

![[Logo de asesor-legal-salud-colombiano]](<https://github.com/bladealex9848/asesor-legal-salud-colombiano/blob/main/assets/logo.jpg>)

# Asesor Legal en Salud Colombiano ■■■

## Descripción

El Asesor Legal en Salud Colombiano es una herramienta de inteligencia artificial que brinda información y orientación sobre el marco legal y ético del sector salud en Colombia.

## Introducción

Con este asistente, podrás obtener respuestas a tus preguntas sobre responsabilidad médica, normas del sector salud, y cuestiones éticas y bioéticas.

## ¿Cómo funciona?

1. Introduce tu pregunta relacionada con el sector salud en Colombia.
2. La aplicación procesa tu solicitud utilizando su base de conocimiento y la API de OpenAI.
3. Recibe una respuesta clara y fundamentada.

## Funcionalidades

- \* Respuestas basadas en legislación y jurisprudencia colombiana.
- \* Especialización en responsabilidad médica, normas del sector salud, ética y bioética.
- \* Interfaz amigable e intuitiva.
- \* Persistencia de la conversación para un diálogo natural.

## Instalación

1. Asegúrate de tener Python 3.8 o superior instalado en tu máquina.
2. Clona este repositorio: `git clone https://github.com/bladealex9848/asesor-legal-salud-colombiano.git`
3. Navega al directorio del proyecto: `cd asesor-legal-salud-colombiano`
4. Instala las dependencias: `pip install -r requirements.txt`
5. Crea un archivo `.streamlit/secrets.toml` y añade tu ASSISTANT\_ID de OpenAI:

```
...  
ASSISTANT_ID = "tu-assistant-id-aqui"  
OPENAI_API_KEY = "tu-clave-api-aqui"  
...
```

6. Configura tu clave API de OpenAI como variable de entorno o en el archivo `secrets.toml`.

## Uso

1. Ejecuta la aplicación: `streamlit run app.py`
2. Abre tu navegador y ve a `http://localhost:8501`
3. Comienza a interactuar con el Asistente Virtual escribiendo tus preguntas o solicitudes.

## Contribuciones

Las contribuciones son bienvenidas. Por favor, haz un fork del repositorio, crea una nueva rama para tus cambios, y envía un pull request.

## Licencia

Este proyecto está bajo la licencia MIT. Consulta el archivo `LICENSE` para más detalles.

## Autor

Creado por Alexander Oviedo Fadul

[GitHub](https://github.com/bladealex9848) | [Website](https://alexander.oviedo.isabellaea.com/) |

[LinkedIn](https://www.linkedin.com/in/alexander-oviedo-fadul-49434b9a/) |

[Instagram](https://www.instagram.com/alexander.oviedo.fadul) |

[Twitter](https://twitter.com/alexanderofadul) | [Facebook](https://www.facebook.com/alexanderof/) | [W

hatsApp](https://api.whatsapp.com/send?phone=573015930519&text;=Hola%20!Quiero%20conversar%20contigo!%20)

![[Logo del Asistente Virtual]](<https://github.com/bladealex9848/Asistente-Virtual/blob/main/logo.jpg>)

# Asistente Virtual: Tu compañero digital multiexperto

## Descripción

El Asistente Virtual es una herramienta de inteligencia artificial diseñada para proporcionar ayuda en español sobre una amplia variedad de temas. Utilizando la potencia de la API de OpenAI, este asistente ofrece respuestas precisas, claras y personalizadas a tus preguntas y tareas.

## Introducción

Con el Asistente Virtual, podrás interactuar con una IA avanzada capaz de asistirte en diversos temas y tareas. Ya sea que necesites información, análisis, cálculos o sugerencias creativas, este asistente está listo para ayudarte en español. Para desarrollar esta aplicación se utilizaron:

- OpenAI API
- Streamlit

## ¿Cómo funciona?

1. El usuario introduce una pregunta o solicitud.
2. La aplicación crea un hilo de conversación con OpenAI.
3. La pregunta se envía al asistente de OpenAI.
4. El asistente procesa la solicitud y genera una respuesta.
5. La respuesta se muestra al usuario en la interfaz de Streamlit.

## Funcionalidades

- **Respuestas en tiempo real**: Proporciona respuestas instantáneas a tus preguntas.
- **Multiexperto**: Capaz de asistir en una amplia gama de temas y tareas.
- **Interfaz amigable**: Diseñado con Streamlit para una experiencia de usuario intuitiva.
- **Persistencia de conversación**: Mantiene el contexto de la conversación para una interacción más natural.

## Instalación

1. Asegúrate de tener Python 3.8 o superior instalado en tu máquina.
2. Clona este repositorio: `git clone https://github.com/bladealex9848/asistente-virtual.git`
3. Navega al directorio del proyecto: `cd asistente-virtual`
4. Instala las dependencias: `pip install -r requirements.txt`
5. Crea un archivo `.streamlit/secrets.toml` y añade tu ASSISTANT\_ID de OpenAI:

```
'''
```

```
ASSISTANT_ID = "tu-assistant-id-aqui"
```

```
'''
```

6. Configura tu clave API de OpenAI como variable de entorno o en el archivo `secrets.toml`.

## Uso

1. Ejecuta la aplicación: `streamlit run app.py`
2. Abre tu navegador y ve a `http://localhost:8501`
3. Comienza a interactuar con el Asistente Virtual escribiendo tus preguntas o solicitudes.

## Contribuciones

Las contribuciones son bienvenidas. Por favor, haz un fork del repositorio, crea una nueva rama para tus cambios, y envía un pull request.

## Licencia

Este proyecto está bajo la licencia MIT. Consulta el archivo `LICENSE` para más detalles.

## Autor

Creado por Alexander Oviedo Fadul

[GitHub](https://github.com/bladealex9848) | [Website](https://alexander.oviedo.isabellaea.com/) |

[Instagram](https://www.instagram.com/alexander.oviedo.fadul) |

[Twitter](https://twitter.com/alexanderofadul) | [Facebook](https://www.facebook.com/alexanderof/) | [WhatsApp](https://api.whatsapp.com/send?phone=573015930519&text;=Hola%20!Quiero%20conversar%20contigo!%20)

![[Logo de ColDisBot](https://github.com/bladealex9848/ColDisBot/blob/main/logo.jpg)]

# ColDisBot: Tu asistente especializado en Derecho Disciplinario Colombiano

## Descripción

ColDisBot es una herramienta de inteligencia artificial diseñada para proporcionar información precisa y actualizada sobre el régimen disciplinario en Colombia. Utilizando la potencia de la API de OpenAI, este asistente ofrece respuestas fundamentadas en la legislación vigente, incluyendo la Ley 734 de 2002, la Ley 1952 de 2019, la Ley 2094 de 2021 y otras normas relevantes.

