

«Talento Tech»

Inteligencia Artificial

Clase 06



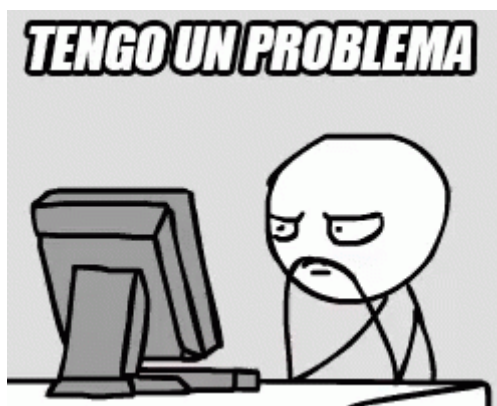
Clase N° 6 | Introducción a Streamlit y configuracion basica

- **Temario:**

- Librerías.
- Introducción a Streamlit, creación de cuenta y su instalación.
- Estructura básica de una aplicación Streamlit.
- Uso de elementos básicos de Streamlit: `st.title()`, `st.text_input()`, `st.text_area()`.

Librerías

Librerías en Python:



¿Qué son las librerías?

Muchas veces en programación nos puede llegar a ocurrir que nos encontramos estancados en el desarrollo de nuestro proyecto debido a que surgen problemas que no sabemos resolver. O en su defecto, requiere mucho código de gran complejidad. Como es algo tan normal que nos suceda, algunos desarrolladores se encargaron de que existan las librerías o módulos.

Estas librerías no son más que un código que ya está realizado por otros programadores. Contienen un conjunto de funcionalidades que nos pueden ayudar a:

- Generar una predicción de datos
- Mejorar los procesos de información, clasificación y regresión de datos
- Generar cálculos numéricos.
- Buscar resultados óptimos.

Sus acciones son infinitas ¿Se te ocurre algo más que pueda hacer una librería?

Importar Librerías

Gracias a Python podremos decidir si descargar toda una librería, o solo descargar unas funciones. Eso dependerá de nuestras necesidades.

Cuando se importa una librería completa, el intérprete de Python va a ejecutar todo el código que contiene, mientras que si solo se importan algunas partes de la librería, sólo se ejecutarán esas partes.

Importación completa de módulos (import)

import Libre : Ejecuta el código que contiene Libre y crea una referencia a él, de manera que pueden invocarse una función f que esté definida en él mediante la sintaxis Libre.f estaríamos aclarando que de la librería Libre, vamos a usar la función f

```
import random
numeros = random.sample(range(0, 100), 25)
print(numeros)
```

De esta forma nosotros importamos la librería random y usamos su función sample, que me permite elegir 25 números únicos entre el rango de 0 a 100.

import Libre as L : Ejecuta el código que contiene Libre y crea una referencia a él con el nombre L, de manera que pueden invocarse un objeto o función f definida en él mediante la sintaxis L.f. Esta forma es similar a la anterior, pero se suele usar cuando el nombre de la librería es muy largo y queremos acortarlo para limpiar el código.

```
import random as rm
numeros = rm.sample(range(0, 100), 25)
print(numeros)
```

En este código, hacemos lo mismo que el código anterior, pero a la librería random, le asignamos el alias de rm. Acortamos el nombre de la librería a una referencia de dos letras.

Importación parcial de módulos (from - import)

- `from L import f, g, ...` : Ejecuta el código que contiene L y crea referencias a las funciones f, g, etc, de manera que pueden ser invocados por su nombre. De esta manera para invocar cualquiera de estos objetos no hace falta precederlos por el nombre de la librería, basta con escribir su nombre.

```
from random import randint, choice
numeroRandom = randint(0, 90)
print(f'El número elegido hoy es el: {numeroRandom}')

colores = ["Rojo", "Verde", "Azul", "Amarilo", "Negro", "Purpura", "Naranja"]
color = choice(colores)
print(f'El color elegido es el: {color}')
```

```
El número elegido hoy e el: 8
El color elegido es el: Verde
```

Importamos de la librería random únicamente la función randint que genera un número aleatorio entre a y b, ambos inclusive y la función choice que selecciona un elemento aleatorio de una secuencia dad (como una lista, tupla, string, etc.).

