursos que necesitan ser configurados y limpiados adecuadamente, como archivos, conexiones de red, o en este caso, el micrófono. La principal ventaja de usar *with* es que asegura que los recursos sean liberados correctamente después de su uso, incluso si ocurre un error durante su uso.

```
with sr.Microphone() as source:
```

El bloque *try except* se usa para manejar errores que pueden ocurrir durante la ejecución del código.

- *try*: Intenta ejecutar el código dentro de este bloque. Si todo funciona bien, se ejecuta sin problemas.
- *except sr.UnknownValueError*: Si ocurre un error del tipo *UnknownValueError*, significa que el reconocedor de habla no pudo entender el audio. En este caso, se ejecuta el siguiente bloque de código.
- *except sr.RequestError*: Si ocurre un error del tipo *RequestError*, significa que hubo un problema al intentar comunicarse con el servicio de Google. En este caso, se ejecuta el siguiente bloque de código.

```
try:
    text = r.recognize_google(audio, language='es-ES')
    print(f"Dijiste: {text}")
except sr.UnknownValueError:
    print("Lo siento, no he podido entender lo que has dicho.")
except sr.RequestError:
    print("Lo siento, ha habido un problema al intentar comunicarme con el servicio de Goo
```

¡El *try except* o conocido como *Try Catch* lo veremos más a fondo en proximas clases!

---