## Introducción

Con ColDisBot, podrás interactuar con una IA especializada capaz de asistirte en diversos aspectos del derecho disciplinario colombiano. Ya sea que necesites información sobre procedimientos, faltas, sanciones o garantías procesales, este asistente está listo para ayudarte. Para desarrollar esta aplicación se utilizaron:

- OpenAI API
- Streamlit

## ¿Cómo funciona?

1. El usuario introduce una pregunta o solicitud relacionada con el derecho disciplinario colombiano.
2. La aplicación crea un hilo de conversación con OpenAI.
3. La pregunta se envía al asistente especializado de OpenAI.
4. El asistente procesa la solicitud y genera una respuesta basada en la legislación vigente.
5. La respuesta se muestra al usuario en la interfaz de Streamlit.

## Funcionalidades

- **Respuestas fundamentadas**: Proporciona información basada en la legislación disciplinaria colombiana.
- **Especialización jurídica**: Enfocado exclusivamente en el derecho disciplinario de Colombia.
- **Interfaz amigable**: Diseñado con Streamlit para una experiencia de usuario intuitiva.
- **Persistencia de conversación**: Mantiene el contexto de la consulta para una interacción más natural.

## Instalación

1. Asegúrate de tener Python 3.8 o superior instalado en tu máquina.
2. Clona este repositorio: `git clone https://github.com/bladealex9848/ColDisBot.git`
3. Navega al directorio del proyecto: `cd ColDisBot`
4. Instala las dependencias: `pip install -r requirements.txt`
5. Crea un archivo `.streamlit/secrets.toml` y añade tu `ASSISTANT_ID` de OpenAI:  
...  
`ASSISTANT_ID = "tu-assistant-id-aqui"`  
...
6. Configura tu clave API de OpenAI como variable de entorno o en el archivo `secrets.toml`.

## Uso

1. Ejecuta la aplicación: ``streamlit run app.py``
2. Abre tu navegador y ve a ``http://localhost:8501``
3. Comienza a interactuar con el Asistente Virtual escribiendo tus preguntas o solicitudes.

## Contribuciones

Las contribuciones son bienvenidas. Por favor, haz un fork del repositorio, crea una nueva rama para tus cambios, y envía un pull request.

## Licencia

Este proyecto está bajo la licencia MIT. Consulta el archivo ``LICENSE`` para más detalles.

## Autor

Creado por Alexander Oviedo Fadul

[GitHub](<https://github.com/bladealex9848>) | [Website](<https://alexander.oviedo.isabellaea.com/>) |

[Instagram](<https://www.instagram.com/alexander.oviedo.fadul>) |

[Twitter](<https://twitter.com/alexanderofadul>) | [Facebook](<https://www.facebook.com/alexanderof/>) | [WhatsApp](<https://api.whatsapp.com/send?phone=573015930519&text;=Hola%20!Quiero%20conversar%20contigo!%20>)

# Consejo Seccional de la Judicatura - Vigilancia Judicial Administrativa

Este proyecto tiene como objetivo automatizar y mejorar el proceso de vigilancia judicial administrativa en los Consejos Seccionales de la Judicatura de Colombia. Utilizando técnicas de inteligencia artificial y procesamiento de lenguaje natural, el sistema es capaz de analizar solicitudes de vigilancia, realizar un control de legalidad y generar proyectos de decisión de manera eficiente.

## Características principales

- **Análisis Automatizado**: Evaluación y procesamiento automático de solicitudes de vigilancia judicial administrativa.
- **Control de Legalidad**: Verificación basada en los criterios del ACUERDO No. PSAA11-8716.
- **Generación de Embeddings**: Creación de embeddings a partir de documentos PDF para análisis detallado.
- **Gestión de Roles**: Funcionalidades asignadas a diferentes roles como Escribiente, Profesional Universitario, Auxiliar de Magistrado y Magistrado.
- **Generación de Documentos**: Producción automática de documentos en formato Markdown con resúmenes y proyectos de decisión.

## Requisitos Técnicos

- **Python 3.7 o superior**
  - Bibliotecas necesarias incluidas en `requirements.txt`.
- | Biblioteca            | Descripción   |
|-----------------------|---|
| crewai                | Biblioteca de IA para análisis de texto y generación de embeddings. |
| sentence_transformers | Herramienta para transformaciones avanzadas de texto en vectores.   |
| PyPDF2                | Biblioteca para manipular archivos PDF en Python.                   |
| python-docx           | Creación y modificación de documentos Word usando Python.           |

## Instalación

1. Clona este repositorio:

```
bash
git clone https://github.com/bladealex9848/consejo-seccional-vigilancia.git
```
2. Accede al directorio del proyecto y instala las dependencias:

```
bash
cd consejo-seccional-vigilancia
pip install -r requirements.txt
```

## Uso

1. Coloca los documentos PDF de las solicitudes de vigilancia en la carpeta `documentos`.
2. Ejecuta el script `generador\_embeddings.py` para procesar los documentos:

```
bash
python generador_embeddings/generador_embeddings.py
```
3. Inicia el proceso de vigilancia judicial administrativa:

```
```bash
python consejo_seccional/main.py
```
```

## Contribución

Para contribuir a este proyecto, sigue estos pasos:

1. Haz un fork del repositorio.
2. Crea una rama para tu nueva función o corrección (`git checkout -b mi-nueva-funcion`).
3. Realiza tus cambios y haz commit (`git commit -m "Agregar nueva función"`).
4. Envía tus cambios (`git push origin mi-nueva-funcion`).
5. Abre una Pull Request en GitHub.

## Licencia

Este proyecto está distribuido bajo la Licencia MIT. Consulta el archivo `LICENSE` para más detalles.