- `from L import *` : Ejecuta el código que contiene L y crea referencias a todos los objetos públicos (aquellos que no empiezan por el carácter `_`) definidos en el módulo, de manera que pueden ser invocados por su nombre.

```
from random import *

numeroRandom = randint(0, 90)
```

```
print(f'El número elegido hoy es el: {numeroRandom}')
```



```
colores = ["Rojo", "Verde", "Azul", "Amarilo", "Negro", "Purpura", "Naranja"]  
color = choice(colores)  
print(f'El color elegido es el: {color}')
```



```
shuffle(colores)  
print(colores)
```

```
El número elegido hoy e el: 3  
El color elegido es el: Naranja  
['Amarillo', 'Azul', 'Rojo', 'Naranja', 'Verde', 'Purpura', 'Negro']
```

Vemos que importamos todas las funciones de random, por lo que usamos cualquier función de su librería sin referirnos a la librería, un alias o una función específica.. En esta parte, además de las funciones vistas, usamos la función **shuffle()** que mezcla aleatoriamente los elementos de una lista modificando así, la lista original.

🦉🦉 ¡A tener en cuenta!

Cuando se importen módulos de esta manera hay que tener cuidado de que no haya coincidencias en los nombres de funciones, variables u otros objetos que tengamos en nuestro código.

También se recomienda que si vas a usar una sola función de un módulo, no importes todas las funciones, sino la que necesites. Es importante la exclusividad cuando tenemos un código muy dinámico.

import os



from os import *



En este caso Python trae varios módulos básicos conocidos como librerías predefinidas . Dos de las más conocidas son:

- **datetime**: Funciones para fechas y tiempos.
- **sys**: Librería muy útil para interactuar con el sistema operativo y el entorno de ejecución de Python.

Podemos conocerlas para practicar.

¿Qué pasa si queremos importar una librería externa?

Si bien Python nos trae muchas librerías incorporadas, cómo las que vimos recientemente. Seguramente vamos a necesitar algunas funcionalidades extras. para eso vamos a recurrir a librerías externas.

Al momento de querer usarlas, vamos a tener que seguir una serie de pasos para poder instalarlas.

Para eso hacemos uso de una librería que viene en Python desde la versión 3.4 llamada “pip”. (¡Así es, usamos una librería para instalar otra!)

Por lo que vamos a utilizar el comando pip, con el que vamos a poder instalar cualquier librería externa. Ahora bien, esta instalación también va a depender de dónde estemos situados.

Por ejemplo, si estamos desde la terminal de Python, y queremos instalar la librería Numpy, vamos a hacer lo siguiente:

```
python -m pip install libreríaQueQueramosInstalar
```

Pero si estamos en nuestro editor de código, solo debemos hacer::

```
pip install libreríaQueQueramosInstalar
```

De este modo vamos a poder instalar cualquier librería en nuestro proyecto, lo que nos permitirá potenciar nuestro trabajo con funciones creadas por la comunidad.

Librerías más utilizadas:

Numpy: es una de las librerías más importantes para trabajar con información. Nos permite trabajar con muchos datos de forma multidimensional, implementando matrices. Además posee funciones matemáticas de alto nivel.

Matplotlib: Esta librería es una de las más usadas ya que nos permite generar gráficos de alta calidad en dos dimensiones, como también publicarlos online o en papel.



Streamlit: Esta librería está de moda por su facilidad y función: nos permite mostrar en un navegador y compartir con otras personas en pocos minutos, todo lo que venimos trabajando con Python.

Librerías que utilizaremos.

Para nuestro proyecto nos enfocaremos únicamente en dos librerías. Por un lado usaremos streamlit, vista anteriormente y por otro, para el sistema de IA, usaremos la librería groq. La librería groq, nos proporcionará lo necesario para construir nuestro chatbot.

Groq.

Groq es un chatbot que podemos usar haciendo click [aca](#), como puede ser Chat GPT, Gemini o Claude. Incluso mejor. ¿Cuál es el potencial de Groq?

Groq es una empresa fundada en 2016 por Jonathan Ross, ex ingeniero de Google. La compañía se destaca por su Tensor Streaming Processor (TSP), un chip diseñado específicamente para tareas de IA que promete rendimientos excepcionales, mayores a sus competidores. Con ellos cualquier chatbot puede responder mucho más rápido, pero además con un coste económico menor.

