

Documentación funcional y técnica

Versión: 1

TP inicial - Laboratorio de Construcción de Software-Proyecto Profesional I

Alumnos:

- Hirschfeld Agustina Camila (camilahirschf@gmail.com)
- Veron Adrián (adrianveron4@hotmail.com)
- Cabrera Juan Manuel (amnauju5@gmail.com)

Profesores:

- Juan Carlos Monteros (jcmonteros@campus.ungs.edu.ar)
- Evelyn Aragon (eve_aragon@hotmail.com)

Revisiones del documento

Fecha	Revisión	Descripción
24/03/2024	1	Primera versión del documento.

Sistemas / Plataformas que intervienen

El objetivo previamente mencionado pretende lograrse mediante el uso del lenguaje de programación Python como herramienta principal para el desarrollo, combinado con el uso de la API “Openweathermap”.

Nombre	Descripción
Python (Lenguaje de programación)	Lenguaje de programación multiparadigma(Orientado a objetos, programación imperativa y programación funcional). Los usos de Python pueden variar entre aplicaciones web, desarrollo de software, ciencia de datos, machine learning (ML), etc.
Openweathermap (API)	API que proporciona datos meteorológicos globales. Entre los datos ofrecidos por el servicio se incluyen datos meteorológicos actuales, pronósticos, predicciones inmediatas y datos meteorológicos históricos.
Deep Translator (API)	Permite la traducción de un texto a diversos idiomas con el objetivo de brindar una mejor experiencia al usuario.
Tkinter	Es una librería del lenguaje de programación Python que se utilizó para crear y desarrollar el chatbot en una aplicación de escritorio.
Sklearn	Es una biblioteca de Python para el aprendizaje automático que proporciona herramientas para la construcción de modelos de aprendizaje automático, incluyendo la extracción de características, la clasificación, la regresión, la reducción de dimensionalidad, entre otros. En el contexto del chatbot, se hace referencia a TfidfVectorizer y cosine_similarity de sklearn, que son dos componentes clave para el procesamiento de texto y la comparación de similitud entre textos.

Descripción:

El objetivo del desarrollo es lograr un chatbot cuya función principal sea brindar información acerca del clima según ubicación geográfica.

El usuario tendrá la posibilidad de realizar distintos tipos de consultas (relacionadas al tópico mencionado) incluyendo pronósticos extendidos por zona geográfica, recomendaciones en base a la actividad planeada por el usuario, Calidad del aire y demás.

Potenciales Mejoras Identificadas:

A)

La idea de implementar un sistema de login con huella digital en un smartphone Android implica combinar varias tecnologías y lenguajes de programación. En primer lugar, es necesario utilizar Android Studio, que es el entorno de desarrollo integrado (IDE) oficial para la creación de aplicaciones Android. Esto significa que, sí o sí, debemos trabajar con Java o Kotlin, los dos lenguajes de programación principales para el desarrollo de aplicaciones Android.

Para integrar el uso de reconocimiento a través de huellas digitales en una aplicación Android, se deben utilizar las API proporcionadas por el sistema operativo, como FingerprintManager en versiones anteriores de Android o BiometricPrompt a partir de Android 9 (Pie). Estas API permiten acceder al hardware del dispositivo para autenticar al usuario mediante su huella digital.

En cuanto a la implementación de machine learning en el chatbot para responder consultas sobre el clima, la implementación de esta funcionalidad puede requerir un enfoque diferente. Aquí es donde entra en juego la necesidad de convertir el modelo de machine learning en una API accesible desde la aplicación móvil. Para lograr esto, se puede utilizar Flask, un framework de Python para desarrollo web, para crear una API que permita utilizar en la aplicación de android, las funcionalidades del modelo de machine learning.

Por otro lado, si optamos por avanzar por el camino de convertir todo lo referido a Machine Learning en una API, el uso de Postman será de gran ayuda, pues Postman es una herramienta que permite probar y depurar APIs, lo que facilita el proceso de desarrollo y prueba de la API.

