Instituto tecnológico de Costa Rica

Unidad de computación

Primer Proyecto Programado

Mi asistente virtual

Integrantes del grupo:

Omar Steven Chacón Porras

María Alejandra Ugalde Rodríguez

Shekinah María Jiménez Sáenz

Sede San Carlos

21 de Abril del 2025

**Introducción**

En este proyecto se va a desarrollar un asistente virtual utilizando Python, con el propósito de manejar y consultar un historial de conversaciones entre usuarios y asistentes virtuales. Este proyecto surge de la necesidad de optimizar la interacción y la gestión de datos en estas conversaciones, haciendo más fácil tareas como registrar nuevas charlas, buscar palabras clave y generar resúmenes.

La importancia de los asistentes virtuales ha crecido significativamente en los últimos años, impulsada por avances en la inteligencia artificial y el procesamiento del lenguaje natural (Clon Geek, 2025; Juscilene Vasquez, 2025). Estos sistemas no solo automatizan tareas repetitivas, sino que también mejoran la experiencia del usuario mediante respuestas rápidas y precisas (SEIDOR, 2024; Tellaeche, 2024). Los asistentes virtuales representan una mejora tecnológica, y al enfocarnos en el área académica, resultan de gran ayuda no solo para los docentes, sino también para los estudiantes. Pueden colaborar en tareas como recordatorios de actividades de aprendizaje y facilitar el acceso a contenidos, contribuyendo así a una educación más dinámica.

Este proyecto está relacionado con las nuevas tecnologías, ya que los asistentes virtuales se están usando cada vez más en diferentes áreas, como la atención al cliente, la organización de tareas y el soporte técnico (Clon Geek, 2025; Juscilene Vasquez, 2025). Además, en este trabajo se van a usar herramientas de programación y estructuras de datos que permitan que el sistema sea más eficiente (SEIDOR, 2024; Tellaeche, 2024).

**Análisis del Problema**

Descripción de la Situación:

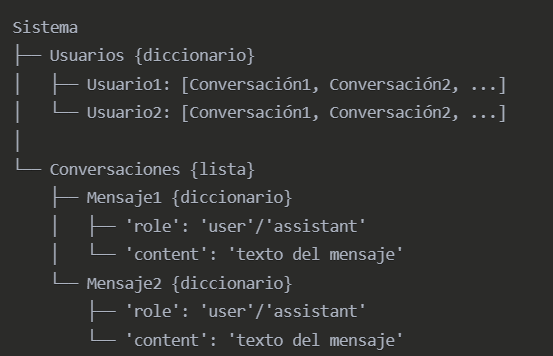
El proyecto "Mi asistente virtual" busca desarrollar un sistema que gestione el historial de conversaciones entre usuarios y asistentes virtuales como ChatGPT. Actualmente, los usuarios que interactúan con asistentes virtuales carecen de una herramienta unificada que les permita registrar, consultar y analizar sus conversaciones previas de manera estructurada y persistente.

Requerimientos Principales

Del análisis de la situación, se destacan los siguientes requerimientos clave:

1. Gestión de Usuarios: Sistema de registro y validación de usuarios.
2. Administración de Conversaciones: Capacidad para registrar nuevas interacciones usuario-asistente.
3. Consulta de Historial: Visualización cronológica de conversaciones previas.
4. Búsqueda por Palabras Clave: Filtrado de conversaciones basado en términos específicos.
5. Generación de Resúmenes: Capacidad de sintetizar conversaciones en 50 palabras.
6. Almacenamiento Persistente: Guardar las conversaciones en archivos de texto.

Con estos requerimientos principales será desarrollado la siguiente estructura de datos:

Figura 1. Gráfico generado por inteligencia artificial. Fuente: Creado con Claude AI, captura de pantalla por la autora (2025).

**Metodología de Solución**

Para desarrollar el sistema, se va a usar un enfoque modular que divida las diferentes funciones en partes independientes.

Módulo de Usuarios: Responsable de crear, validar y administrar a los usuarios.

Módulo de Conversaciones: Se ocupa de generar y administrar las conversaciones.

Módulo de Consultas: Implementación de los filtros y búsquedas dentro del historial.

Módulo de Almacenamiento: Responsable de mantener los datos guardados utilizando archivos de texto.

Módulo de Interfaz: Desarrollo de la interfaz de usuario (consola o gráfica).

**Solución del problema**

Para desarrollar el sistema de gestión de conversaciones con asistentes virtuales, se empleó una arquitectura modular en Python que facilita la interacción entre el usuario y el asistente, además de almacenar las conversaciones y proporcionar diversas opciones de consulta. El proyecto se fundamentó en principios de programación estructurada y orientada a objetos, garantizando así que el sistema sea sencillo de mantener y ampliar en el futuro.

**Etapas de la Solución**

1. Diseño de la Estructura de Datos: La estructura de datos implementada es jerárquica:

- Se utiliza un diccionario principal “usuarios” donde cada clave es el nombre de usuario

- El valor asociado a cada usuario es una lista de conversaciones o "registros".

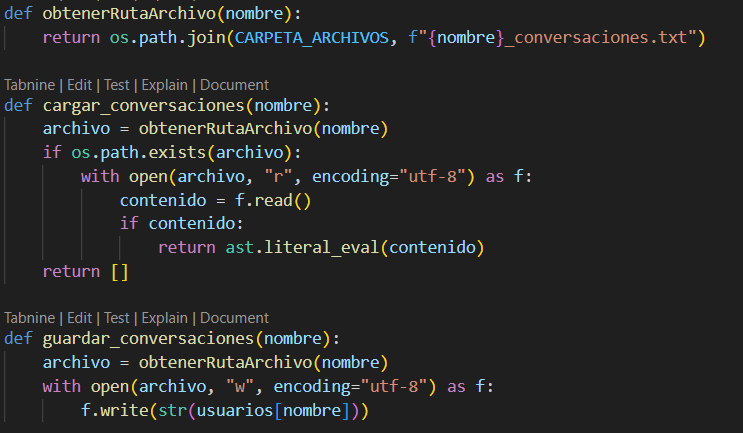
- Cada registro es un diccionario con dos claves: "fecha" (cadena de texto con formato DD-MM-AAAA) y "mensajes" (lista).

- La lista "mensajes" contiene diccionarios que representan cada interacción, con dos atributos: "role" (que puede ser "usuario" o "asistente") y "content" (el texto del mensaje).

Esta estructura permite acceder eficientemente al historial de un usuario específico y mantener organizadas cronológicamente las conversaciones con su respectivo contenido, facilitando operaciones como búsquedas y generación de resúmenes.

2. Gestión de Persistencia

El sistema implementa un mecanismo de persistencia basado en archivos de texto para almacenar las conversaciones de cada usuario:

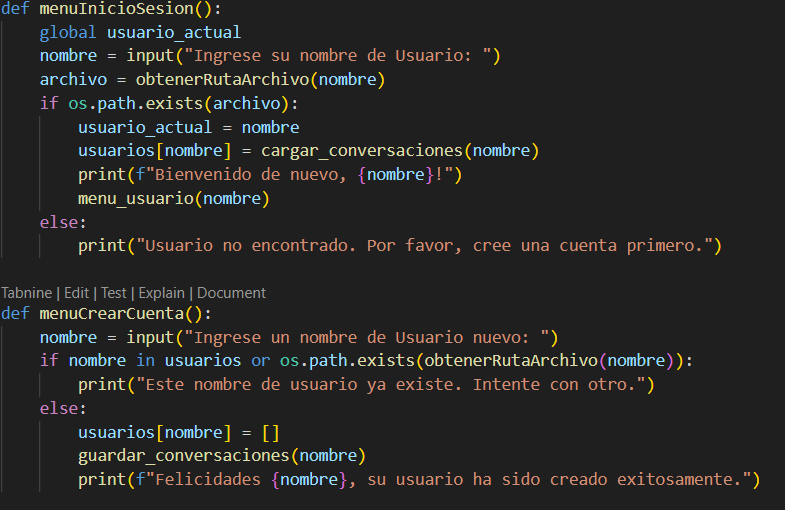


Estas funciones manejan la serialización y deserialización de las estructuras de datos Python a texto plano y viceversa, utilizando la representación literal de Python (str() y ast.literal\_eval()), lo que permite preservar la estructura jerárquica de las conversaciones.

3. Implementación de Funcionalidades Principales:

3.1 Registro y Validación de Usuarios:

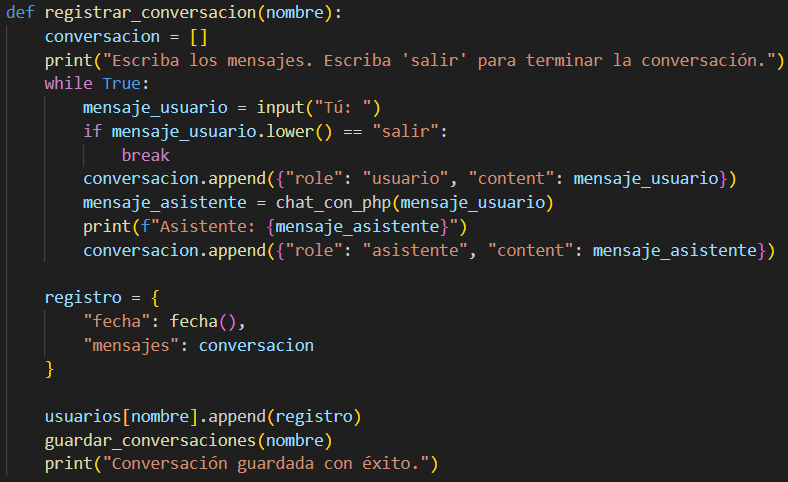
El sistema facilita la creación de nuevos usuarios y la verificación de usuarios ya registrados



Estas funciones se encargan de comprobar si el usuario ya existe revisando los archivos en el disco, asegurando que cada nombre de usuario sea único.

3.2 Registro de Conversaciones:

El sistema permite al usuario mantener conversaciones con el asistente virtual a través de una interfaz de consola



Esta función crea un ciclo interactivo donde el usuario puede enviar mensajes y recibir respuestas del asistente virtual, gracias a la conexión con una API externa (chat\_con\_php). Cada mensaje se guarda en la estructura de datos y se almacena al finalizar la conversación.

3.3 Consulta de Historial:

La función de historial de conversaciones permite al usuario revisar los mensajes que ha intercambiado previamente con el asistente.

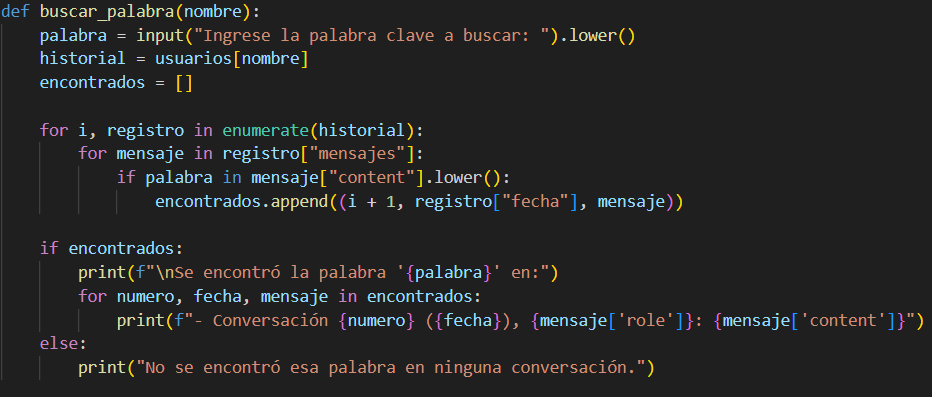
Texto

El contenido generado por IA puede ser incorrecto.

Esta función despliega cada conversación numerada, diferenciando visualmente entre los mensajes del usuario y las respuestas del asistente, además de acomodar los mensajes a la derecha si viene del usuario y a la izquierda si viene del asistente.

3.4 Búsqueda por Palabra Clave:

El sistema permite localizar mensajes específicos mediante la búsqueda de palabras clave:



La función realiza una búsqueda de texto sencilla pero útil, detectando si la palabra clave aparece en cualquier mensaje dentro de las conversaciones, sin importar si está en mayúsculas o minúsculas.

3.5 Generación de Resúmenes:

El sistema puede generar resúmenes concisos de conversaciones seleccionadas:

Texto

El contenido generado por IA puede ser incorrecto.

Esta función toma la conversación que el usuario seleccione y se la manda al Bot para que haga un resumen no más de 50 palabras

4. Sistema de Menús y Flujo de Control:

El sistema utiliza un enfoque basado en menús para la navegación entre funcionalidades:



Este sistema de menús ofrece una interfaz sencilla pero eficaz para el usuario, aprovechando la estructura moderna de match-case introducida en Python 3.10, lo que contribuye a que el código sea más claro y fácil de entender.

Integración con API Externa

El sistema integra una API externa para la funcionalidad del asistente virtual:



Esta integración conecta la aplicación principal con un servicio de inteligencia artificial externo usando el módulo API.py. La función chat\_con\_php envía los mensajes del usuario al asistente virtual y recibe sus respuestas, simplificando el procesamiento del lenguaje natural. Esta arquitectura modular permite aprovechar las capacidades avanzadas del asistente sin tener que implementar los algoritmos localmente. Además, facilita las actualizaciones futuras: si el asistente virtual mejora o cambia, solo se necesita modificar el módulo API sin afectar el resto del sistema.

**Análisis de resultados**

| **Tarea/Requerimiento** | **Estado** | **Observaciones** |
| --- | --- | --- |
| Módulo de Usuarios | Completo | Se implementó correctamente la funcionalidad de registro y validación de usuarios mediante archivos de texto. |
| Módulo de Conversaciones | Completo | El sistema permite registrar nuevas conversaciones con el asistente virtual, almacenando correctamente la estructura de mensajes. |
| Módulo de Consultas | Completo | Se implementaron satisfactoriamente las funcionalidades de visualización del historial y búsqueda por palabras clave en las conversaciones. |
| Módulo de Almacenamiento | Completo | El mecanismo de persistencia mediante archivos de texto funciona correctamente, permitiendo guardar y recuperar conversaciones entre sesiones. |
| Módulo de Interfaz | Completo | Interfaz de consola sencilla que cumple con los requerimientos del proyecto |
| Generación de Resúmenes | Completo | Se implementó un sistema avanzado de resumen utilizando inteligencia artificial para condensar conversaciones en textos concisos y significativos. |

**Conclusiones**

Utilizar estructuras de datos jerárquicas, como diccionarios y listas anidadas, para gestionar las conversaciones es una manera eficaz de organizar y administrar el historial de interacciones entre el usuario y el asistente.Esto permite realizar tareas como búsquedas y generación de resúmenes de manera más sencilla.

El mecanismo de persistencia basado en archivos de texto plano y la serialización/deserialización mediante ast.literal\_eval() proporciona una solución simple pero robusta para el almacenamiento a largo plazo de las conversaciones, aunque podría presentar limitaciones en escenarios con grandes volúmenes de datos.

Al integrar una API externa (chat\_con\_php) para que el asistente virtual funcione, se demuestra que el sistema es flexible y puede ampliarse fácilmente. Esta separación entre la lógica del procesamiento del lenguaje natural y la gestión de conversaciones hace que sea más sencillo realizar mejoras o modificaciones en el backend del asistente en el futuro.

El algoritmo de resumen utiliza inteligencia artificial para crear síntesis concisas de las conversaciones, ofreciendo una solución avanzada que extrae la esencia del contenido de manera más efectiva que métodos simples como la extracción de las primeras palabras.

El diseño modular del código hace que el sistema sea más fácil de entender y mantener, además de permitir que en el futuro se le puedan agregar mejoras como una interfaz gráfica más completa o funciones de análisis avanzado de conversaciones.

**Recomendaciones**

* Considerar la implementación de un sistema de sincronización con la nube para permitir el acceso al historial de conversaciones desde múltiples dispositivos, aumentando así la portabilidad y seguridad de la información almacenada.
* Explorar la posibilidad de incorporar algoritmos de resumen más avanzados basados en técnicas de aprendizaje automático para mejorar la calidad de los resúmenes generados, especialmente para conversaciones técnicas o especializadas.
* Implementar un mecanismo de etiquetado automático de conversaciones basado en el contenido, utilizando técnicas de procesamiento de lenguaje natural para categorizar las interacciones sin intervención manual del usuario.

**Referencias**

Claude AI. (2025). Gráfico generado por inteligencia artificial sobre requerimientos del problema “Mi asistente virtual” [Captura de pantalla]. Creado por la autora a través de Claude AI (<https://claude.ai>).

Clon Geek. (2025). Las últimas tendencias en tecnología de asistentes virtuales. Recuperado de <https://clongeek.com/las-ultimas-tendencias-en-tecnologia-de-asistentes-virtuales/>

Juscilene Vasquez. (2025). Tendencias para asistentes virtuales en 2025: Qué esperar del mercado. Recuperado de <https://www.infonoar.com/tendencias-para-asistentes-virtuales-en-2025-que-esperar-del-mercado/>

SEIDOR. (2024). El Papel de los Asistentes Virtuales en la Vida Cotidiana: Pasado, Presente y Futuro. Recuperado de <https://www.seidor.com/es-es/blog/papel-asistentes-virtuales-vida-cotidiana-pasado-presente-futuro>

Tellaeche, P. (2024). Asistentes digitales con inteligencia artificial. Consultoría TACS. Recuperado de <https://www.consultoriatacs.com/post/asistentes-digitales-con-inteligencia-artificial>