Afirma ofrecer tiempos de respuesta extremadamente rápidos, especialmente en la inferencia de modelos de lenguaje (LLMs), superando en velocidad a muchas soluciones existentes.

Esta tecnología encuentra aplicación en diversos campos que requieren procesamiento de IA de baja latencia y alto rendimiento, como el procesamiento de lenguaje natural, la visión por computadora, y análisis de datos en tiempo real, siendo particularmente útil en industrias como la financiera, la salud y la investigación científica. En el competitivo mercado de aceleradores de IA, Groq se enfrenta a gigantes tecnológicos como NVIDIA y Google, así como a otras startups especializadas en chips de IA, diferenciándose por su arquitectura única y su enfoque en la eficiencia y velocidad de procesamiento, lo que podría redefinir los estándares de rendimiento en el campo de la inteligencia artificial.

Además, a diferencia de otras empresas que tiene chatbot, Groq nos ofrece su api pública con sus modelos para que pueda ser utilizado en tiempo real desde un programa externo.



Groq trae su propia librería que nos permitirá utilizar su motor de inteligencia artificial mediante una API.

¿Qué es una API?

API significa Application Programming Interface (Interfaz de Programación de Aplicaciones). Es un conjunto de definiciones y protocolos que permiten que dos aplicaciones se comuniquen entre sí. Las APIs facilitan la interacción entre diferentes sistemas, permitiendo que un software acceda a las funcionalidades o datos de otro.

Ejemplo

Pensá en una API como un camarero en un restaurante. Tú (el cliente) pides algo del menú (el servicio que quieres). El camarero (la API) lleva tu pedido a la cocina (el servidor) y luego te trae la comida (la respuesta del servidor). No necesitas saber cómo se prepara la comida; podrías interactuar solamente con el camarero para obtener lo que deseas.

Usando la API de groq

Vamos a crearnos una cuenta en groq.cloud donde buscaremos una clave para tener acceso a su API. Para usarla tenemos que estar logueados en su página por lo que vamos a seguir este paso a paso que nos va a permitir crearnos una cuenta de y guardar su clave de acceso.

PASO A PASO DE SU INGRESO

```
<iframe style="border: 1px solid rgba(0, 0, 0, 0.1);" width="600" height="500"
src="https://on.driveway.app/guides/5ErKBeL/embed" allow="clipboard-write"
allowfullscreen></iframe>
```

Vamos a mandarnos nuestra clave API a nuestro mail. Nos tenemos que asegurar de que seamos los únicos en tener acceso.

Instalando la librería

Para hacer uso de la librería de groq, lo primero que tenemos que hacer es instalar la librería en nuestro sistema. Para eso hacemos la instalamos igual que con streamlit.



```
python -m pip install groq
```

Ya tenemos las librerías necesarias para crear nuestro chatbot con streamlit y Groq! Vamos a comenzar a darle estilo a nuestro proyecto.

Streamlit

Streamlit es una biblioteca de Python de código abierto que permite crear aplicaciones web interactivas y personalizadas de manera rápida y sencilla. Es muy popular entre científicos de datos y desarrolladores porque permite transformar scripts de python en aplicaciones web sin necesidad de conocimientos avanzados en desarrollo web. Además de que te permite compartir tu proyecto alojando tu aplicación en su propio hosting.

[Ejemplos de aplicaciones con streamlit](#)

Streamlit se utiliza principalmente para:

- Crear dashboards interactivos.
- Visualizar y compartir análisis de datos.
- Desarrollar prototipos de aplicaciones de machine learning e IA.
- Construir interfaces de usuario para modelos de machine learning.

Cómo crear una cuenta en Streamlit

```
<iframe style="border: 1px solid rgba(0, 0, 0, 0.1);" width="600" height="500"
src="https://on.driveway.app/guides/xLRr70L/embed" allow="clipboard-write"
allowfullscreen></iframe>
```

Tu primer código con Streamlit

Para comenzar a utilizar streamlit será necesario instalarlo en nuestra computadora. Para eso lo instalaremos como cualquier librería externa:

```
python -m pip install streamlit
```

La documentación de streamlit está muy bien documentado todas sus funciones. Nosotros vamos a ver algunas nomás, pero te recomendamos que hagas [click acá](#) para ver su sitio web. Vamos a escribir un código básico para crear una aplicación simple que permita al usuario ingresar su nombre y mostrar un saludo personalizado.