En resumen, el desarrollo de una aplicación Android con un login con huella digital y un chatbot para consultas sobre el clima implica un enfoque variado que combina el uso de Android Studio con Java/Kotlin para el desarrollo de la interfaz de usuario y la funcionalidad de la huella digital, Flask para convertir el modelo de machine learning en una API accesible, y Postman para probar y consumir esta API desde la aplicación móvil. Este enfoque integrado permite crear una aplicación robusta y funcional que aprovecha diversas tecnologías para proporcionar una experiencia óptima al usuario.

B)

- ☐ Añadir más preguntas y respuestas para optimizar el razonamiento del chatbot al momento de dar una respuesta acorde a la entrada del usuario.
- ☐ Implementar un GUI más estilizado con el fin de mejorar la experiencia del usuario.
- ☐ Poder obtener información detallada específica respecto al clima.

- ☐ Poder especificar la cantidad de días o el formato que se desea al consultar el pronóstico de los próximos días.

Volumetría:

La cantidad de consultas a las que puede responder el bot se encuentra limitada a un número fijo, es decir, la cantidad de consultas a las que puede responder no es variable entre una actualización y otra.

El tiempo de respuesta de algunas solicitudes al bot se encuentra limitado a la api Openweathermap lo que puede ocasionar demoras al momento de obtener un repuesto e incluso causar el crasheo del software.

Por otro lado, las palabras y/o oraciones comprendidas por el Bot para una misma consulta pueden variar en base a la redacción del usuario.

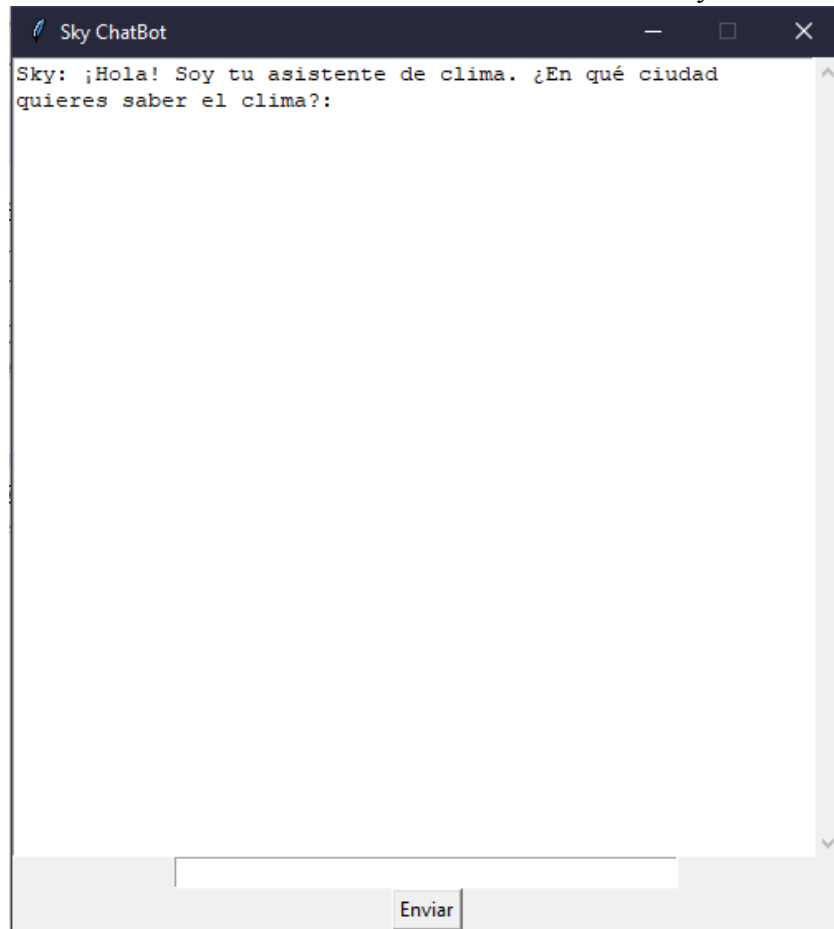
Notas Adicionales:

Es importante señalar que, a pesar de encontrarse en una etapa temprana de desarrollo, el programa ha logrado implementar y ejecutar las funcionalidades básicas necesarias para su operatividad, alcanzando niveles de eficiencia y calidad notablemente altos.