## Autor

Este código fue creado por **Alexander Oviedo Fadul** como ejemplo de una Inteligencia Artificial entrenable. Si tiene alguna pregunta o sugerencia, no dude en contactar al autor a través de GitHub o enviando un correo electrónico a [alexander.oviedo.fadul@gmail.com](mailto:alexander.oviedo.fadul@gmail.com).

## Contribuciones

Este código es un ejemplo y está disponible para que cualquiera pueda utilizarlo y modificarlo. Si desea contribuir con mejoras o correcciones, puede enviar una solicitud de pull a través de GitHub. Todas las contribuciones serán revisadas y, si se aprueban, serán incorporadas al repositorio.

## Agradecimientos

**Agradezco a Dios, a mi familia y amigos.** Sin su ayuda, esta aplicación no habría sido posible.

## Donaciones

Somos un equipo dedicado al desarrollo de soluciones de software en el ámbito judicial, con un enfoque especial en la automatización y mejora de los procesos de vigilancia judicial administrativa. Nuestro sistema utiliza técnicas avanzadas de inteligencia artificial para analizar solicitudes de vigilancia, asegurar el cumplimiento de los estándares legales y optimizar la toma de decisiones judiciales.

A lo largo de su implementación, hemos observado la significativa eficiencia y valor que nuestro sistema ha aportado a los operadores judiciales y partes interesadas, permitiéndoles gestionar los procesos con mayor precisión y en menor tiempo. Sin embargo, para continuar con el desarrollo y mejora de este proyecto innovador, necesitamos apoyo económico.

Tu contribución nos permitirá mantener y mejorar nuestro sistema, asegurando que siga siendo una herramienta efectiva y actualizada para la gestión judicial. Cada donación, no importa su tamaño, es crucial para nosotros y nos ayuda a continuar ofreciendo y mejorando este servicio indispensable.

Agradecemos de antemano tu apoyo y esperamos que nuestro sistema de vigilancia judicial administrativa siga siendo una herramienta valiosa en tu labor diaria.

Gracias por tu tiempo y consideración.

### Métodos:

Pypal

**Recuerda que tu imaginación es tu única frontera.**



# Constitutor Colombiano 1991

![[Logo del Constitutor Colombiano

1991]]([https://raw.githubusercontent.com/bladealex9848/Constitutor\\_Colombiano\\_1991/main/logo.png](https://raw.githubusercontent.com/bladealex9848/Constitutor_Colombiano_1991/main/logo.png))

## Descripción

Este proyecto es una aplicación web desarrollada con Streamlit y la API de OpenAI, enfocada en proporcionar información y responder consultas relacionadas con la Constitución Política de Colombia de 1991.

## Configuración

Para configurar y ejecutar este proyecto, sigue estos pasos:

1. Clona el repositorio a tu máquina local.
2. Instala las dependencias utilizando `pip`:

```
'''
```

```
pip install -r requirements.txt
```

```
'''
```

3. Configura una variable de entorno `OPENAI\_API\_KEY` con tu clave API de OpenAI.

4. Ejecuta la aplicación con Streamlit:

```
'''
```

```
streamlit run app.py
```

```
'''
```

## Uso

Una vez en ejecución, la aplicación te permitirá hacer preguntas relacionadas con la Constitución Colombiana, las cuales serán respondidas por el asistente de IA.

## Licencia

[Incluir información sobre la licencia aquí, si aplica]

![[Logo del detector de emociones en tiempo real]](https://raw.githubusercontent.com/bladealex9848/DeteccionDeEmociones/main/logo.png)

# Real-Time Emotion Detector

Este proyecto es un detector de emociones en tiempo real que utiliza la cámara web y un modelo preentrenado para clasificar las emociones de las caras detectadas.

## Configuración

Para ejecutar esta aplicación, necesitarás instalar las dependencias necesarias y configurar tu entorno.

### ### Dependencias

Instala las dependencias utilizando `pip` con el archivo `requirements.txt` proporcionado.

```
```sh
pip install -r requirements.txt
```
```

### ### Ejecución

Para iniciar la aplicación, ejecuta el siguiente comando:

```
```sh
streamlit run app.py
```
```

Donde `app.py` es el nombre de tu script de Streamlit.

## Contribuciones

Las contribuciones son siempre bienvenidas. No dudes en enviar un Pull Request o abrir un Issue si tienes alguna sugerencia o identificas algún error.

## Licencia

Este proyecto está licenciado bajo la [Licencia MIT](LICENSE) - mira el archivo LICENSE para más detalles.

## Créditos

Creado por Alexander Oviedo Fadul. Encuéntrame en: [GitHub](https://github.com/bladealex9848), [Website](https://alexander.oviedo.isabellaea.com/), [Instagram](https://www.instagram.com/alexander.oviedo.fadul), [Twitter](https://twitter.com/alexanderofadul), [Facebook](https://www.facebook.com/alexanderof/), [WhatsApp](https://api.whatsapp.com/send?phone=573015930519&text;=Hola%20!Quiero%20conversar%20contigo!%20).

![[Logo de Generador de Turnos]]([https://raw.githubusercontent.com/bladealex9848/Generador\\_Fechas/main/logo.png](https://raw.githubusercontent.com/bladealex9848/Generador_Fechas/main/logo.png))

# Generador de Turnos Streamlit App

## Descripción

Esta aplicación de Streamlit genera un programa de turnos basado en un rango de fechas seleccionado, permitiendo excluir o incluir fines de semana y festivos, según se desee.

## Características

- Carga de festivos desde un archivo CSV.
- Selección de rango de fechas para generar turnos.
- Opción para incluir o excluir fines de semana.
- Opción para incluir o excluir festivos.
- Generación de turnos en orientación horizontal o vertical.
- Exportación de turnos a un archivo Excel.

## Uso

1. Carga un archivo de festivos (si se requiere).
2. Selecciona el rango de fechas de interés.
3. Decide si incluir o no fines de semana y festivos.
4. Elige la orientación de la salida (horizontal o vertical).
5. Genera y descarga el archivo Excel con los turnos programados.

## Contribuciones

Las contribuciones son siempre bienvenidas. No dudes en enviar un Pull Request o abrir un Issue si tienes alguna sugerencia o identificas algún error.

## Licencia

Este proyecto está licenciado bajo la [Licencia MIT](LICENSE) - mira el archivo LICENSE para más detalles.