Vamos a usar las siguientes funciones de streamlit:

- set_page_config:** Configura parámetros de la página, como el título, el icono y el diseño (si quiero que este centrado o a la izquierda)
- title:** Muestra un título en la aplicación.
- text_input:** Crea un cuadro de entrada de texto donde el usuario puede escribir.
- button:** Crea un botón que el usuario puede presionar.
- write:** Muestra texto, números, datos y otros elementos en la aplicación.

Lo primero que vamos a hacer es importar en nuestro archivo .py la librería instalada.

```
import streamlit as st
```

Luego vamos a usar la función **title** de la librería streamlit que importamos como st, para agregarle un título a nuestra web.

```
# Título de la aplicación  
st.title("Mi primera aplicación con Streamlit")
```

Si queremos ver si nuestra página va funcionando, podemos adelantarnos hasta #mostrando mi aplicación para ver el resultado en la web.

Una vez que creamos el título de mi sitio web, vamos a distinguir a nuestra página con un nombre para la pestaña y un ícono que la diferencie del resto de pestañas que podamos tener abiertas.

Para eso usaremos la función **set_page_config** de streamlit para pasarle el título a la página y el ícono a la página. Esta función de streamlit tiene que ser la primera función que lea la librería, por lo tanto tiene que estar por encima del título y por debajo de las librerías importadas.

```
import streamlit as st

# ESTO ES NUEVO
st.set_page_config(page_title="Mi chat de IA", page_icon="🐼")

# Título de la aplicación
st.title("Mi Primera Aplicación con Streamlit")
```

Luego usaremos la función `text_input`. Esta función guardará en la variable, en nuestro caso nombre, el valor que ingrese el usuario por pantalla.

```
# Entrada de texto

nombre = st.text_input("¿Cuál es tu nombre?")
```

Ahora con la ayuda de un condicional y el uso de un boton vamos a mostrar un mensaje.

```
# Botón para mostrar el saludo

if st.button("Saludar"):

    st.write(f"¡Hola, {nombre}! gracias por venir a Talento Tech.")
```

Nota! 😊 La función `write` sirve para renderizar en pantalla cualquier cosa que quiera mostrar. Todo va a salir por el navegador con su particularidad. En el caso de nuestro mensaje, podríamos usar también la función `subheader` (`st.subheader`), que te permite agregar un subtítulo. Este recibe tres parámetros que hacen que se distinga el titulo y subtítulo del resto de contenido de la página. Te recomiendo ver la docu [ACA!](#)

El código completo tiene que verse de esta forma:

```
import streamlit as st

st.set_page_config(page_title="Mi chat de IA", page_icon="🐼")

st.title("Mi primera aplicación con Streamlit")

nombre = st.text_input("¿Cuál es tu nombre?")

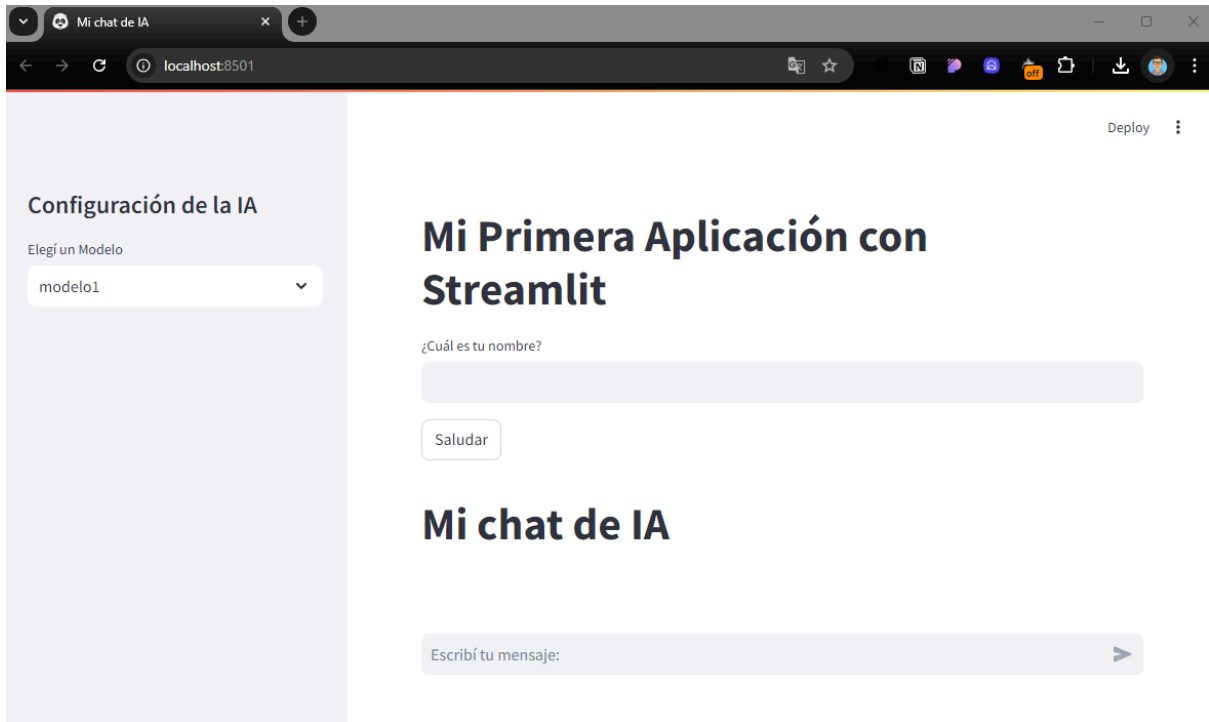
if st.button("Saludar"):
    st.write(f"¡Hola, {nombre}! gracias por venir a Talento Tech")
```