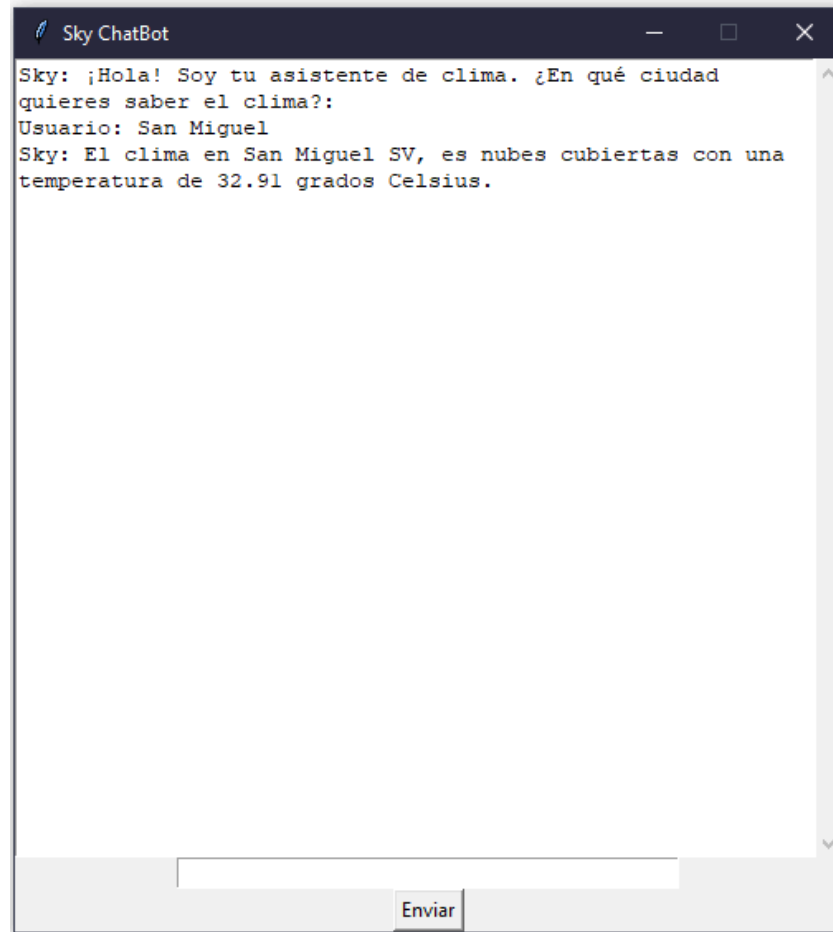
Cabe destacar que la experiencia de los usuarios de este software no se encontrará afectada por error alguno más allá de una mala interpretación que pueda generar el algoritmo del machine learning o la demora para interactuar con los servidores de la API "Openweathermap".

Funcionalidades:

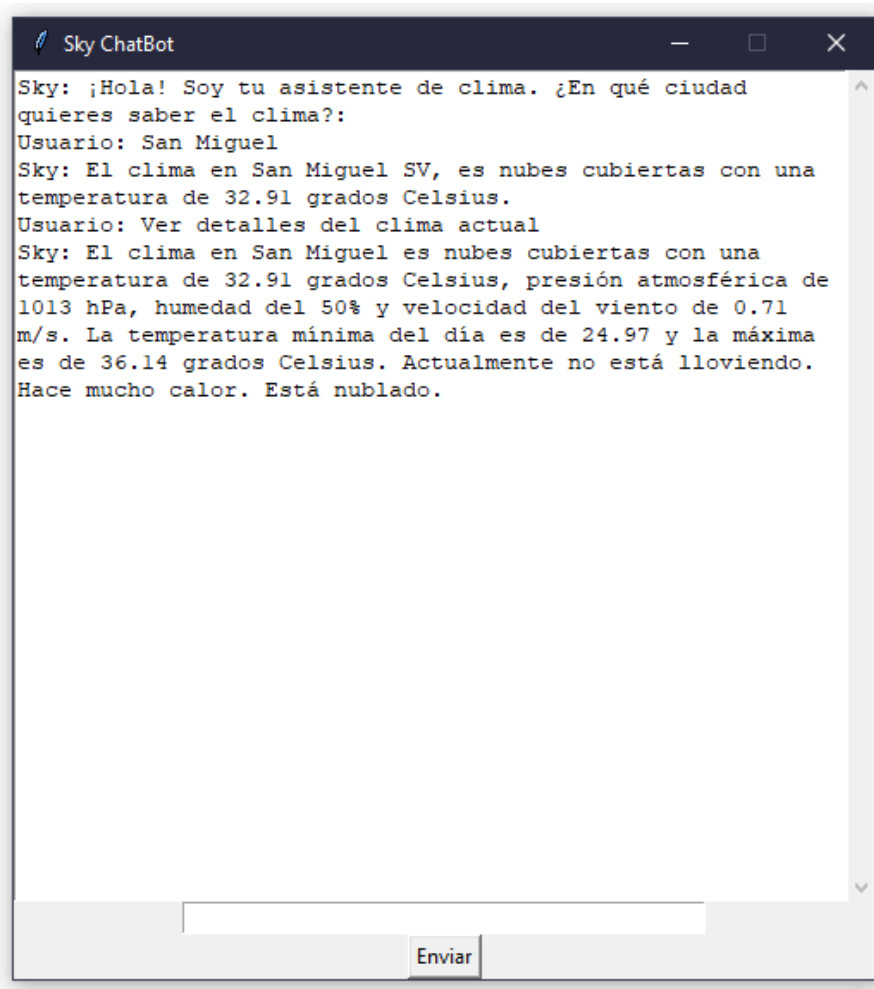
- El software está pensado para ejecutar el script controller.py el cual abrirá una ventana que le permitirá al usuario interactuar con el chatbot llamado “Sky”.



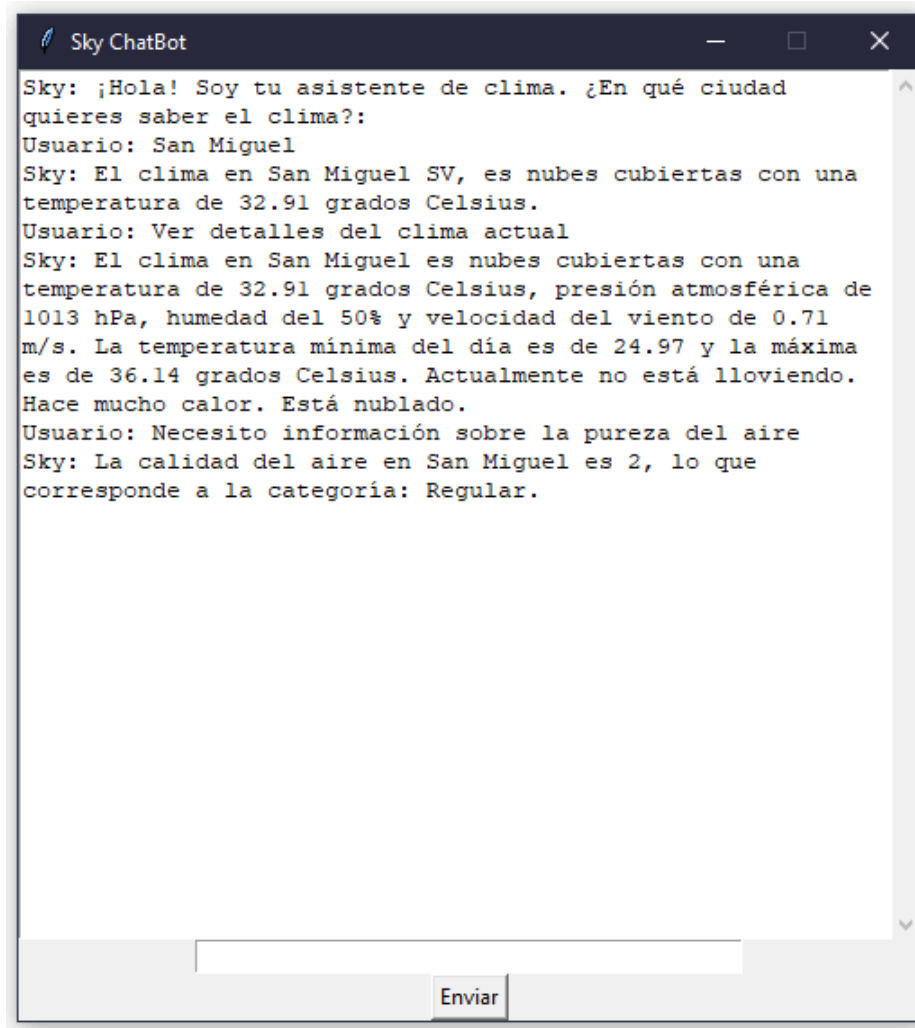
- Notemos que en el cuadro inferior de la pantalla el usuario será libre de introducir cualquier tipo de texto y este recibirá una respuesta acorde a la entrada. Siendo la primera pregunta una ciudad y el bot responderá en función de la entrada del usuario.