## Créditos

Creado por Alexander Oviedo Fadul. Encuéntrame en: [GitHub](<https://github.com/bladealex9848>), [Website](<https://alexander.oviedo.isabellaea.com/>), [Instagram](<https://www.instagram.com/alexander.oviedo.fadul>), [Twitter](<https://twitter.com/alexanderofadul>), [Facebook](<https://www.facebook.com/alexanderof/>), [WhatsApp](<https://api.whatsapp.com/send?phone=573015930519&text;=Hola%20!Quiero%20conversar%20contigo!%20>).

|   |   |   |                                     |  |   |
|---|---|---|-------------------------------------|--|---|
| <b>PROCESO:</b>   | <b>Comunicación Institucional</b>   |   | <b>Fecha:</b>                       | <b>27/08/2024</b>                                  |   |
| Nivel Central   | <input type="checkbox"/>  | Nivel Seccional   | <input checked="" type="checkbox"/> | Nombre de la dependencia                           | Consejo Seccional de la Judicatura de Sucre |
| <b>HALLAZGO</b>   | <input type="checkbox"/>  | <b>OPORTUNIDAD</b>                                      | <input type="checkbox"/>            | <b>ACCIÓN DE GESTIÓN</b>                           |   |
| <b>FUENTE DE INFORMACION:</b>   | <input type="checkbox"/>  | Análisis de indicadores                                 | <input checked="" type="checkbox"/> | Auditoría externa                                  | <input type="checkbox"/>                    |
| Auditoría interna   | <input type="checkbox"/>  | Evaluación de la satisfacción de las partes interesadas | <input checked="" type="checkbox"/> | Revisión del SIGCMA por parte de la Alta Dirección | <input checked="" type="checkbox"/>         |
| Salida no conforme  | <input type="checkbox"/>  | Quejas, reclamos y sugerencias                          | <input checked="" type="checkbox"/> | Valoración de los riesgos y/o peligros             | <input type="checkbox"/>                    |
| Inspecciones seguridad  | <input type="checkbox"/>  | Análisis vulnerabilidad de emergencias                  | <input type="checkbox"/>            | Otro ¿Cual?: Control Interno                       | <input type="checkbox"/>                    |
| <b>DESCRIPCIÓN DEL HALLAZGO, OPORTUNIDAD O ACCIÓN</b>   |   |   |                                     |  |   |
| Se ha identificado la oportunidad de implementar un sistema integral para la gestión de las solicitudes de vigilancia judicial administrativa en el Consejo Seccional de la Judicatura de Sucre. Actualmente, el proceso se realiza de manera manual utilizando correo electrónico, archivos de Excel como base de datos y atención mediante correo electrónico y llamadas telefónicas. Aunque este método ha permitido mantener tiempos de atención aceptables, se reconoce el potencial para mejorar la eficiencia, trazabilidad y seguimiento de las solicitudes mediante la implementación de un sistema digital integrado. |   |   |                                     |  |   |
| <b>CORRECCION (Solo para hallazgos)</b>   |   |   | Aplica                              |  | <input type="checkbox"/>                    |
|   |   |   | No Aplica                           |  | <input checked="" type="checkbox"/>         |
| <b>Actividades</b>  | <b>Responsable de la actividad</b>  | <b>Fecha ejecución</b>                                  | <b>Seguimiento</b>                  |  |   |
| 1.  |   |   |                                     |  |   |
| 2.  |   |   |                                     |  |   |
| 3.  |   |   |                                     |  |   |
| <b>ACCION PREVENTIVA</b>  | <input type="checkbox"/>  | <b>ACCION CORRECTIVA</b>                                | <input type="checkbox"/>            | <b>ACCION DE MEJORA</b>                            |   |
| <b>CAUSAS DEL HALLAZGO, OPORTUNIDAD O ACCIÓN</b>  |   |   |                                     |  |   |
| 1. Proceso manual de registro y seguimiento de solicitudes de vigilancia judicial administrativa.<br>2. Uso de herramientas no especializadas (correo electrónico y Excel) para la gestión de información.<br>3. Potencial para mejorar la trazabilidad y el seguimiento en tiempo real de las solicitudes.<br>4. Oportunidad de centralizar la comunicación entre las partes involucradas en el proceso.<br>5. Posibilidad de implementar herramientas tecnológicas para la generación automatizada de informes y análisis de datos.   |   |   |                                     |  |   |
| <b>DESCRIPCIÓN DE ACTIVIDADES PARA GESTIONAR LAS CAUSAS</b>   |   |   |                                     |  |   |
| <b>Actividades a realizar</b>   | <b>Responsable de la actividad</b>  | <b>Fecha inicio</b>                                     | <b>Fecha fin</b>                    |  |   |
| 1. Desarrollar e implementar el aplicativo web "Gestión Integral de Vigilancias".   | Alexander Oviedo Fadul, Profesional Universitario Grado 11 del Consejo Seccional de la Judicatura de Sucre      | 15/01/2024  | 30/06/2024                          |  |   |
| 2. Realizar pruebas piloto del sistema con el equipo de vigilancias judiciales.   | Luis Felipe Gandara Taboada, Profesional Universitario Grado 11 del Consejo Seccional de la Judicatura de Sucre | 1/07/2024   | 5/07/2024                           |  |   |
| 3. Capacitar al personal del Consejo Seccional en el uso del nuevo sistema.   | Francisco Montes , Escribiente del Consejo Seccional de la Judicatura de Sucre                                  | 1/07/2024   | 5/07/2024                           |  |   |
| 4. Implementar el sistema en todos los procesos de vigilancia judicial administrativa.  | Ivon Alejandra Vasquez Palencia, Auxiliar Despacho 02 del Consejo Seccional de la Judicatura de Sucre           | 5/07/2024   | En curso                            |  |   |
| 5. Desarrollar e implementar la interfaz para usuarios de justicia.   | Alexander Oviedo Fadul, Profesional Universitario Grado 11 del Consejo Seccional de la Judicatura de Sucre      | 6/08/2024   | En curso                            |  |   |
| 6. Desarrollar e implementar la interfaz para jueces objeto de vigilancia.  | Alexander Oviedo Fadul, Profesional Universitario Grado 11 del Consejo Seccional de la Judicatura de Sucre      | 6/08/2024   | En curso                            |  |   |

| PROCESO:  | Comunicación Institucional  | Fecha:   | 27/08/2024        |
|---|---|--|-------------------|
| 7. Realizar seguimiento y evaluación continua del sistema para identificar mejoras.           | Fabián Elías Paternina Martínez,<br>Magistrado Despacho 01 del Consejo<br>Seccional de la Judicatura de Sucre | 5/07/2024  | Continuo          |
| SEGUIMIENTO   |   |  |                   |
| Seguimiento de las actividades realizadas   |   | Responsable del Seguimiento  | Fecha Seguimiento |
| 1. Desarrollo del aplicativo: 100% completado al 30/06/2024.                                  |   | Rozana Beatriz Abello Albino,<br>Magistrada Presidenta del Consejo<br>Seccional de la Judicatura de Sucre                | 30/06/2024        |
| 2. Pruebas piloto: 100% completado al 05/07/2024.   |   | Maria Claudia Lozano Acosta,<br>Profesional Universitario Grado 11<br>del Consejo Seccional de la<br>Judicatura de Sucre | 5/07/2024         |
| 3. Capacitación del personal: 100% completado al 05/07/2024.                                  |   | Maria Claudia Lozano Acosta,<br>Profesional Universitario Grado 11<br>del Consejo Seccional de la<br>Judicatura de Sucre | 5/07/2024         |
| 4. Implementación del sistema: 80% completado al 27/08/2024. En uso activo.                   |   | Maria Claudia Lozano Acosta,<br>Profesional Universitario Grado 11<br>del Consejo Seccional de la<br>Judicatura de Sucre | 27/08/2024        |
| 5. Interfaz para usuarios de justicia: 50% completado al 27/08/2024. En desarrollo.           |   | Maria Claudia Lozano Acosta,<br>Profesional Universitario Grado 11<br>del Consejo Seccional de la<br>Judicatura de Sucre | 27/08/2024        |
| 6. Interfaz para jueces: 40% completado al 27/08/2024. En desarrollo.                         |   | Maria Claudia Lozano Acosta,<br>Profesional Universitario Grado 11<br>del Consejo Seccional de la<br>Judicatura de Sucre | 27/08/2024        |
| 7. Seguimiento y evaluación: En curso, reuniones trimestrales realizadas desde el 05/07/2024. |   | Fabián Elías Paternina Martínez,<br>Magistrado Despacho 01 del<br>Consejo Seccional de la Judicatura<br>de Sucre         | 5/07/2024         |

| REVISIÓN  |                                     |                          |  |
|---|-------------------------------------|--------------------------|--|
| Las actividades tomadas:  | SI                                  | NO                       | Justificación  |
| Fueron adecuadas?   | <input checked="" type="checkbox"/> | <input type="checkbox"/> | El sistema implementado mejora significativamente el proceso de gestión de las solicitudes de vigilancia judicial administrativa, manteniendo la eficiencia del sistema manual anterior y añadiendo beneficios adicionales.  |
| Fueron Convenientes?  | <input checked="" type="checkbox"/> | <input type="checkbox"/> | La inversión en el sistema se justifica por la mejora en la trazabilidad y el seguimiento de las solicitudes, así como por la centralización de la información, lo que facilita la generación de informes y el análisis de datos.                                  |
| Fueron eficaces?  | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Se ha logrado el objetivo de mejorar la gestión de las solicitudes de vigilancia judicial administrativa. La trazabilidad de las solicitudes es ahora del 100%, y se ha mantenido la eficiencia en los tiempos de atención, con potencial para mejorarlos aún más. |
| REVISÓ: Responsable del proceso o enlace seccional: Nombre, Cargo y Firma                         |                                     |                          | Rozana Beatriz Abello Albino Magistrada del Consejo Seccional de la Judicatura de Sucre Despacho 02  |
| Estado de la Acción: Abierta <input checked="" type="checkbox"/> Cerrada <input type="checkbox"/> |                                     |                          |  |
| PARA USO EXCLUSIVO DEL COORDINADOR DE MEJORAMIENTO  |                                     |                          |  |
| ACCION PREVENTIVA No.   | ACCION CORRECTIVA No.               | ACCION DE MEJORA No.     |  |

Se llevarán a cabo reuniones trimestrales de seguimiento en las que cada responsable presentará el avance de sus actividades asignadas. Para visualizar el progreso de manera eficiente, se implementará un tablero Kanban. Como parte del proceso de documentación, se recopilarán evidencias tales como capturas de pantalla del sistema en desarrollo, registros detallados de las sesiones de capacitación realizadas y resultados comprehensivos de las pruebas piloto.

La efectividad de la implementación se evaluará mediante tres indicadores clave: la mejora en la trazabilidad de las solicitudes, la optimización en los tiempos de procesamiento y el incremento en los niveles de satisfacción de los usuarios.

Es importante señalar que la acción de mejora se mantiene en estado abierto debido a que las interfaces para usuarios de justicia y jueces objeto de vigilancia, componentes críticos para la implementación integral del sistema, aún se encuentran en fase de desarrollo. El cierre de esta acción de mejora se considerará una vez que se hayan completado estas interfaces, se haya evaluado su impacto de manera exhaustiva y se haya obtenido la aprobación formal del Consejo Superior de la Judicatura.



Rama Judicial  
Consejo Superior de la Judicatura  
República de Colombia

Consejo Superior de la Judicatura  
Unidad de Desarrollo y Análisis Estadístico  
REPORTE DE ACCIONES PREVENTIVAS CORRECTIVAS Y DE MEJORA

**SIGCMA**  
- Coordinación Nacional -

|          |                              |                            |           |        |                                  |
|----------|------------------------------|----------------------------|-----------|--------|----------------------------------|
| PROCESO: |                              | Comunicación Institucional |           | Fecha: | 27/08/2024                       |
| F-ESG-05 | COORDINACIÓN NACIONAL SIGCMA |                            | SIGCMA    |        | COORDINACION NACIONAL DEL SIGCMA |
| VERSIÓN  | FECHA                        |                            | FECHA     |        | FECHA                            |
| 5        | 11/06/2022                   |                            | 1/06/2023 |        | 12/07/2023                       |

# Sistema de Gestión de Expedientes Electrónicos Judiciales

## Tabla de Contenidos

1. [Descripción](#descripción)
2. [Características](#características)
3. [Estructura del Proyecto](#estructura-del-proyecto)
4. [Instalación](#instalación)
5. [Uso](#uso)
6. [Manual Técnico](#manual-técnico)
7. [Manual de Usuario](#manual-de-usuario)
8. [Contribución](#contribución)
9. [Registro de Cambios](#registro-de-cambios)
10. [Créditos](#créditos)
11. [Licencia](#licencia)

## Descripción

El Sistema de Gestión de Expedientes Electrónicos Judiciales es una solución avanzada que combina automatización robótica de escritorio (RDA) con tecnologías modernas para optimizar la gestión de expedientes electrónicos en el ámbito judicial. Este sistema está diseñado para cumplir con los estrictos estándares del Plan Estratégico de Transformación Digital de la Rama Judicial, así como con los requisitos técnicos y funcionales del acuerdo PCSJA20-11567 de 2020, que establece el protocolo para la gestión de documentos electrónicos, su digitalización y la conformación del expediente electrónico, en su versión 2.

## Características Principales

- **Automatización de la Creación del Índice Electrónico**: El sistema automatiza el proceso de creación del índice electrónico, reduciendo significativamente el tiempo y los recursos necesarios para esta tarea crítica.
- **Extracción de Metadatos de Archivos**: Utiliza técnicas avanzadas para extraer metadatos de diversos tipos de archivos, asegurando una completa y precisa documentación de cada expediente.
- **Generación de Índices en Formatos Excel**: Facilita la generación de índices en formato Excel, compatible con las necesidades específicas de la gestión documental judicial.
- **Interfaces de Usuario Intuitivas**.
- **Versión de Escritorio con PyQt5**: Ofrece una interfaz de usuario intuitiva y amigable, desarrollada con PyQt5, que permite una experiencia de usuario fluida y eficiente.
- **Versión Web con Streamlit**: Además, cuenta con una interfaz web desarrollada utilizando Streamlit, que brinda flexibilidad y accesibilidad para usuarios que prefieren o necesitan interactuar a través de la web.
- **Cumplimiento con los Estándares Judiciales Colombianos**: El sistema se ajusta rigurosamente a los estándares y protocolos judiciales colombianos, garantizando su aplicabilidad y aceptación en el contexto legal y judicial del país.

Este proyecto representa un avance significativo en la gestión de expedientes electrónicos, ofreciendo una solución integral que no solo mejora la eficiencia y la precisión de los procesos judiciales, sino que también fomenta la transparencia y la accesibilidad de la información en el sistema judicial.

## Estructura del Proyecto

```

...
GestionExpedienteElectronico/
■
■■■■ app.py
■■■■ main.py
■■■■ index_generator.py
■■■■ file_utils.py
■■■■ metadata_extractor.py
■■■■ excel_handler.py
■
■■■■ assets/
■ ■■■■ 00IndiceElectronicoC0.xlsm
| ■■■■ guia_uso.pdf
| ■■■■ logo.png
| ■■■■ logo_CSJ_Sucre.png
| ■■■■ logo_CSJ_Sucre.jpg
|
■■■■ pages/
■ ■■■■ 1_■_Hoja_de_Ruta.py
| ■■■■ 2_■_Experto_en_Expediente_Electronico.py
|
■■■■ tests/
■ ■■■■ test_expediente_processor.py
■
■■■■ requirements.txt
■■■■ .gitignore
■■■■ README.md
...

```

## Instalación

```

1. Clonar el repositorio:
git clone https://github.com/bladealex9848/GestionExpedienteElectronico.git
2. Navegar al directorio del proyecto:
cd GestionExpedienteElectronico
3. Crear un entorno virtual:
- En Windows:
...

python -m venv venv
.\venv\Scripts\activate
...

- En macOS y Linux:
...

python3 -m venv venv
source venv/bin/activate
...

4. Instalar las dependencias:
...

pip install --upgrade pip
pip install watchdog
pip install -r requirements.txt
...

```



## Uso

Para ejecutar la versión de escritorio:

...

```
python app.py
```

...

Para ejecutar la versión web:

...

```
streamlit run ■_Inicio.py.py
```

...

Siga las instrucciones en la interfaz de usuario para cargar y procesar los expedientes.

## Manual Técnico

### ### 1. Arquitectura del Sistema

El Sistema de Gestión de Expedientes Electrónicos Judiciales está construido utilizando las siguientes tecnologías:

- Lenguaje de programación: Python 3.x
- Frameworks:
  - Interfaz de escritorio: PyQt5
  - Interfaz web: Streamlit
- Principales librerías:
  - pandas: Para manipulación y análisis de datos
  - openpyxl: Para manejo de archivos Excel
  - PyPDF2: Para extracción de metadatos de archivos PDF
  - xlwings: Para interacción avanzada con Excel (macros)

### ### 2. Estructura del Proyecto

...

GestionExpedienteElectronico/

■

- app.py # Aplicación de escritorio (PyQt5)
- ■\_Inicio.py # Aplicación web (Streamlit)
- index\_generator.py # Lógica de generación del índice
- file\_utils.py # Utilidades para manejo de archivos
- metadata\_extractor.py # Extracción de metadatos
- excel\_handler.py # Manejo de archivos Excel

■

- assets/ # Recursos estáticos
- pages/ # Páginas de la aplicación web
- tests/ # Pruebas unitarias

■

- requirements.txt # Dependencias del proyecto
- .gitignore
- README.md

...

### ### 3. Componentes Principales

#### #### 3.1 app.py (Aplicación de Escritorio)

- Clase `MainWindow`: Interfaz gráfica principal
- Clase `IndexGeneratorThread`: Hilo para procesamiento asíncrono

#### #### 3.2 ■\_Inicio.py (Aplicación Web)

- Función `main()`: Punto de entrada de la aplicación Streamlit
- Funciones auxiliares para manejo de archivos y UI

#### #### 3.3 index\_generator.py

- ``generate_index_from_scratch()``: Genera el índice sin plantilla
- ``generate_index_from_template()``: Genera el índice usando plantilla Excel

#### 3.4 file\_utils.py

- ``rename_files()``: Renombra archivos según el protocolo
- ``get_file_metadata()``: Obtiene metadatos básicos de archivos

#### 3.5 metadata\_extractor.py

- Funciones específicas para extraer metadatos de diferentes tipos de archivos (PDF, Word, Excel, imágenes)

#### 3.6 excel\_handler.py

- ``save_excel_file()``: Guarda el índice en formato Excel
- ``create_new_excel()``: Crea un nuevo archivo Excel con formato
- ``fill_template_xlwings()``: Llena la plantilla Excel usando xlwings

### 4. Flujo de Datos

1. El usuario selecciona la carpeta del expediente o carga archivos (versión web).
2. Se renombran los archivos según el protocolo.
3. Se extraen los metadatos de cada archivo.
4. Se genera el índice electrónico (DataFrame).
5. Se crea o actualiza el archivo Excel del índice.
6. Se presenta el resultado al usuario.