Mostrando mi aplicación en el navegador

Para mostrar mi aplicación en la web, será necesario que abra la terminal, desde abajo o buscando en la barra superior de visual studio code, view => y luego terminal. En la terminal, me aseguro de estar en el mismo lugar que mi archivo, y ejecutar la siguiente línea de código:

```
python -m streamlit run nombreDeMiArchivo
```

Si mi archivo se llamada app.py, entonces reemplazo nombreDeMiArchivo por app.py



Si hicimos correctamente el código, cuando le escriba un nombre y presione saludar, saldrá un mensaje de bienvenida! Recordá que el boton viene por defecto en falso, y para pasarlo a verdadero (o True) será necesario presionarlo.

Mi Primera Aplicación con Streamlit

¿Cuál es tu nombre?

Lionel

Saludar

¡Hola, Lionel! gracias por venir a Talento Tech

Mi primera aplicación con streamlit

Además de las funciones que vimos de Streamlit, la librería nos ofrece tres herramientas más que darán sentido a nuestro chatbot. Podemos optar por dejar el código anterior o crear un nuevo archivo.py. Usaremos las siguientes funciones de streamlit y las organizaremos para que nuestro chatbot tenga forma.

1. **sidebar():** El sidebar en Streamlit permite agregar widgets y otros elementos de la interfaz de usuario en una barra lateral. Esto es útil para mantener la interfaz principal más limpia y organizada. Si quiero crear un título en mi barra lateral, lo que voy a escribir es lo siguiente:

```
st.sidebar.title("título de mi barra lateral")
```

Si quiero escribir un mensaje en mi barra lateral, el código me quedaría de esta forma:

```
st.sidebar.write("texto en mi barra lateral.")
```

2. **selectbox()** : Esta función de Streamlit nos ayuda a crear un cuadro desplegable que permite a los usuarios seleccionar una opción de una lista.

Sintaxis

```
st.selectbox(label, options, index)
```

Label: Texto que describe el cuadro desplegable.

Options: Lista de opciones entre las que el usuario puede elegir, por lo que genera una lista.

Index: Índice de la opción seleccionada inicialmente (opcional).

3. **chat_input():** Esta función no ayuda a crear un campo de entrada de texto especializado para aplicaciones de chat, donde los usuarios pueden escribir mensajes.

Sintaxis :

`st.chat_input(label)`

Label: Texto que describe el campo de entrada.

Lo que haremos será **crear una barra lateral con la función sidebar de streamlit**, donde le agregaremos un título con el mensaje: “Configuración de la IA” y **mostraré con la función selectbox de streamlit**, también dentro de mi barra lateral, las opciones disponibles de los modelos. Podemos comenzar por generar una variable llamada modelos que sea una lista de tres opciones de modelos. Comencemos por generar la variable de esta forma (luego cuando usemos groq, sabremos cuales modelos vamos a utilizar):

```
# Configuración básica
MODELOS = ['modelo1', 'modelo2', 'modelo3']
```

Agregamos la configuración básica de nuestra aplicación con la función set_page_config:

```
st.set_page_config(page_title="Mi chat de IA", page_icon="🤖",
layout="centered")
```

Luego voy a crear una función llamada **configuracion_pagina** (o el nombre que quieras, recordá que el nombre tiene que ser representativo de lo que va a hacer la función).