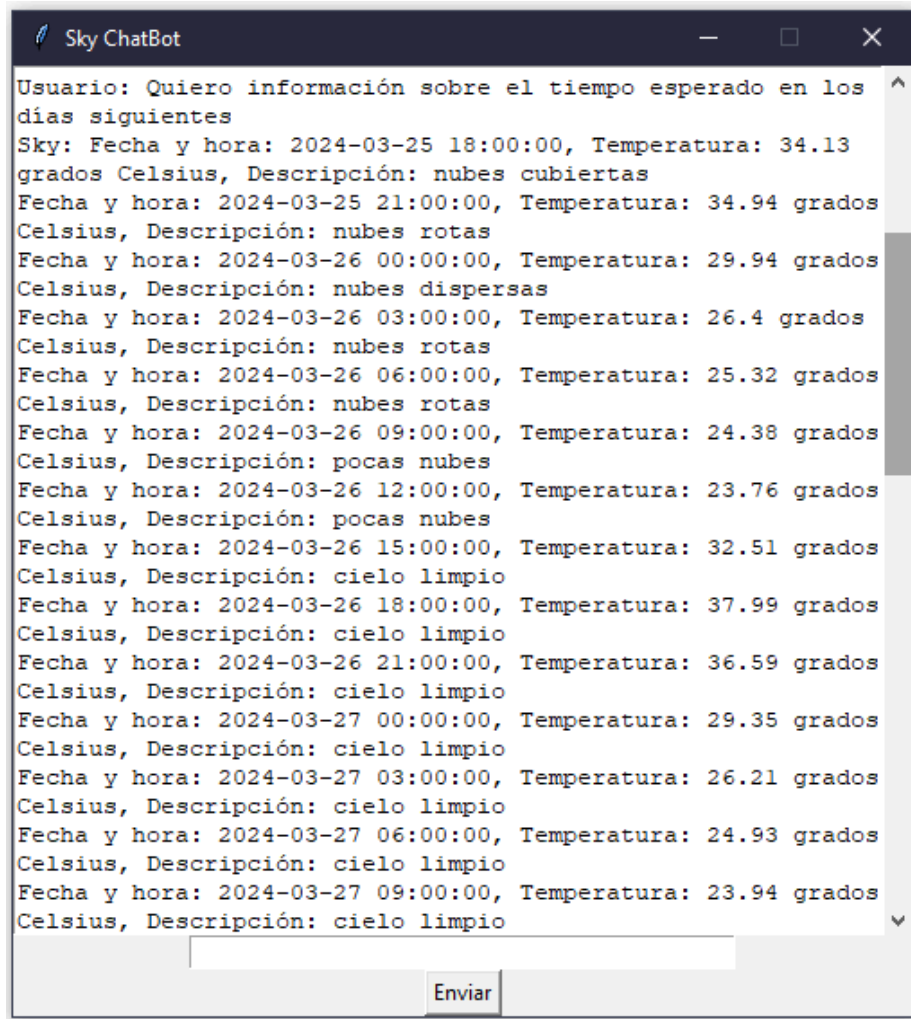
- Ahora al bot se le podrán realizar diversas preguntas que tengan relación con asuntos climáticos tales como:
Información detallada del clima actual:



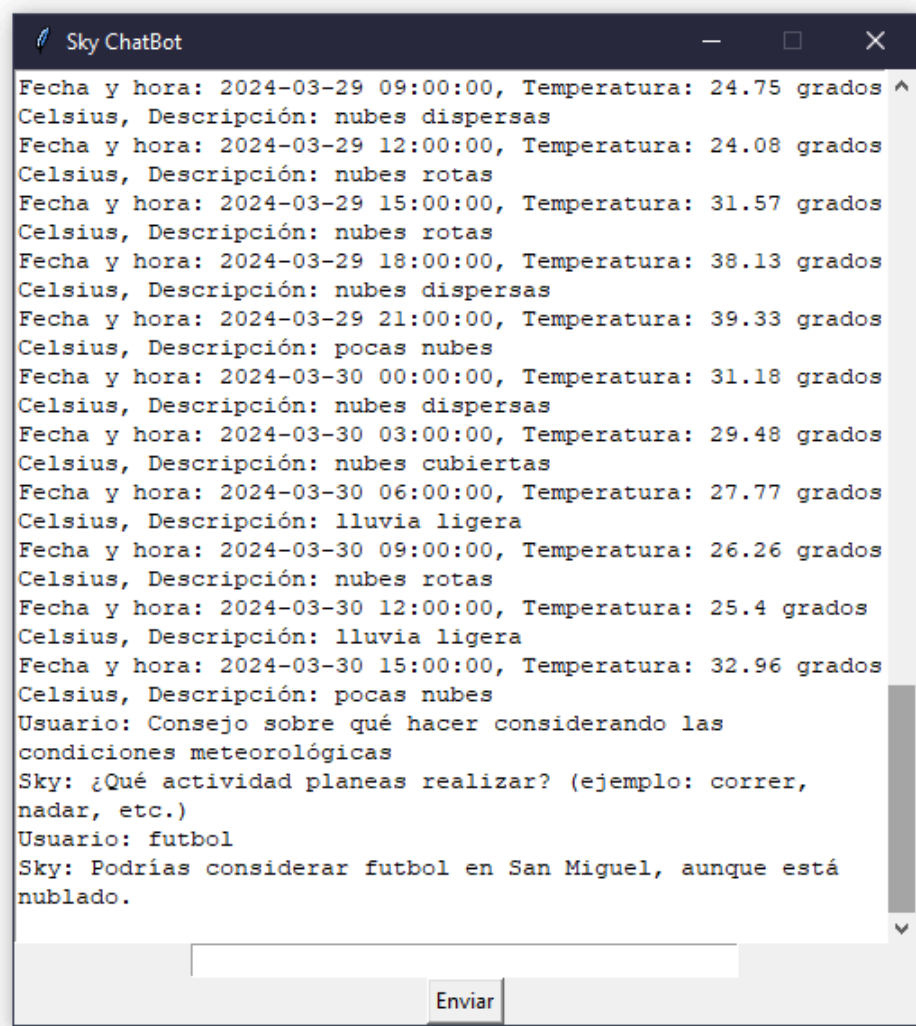
- También se le puede consultar al chatbot por la calidad del aire:



- Obtener información sobre el clima en los próximos días:



- Recomendación de actividad según el clima:



- Cambiar de ubicación, obteniendo nueva información acorde:

