



**Trabajo Final de Grado – Prototipo Tecnológico**

**Carrera:** Licenciatura en Informática

*“Manual de Usuario”*

**Autor:** César Eduardo González

**Legajo:** VINF013326

**Tutor:** Gamboa, Leonardo Gabriel

San Salvador de Jujuy, Provincia de Jujuy – Junio del 2025

## Índice

<i>Objetivo del Sistema</i> .....	3
<i>Tecnologías utilizadas</i> .....	3
<i>Lenguajes de Programación</i> .....	3
<i>Infraestructura sugerida</i> .....	3
<i>Instalación y Prueba del Sistema</i> .....	3
<i>Requisitos Previos</i> .....	3
<i>Ejecución del Prototipo (Símbolo del Sistema)</i> .....	4
<i>Interacción del Usuario</i> .....	4
<i>Recomendaciones para uso efectivo</i> .....	4
<i>Documentación Extra</i> .....	4

## ***Objetivo del Sistema***

Brindar asistencia académica automatizada a estudiantes del nivel secundario mediante un asistente virtual que responde consultas escolares usando lenguaje natural. La herramienta busca fomentar el aprendizaje autónomo y facilitar el acceso a contenidos desde navegador o entorno local.

## ***Tecnologías utilizadas***

### ***Lenguajes de Programación***

- Python 3.11 para el backend
- HTML5, CSS3 y JavaScript (frontend web)
- SQL para la base dedatos
- Tkinter para pruebas locales de escritorio

## ***Infraestructura sugerida***

- Navegador web Google Chrome o Mozilla Firefox actualizado
- Python Instalado / Pycharm (IDE - Entorno de Desarrollo Integrado)
- PostgreSQL/SQLite

## ***Instalación y Prueba del Sistema***

### ***Requisitos Previos***

- Tener instalado Python 3.11 o superior
- Tener acceso a internet
- Instalar los paquetes en el símbolo del sistema línea por línea:
  - ❖ ***pip install flask***: permite instalar Flask, un framework para construir APIs web en Python. Se usa básicamente para el backend del asistente virtual.
  - ❖ ***pip install openai***: librería que permite conectarse a modelos de lenguaje para generar respuestas automáticas.
  - ❖ ***pip install spacy***: biblioteca avanzada de procesamiento del lenguaje natural utilizada para entender preguntas en lenguaje humano.
  - ❖ ***pip install sqlalchemy***: herramienta de mapeo objeto-relacional. Permite gestionar la base de datos de forma sencilla con Python.

- ❖ ***pip install bcrypt***: biblioteca usada para encriptar contraseñas, mejorando la seguridad del sistema de usuarios.
- ❖ ***pip install pytest***: framework para crear y ejecutar pruebas automáticas. Se usa para verificar que el sistema funcione correctamente.
- ❖ ***pip install tk***: la librería gráfica nativa de Python. Permite probar el asistente en una ventana local estilo aplicación de escritorio.

### ***Ejecución del Prototipo (Símbolo del Sistema)***

Desde la carpeta del proyecto, ejecutar el archivo de Python llamado “*asistente\_virtual.py*”. Esto levantará un servidor local en `http://localhost:5000` o abrirá la interfaz con Tkinter.

### ***Interacción del Usuario***

- 1) *Ingreso*: Se accede mediante navegador o ventana de escritorio.
- 2) *Consulta*: El usuario escribe una duda en lenguaje natural.
- 3) *Procesamiento*: El backend interpreta la consulta con técnicas de NLP.
- 4) *Respuesta*: Se muestra una respuesta textual basada en los datos cargados.
- 5) *Registro*: Se guarda la interacción para análisis futuro.

### ***Recomendaciones para uso efectivo***

- Utilizar preguntas claras y específicas
- Para consultas que no son respondidas correctamente, verificar la conexión a internet o si el modelo se encuentra bien cargado.
- En caso de error:
  - Verificar que todos los paquetes se encuentren instalados
  - Comprobar que el archivo “*main.py*” se ejecute correctamente

### ***Documentación Extra***

- Repositorio de GitHub: <https://github.com/CEGJujuy/AV-Demo-V1.git>
- Autor: César Eduardo González / Legajo VINF013326
- Tutor Académico: Leonardo Gamboa