Esta función, generará el título principal de la aplicación, creará el sidebar con un título y generará un selectbox, para seleccionar uno de los modelos. Una idea puede ser generar la función de esta forma:

```
def configurar_pagina():
    # Agregamos un título principal a nuestra página
    st.title("Mi chat de IA")
    st.sidebar.title("Configuración de la IA") # Creamos un sidebar con un
    título.
    elegirModelo = st.sidebar.selectbox('Elegí un Modelo', options=MODELOS,
    index=0)
    return elegirModelo
modelo = configurar_pagina()
```



Luego guardamos en una variable llamada `modelo`, el modelo que se genera cuando configuro mi aplicación.

Ya tenemos la estructura de nuestra aplicación formada!

Elegimos un modelo en particular de los que están disponibles, igual que en groq.com.

Ahora vamos a crear una variable llamada `modelo` que guarde los modelos asignados en mi función `configurar_pagina()` y también generar una variable llamada `mensaje` que guarde el mensaje que mandemos por el navegador.

```
modelo = configurar_pagina()
mensaje = st.chat_input("Escribí tu mensaje:")
```

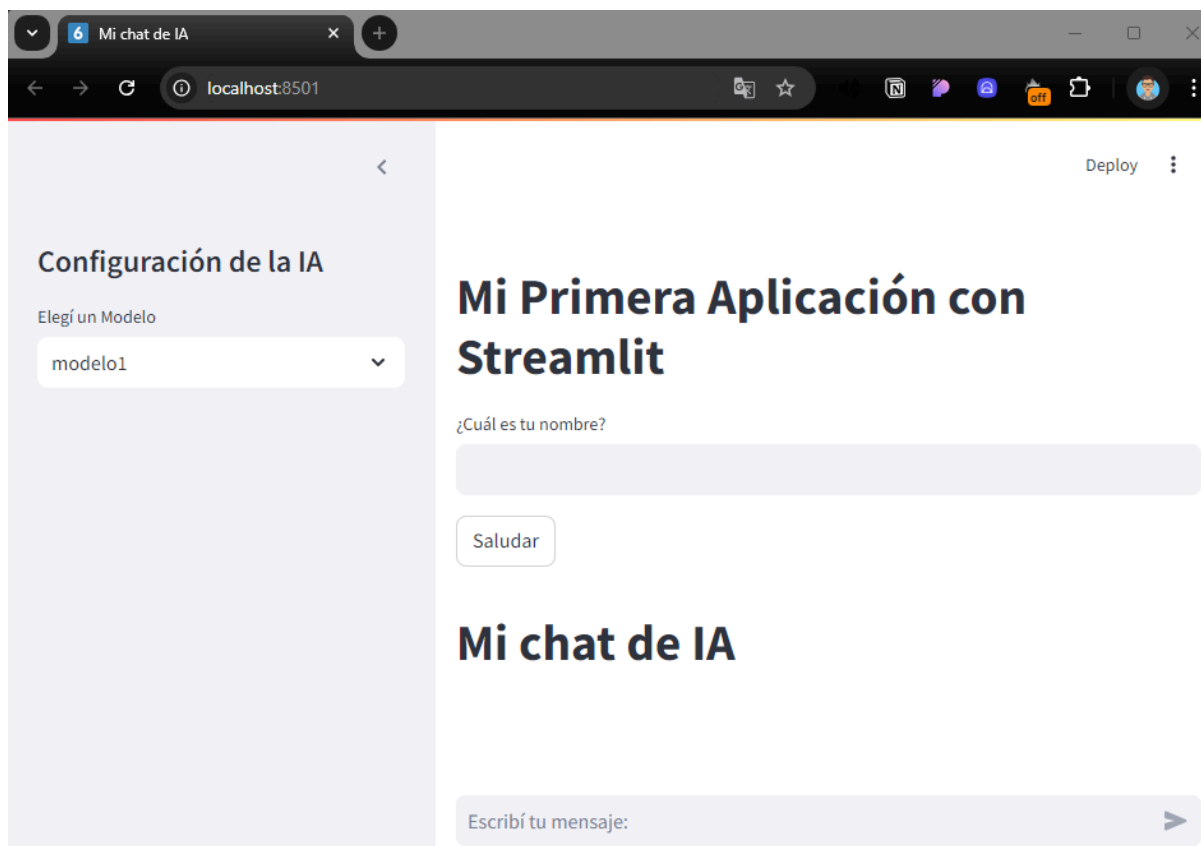
Como generé la estructura de mi aplicación dentro de una función, será necesario que llamemos (o ejecutemos) a esa función para que nuestra aplicación cobre vida.

Recordá (nuevamente) que puedes ver como va tu aplicación ejecutando en la terminal el siguiente comando:

```
python -m streamlit run nombreDeMiArchivo
```

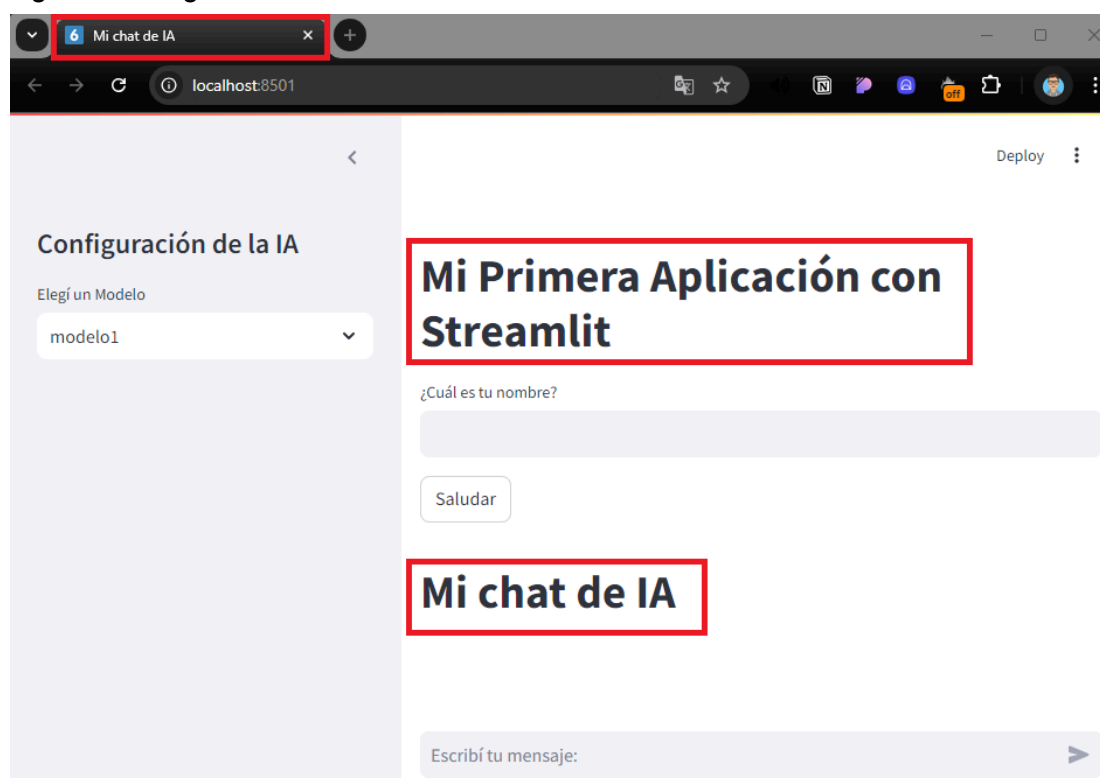
Si quiero ver si lo que escribo en mi cuadro de texto está en mi programa, podría usar la función `print`, para imprimir mi variable `mensaje` por la terminal.

Cómo debería quedar:



Desafío N° 6:

Este desafío obligatorio consiste en modificar lo que está dentro del recuadro rojo en la siguiente imagen:



Pistas:

- 1.- En la parte superior izquierda, se modifica en la función **st.set_page_config**.
- 2.- Los otros dos recuadros que están en rojo, se modifican ubicando los **st.title**.



Buenos Aires
aprende
Agencia de Actividades para el Futuro

BA Buenos
Aires
Ciudad