### 5. Seguridad y Manejo de Errores

- Validación de entradas de usuario en la interfaz.
- Manejo de excepciones en procesos críticos como la lectura de archivos y generación del índice.
- Uso de directorios temporales para el manejo seguro de archivos en la versión web.

### 6. Requisitos del Sistema

- Python 3.7+
- Dependencias listadas en ``requirements.txt``
- Para la versión de escritorio: Sistema operativo compatible con PyQt5
- Para la versión web: Navegador web moderno

### 7. Instalación y Configuración

1. Clonar el repositorio:

```
git clone https://github.com/bladealex9848/GestionExpedienteElectronico.git
```

2. Crear y activar un entorno virtual.
3. Instalar dependencias:

```
pip install -r requirements.txt
```

### 8. Despliegue

- Versión de Escritorio: Generar ejecutable usando PyInstaller
- Versión Web: Desplegar en plataforma compatible con Streamlit (ej. Streamlit Share, Heroku)

### 9. Mantenimiento y Actualización

- Actualizar dependencias periódicamente.
- Realizar pruebas unitarias antes de cada actualización mayor.
- Mantener compatibilidad con versiones anteriores de los formatos de archivo.

### 10. Consideraciones de Rendimiento

- La aplicación está diseñada para manejar expedientes de tamaño moderado.
- Para expedientes muy grandes, considerar procesamiento por lotes.

### 11. Integración y APIs

- Actualmente, el sistema funciona de manera independiente.
- Potencial para futura integración con sistemas de gestión judicial mediante APIs.

### 12. Logging y Monitoreo

- Implementar un sistema de logging para rastrear errores y uso.

- Considerar la integración de herramientas de monitoreo para la versión web.  
Este manual técnico proporciona una visión general completa de la arquitectura y funcionamiento del Sistema de Gestión de Expedientes Electrónicos Judiciales. Para más detalles sobre componentes específicos, consulte los comentarios en el código fuente de cada archivo.

## Manual de Usuario

### ### 1. Introducción

El Sistema de Gestión de Expedientes Electrónicos Judiciales es una herramienta diseñada para facilitar la creación y gestión de índices electrónicos para expedientes judiciales. Este manual le guiará a través de las funcionalidades principales del sistema, tanto en su versión de escritorio como en su versión web.

### ### 2. Requisitos del Sistema

- Para la versión de escritorio:
- Sistema operativo: Windows 10/11, macOS 10.14+, o Linux (distribución reciente)
- 4 GB de RAM mínimo
- 200 MB de espacio en disco duro
- Para la versión web:
- Navegador web actualizado (Chrome, Firefox, Safari, Edge)
- Conexión a internet estable

### ### 3. Instalación (Versión de Escritorio)

1. Descargue el ejecutable portable desde [enlace de descarga](https://gestionexpedienteelectronico.streamlit.app).
2. Ejecute el archivo descargado y siga las instrucciones en pantalla.
3. No se requiere instalación adicional.

### ### 4. Acceso (Versión Web)

1. Abra su navegador web.
2. Visite la URL: [URL de la aplicación web](https://gestionexpedienteelectronico.streamlit.app).
3. No se requiere instalación ni registro.

### ### 5. Interfaz de Usuario

#### #### 5.1 Versión de Escritorio

- **Ventana Principal**: Contiene todos los controles necesarios para la gestión de expedientes.
- **Botón "Seleccionar Carpeta"**: Permite elegir la carpeta que contiene los archivos del expediente.
- **Botón "Generar Índice"**: Inicia el proceso de creación del índice electrónico.
- **Opciones de Generación**: Permite elegir entre generar desde cero o usar una plantilla.
- **Barra de Progreso**: Muestra el avance del proceso de generación.
- **Área de Información**: Muestra mensajes y resultados del proceso.

#### #### 5.2 Versión Web

- **Panel Principal**: Área central donde se cargan los archivos y se inicia el proceso.
- **Barra Lateral**: Contiene enlaces a recursos adicionales, marco normativo, hoja de ruta, y chat bot experto en expedientes electrónicos.
- **Área de Carga de Archivos**: Permite subir los documentos del expediente.
- **Botón "Generar Índice Electrónico"**: Inicia el proceso de creación del índice.

### ### 6. Uso Básico

#### #### 6.1 Generar un Índice Electrónico (Versión de Escritorio)

1. Abra la aplicación.
2. Haga clic en "Seleccionar Carpeta" y elija la carpeta que contiene los archivos del expediente.
3. Seleccione la opción de generación (desde cero o usar plantilla).
4. Haga clic en "Generar Índice".
5. Espere a que el proceso termine. El índice se guardará en la misma carpeta del expediente.

#### #### 6.2 Generar un Índice Electrónico (Versión Web)

1. Acceda a la aplicación web.
2. Arrastre y suelte los archivos del expediente en el área de carga o use el botón para seleccionarlos.

3. Una vez cargados todos los archivos, haga clic en "Generar Índice Electrónico".
4. Espere a que el proceso termine. Se le proporcionará un enlace para descargar el índice generado.

#### ### 7. Funciones Avanzadas

- **\*\*Uso de Plantillas\*\***: Para usar una plantilla personalizada, seleccione la opción correspondiente antes de generar el índice (solo versión de escritorio).
- **\*\*Descarga de Recursos\*\***: Utilice los enlaces en la barra lateral (versión web) o los botones correspondientes (versión de escritorio) para descargar la plantilla Excel y la guía de uso.

#### ### 8. Solución de Problemas

- **\*\*El índice no se genera\*\***: Asegúrese de que todos los archivos en la carpeta sean válidos y accesibles.
- **\*\*Errores en la carga de archivos (versión web)\*\***: Verifique que el tamaño total de los archivos no exceda el límite permitido.
- **\*\*La aplicación se cierra inesperadamente\*\***: Asegúrese de tener la última versión instalada y que su sistema cumpla con los requisitos mínimos.

#### ### 9. Mejores Prácticas

- Organice los archivos del expediente en una sola carpeta antes de generar el índice.
- Utilice nombres de archivo descriptivos y cortos.
- Actualice regularmente el índice si añade o modifica documentos en el expediente.

#### ### 10. Soporte y Contacto

Para obtener ayuda adicional o reportar problemas:

- Visite nuestra [página de GitHub](<https://github.com/bladealex9848/GestionExpedienteElectronico/issues>) para reportar problemas.
- Contacte al soporte técnico en [correo electrónico de soporte](mailto:aoviedof@cendoj.ramajudicial.gov.co) para preguntas más específicas o problemas técnicos.

#### ### 11. Glosario

- **\*\*Índice Electrónico\*\***: Documento que lista y describe todos los archivos que componen un expediente judicial electrónico.
- **\*\*Metadatos\*\***: Información sobre los archivos, como fecha de creación, tamaño, tipo, etc.
- **\*\*Plantilla\*\***: Archivo Excel preformateado para la creación de índices electrónicos.

#### ### 12. Actualizaciones

- Visite regularmente [URL del repositorio o sitio web](<https://gestionexpedienteelectronico.streamlit.app/>) para obtener las últimas actualizaciones y mejoras del sistema.

Este manual de usuario proporciona una guía completa para utilizar el Sistema de Gestión de Expedientes Electrónicos Judiciales. Para información más detallada sobre aspectos técnicos o legales, consulte el Manual Técnico o la documentación legal proporcionada.

## Contribución

Planeamos añadir más funcionalidades al Sistema de Gestión de Expedientes Electrónicos Judiciales con el tiempo. Las contribuciones son bienvenidas.

## Registro de Cambios

### 2024-08-18: Actualización mayor y revisión externa (v.1.3.2)

#### #### Documentación

- Actualización del ABC "Protocolo para la Gestión de Documentos Electrónicos en la Rama Judicial" a la versión 1.2.0.
- Revisión y actualización de la guía de usuario con mejoras en claridad y consistencia.
- Modificación de la recomendación sobre el prefijo numérico en la nomenclatura de archivos, permitiendo mayor flexibilidad.

- Inclusión de referencias a documentación adicional sobre el proceso de transformación digital de la Rama Judicial:

- \* Plan de Digitalización de Expedientes 2020-2022

- \* CIRCULAR PCSJC20-32

- \* Plan Estratégico de Transformación Digital

- \* Programa de Expediente Electrónico

- \* Sistema Integrado Único de Gestión Judicial - Core

- \* Proyectos de Transición

- Actualización de las referencias bibliográficas al formato APA séptima edición.

- Mejora en la estructura del documento con la adición de saltos de página para mejor legibilidad.

#### Software

- Corrección de errores en la generación del índice en formato Excel.

- Resolución del problema de ventanas sin mensajes durante la ejecución del programa.

- Implementación de mejoras en el manejo de archivos y la generación del índice electrónico, contribuidas por [HammerDev99](https://github.com/HammerDev99):

- \* Optimización del algoritmo de renombrado de archivos.

- \* Función más robusta para la extracción de metadatos de diferentes tipos de archivos.

- \* Mejora en la lógica de generación del índice electrónico.

- Inicio del proceso para resolver el problema de seguridad que marca el ejecutable como sospechoso.

#### Nuevas Características

- Mantenimiento de las nuevas páginas implementadas en la versión anterior:

- \* Hoja de Ruta (1\_■\_Hoja\_de\_Ruta.py): Visualización interactiva del progreso del proyecto.

- \* Experto en Expediente Electrónico (2\_■\_Experto\_en\_Expediente\_Electronico.py): Chatbot especializado en gestión documental judicial.

#### Revisión Externa

- Revisión completa realizada por Daniel Arbeláez Álvarez, Técnico CSP de Bello, Antioquia.

- Incorporación de sugerencias y correcciones basadas en la revisión externa.

#### Próximos Pasos

- Continuar la optimización del software basándose en el feedback recibido.

- Implementar soluciones para los problemas de seguridad identificados.

- Actualizar la documentación para reflejar las últimas normativas y planes de transformación digital de la Rama Judicial.

### 2024-08-15: Actualización menor y corrección de errores (v.1.3.1)

- Corrección de errores menores en la interfaz de usuario y el proceso de generación de índices.

- Supresión de las carpetas marco\_normativo y temp\_expediente en la versión web y escritorio para optimizar el espacio de almacenamiento y carga de archivos.

- Mejoras en la gestión de errores y mensajes de usuario para una experiencia más fluida y menos propensa a errores.

- En la versión de escritorio se eliminó el guardar el índice electrónico (xlsx) en la carpeta del expediente y se dejó solamente el guardado del índice electrónico (xlsm) en la carpeta del expediente para optimizar el espacio de almacenamiento y carga de archivos.

### 2024-08-10: Actualización mayor y mejoras en la versión web (v.1.3.0)

- Rediseño completo de la interfaz web, con un panel lateral mejorado.

- Implementación de descarga de recursos adicionales y marco normativo desde el panel lateral.

- Adición de badges de GitHub y contador de visitantes en la página principal.

- Mejora en la configuración de Streamlit para una mejor experiencia de usuario.

- Optimización del manejo de archivos temporales para mejorar la seguridad y el rendimiento.

- Implementación de manejo de errores robusto, especialmente para la carga de imágenes y recursos.

- Preparación del proyecto para generación de ejecutable para Windows.

- Actualización de la documentación y guías de usuario.

### 2024-08-10: Actualización de la versión web (v.1.2.0)

- Rediseño de la interfaz web, moviendo recursos adicionales y créditos al panel lateral.

- Simplificación del proceso de generación de índices, eliminando la opción de usar plantilla.

- Implementación de manejo de archivos temporales para mejorar la seguridad y el rendimiento en entornos multi-usuario.
  - Adición de funcionalidad para comprimir y descargar el expediente completo con el índice generado.
  - Mejora en la gestión de recursos del servidor al procesar múltiples solicitudes simultáneas.
- ### 2024-08-09: Actualización y correcciones (v.1.1.1)
- Mejora en el manejo de la plantilla Excel con macros (.xlsm).
  - Corrección del problema de generación de archivos .xlsx adicionales.
  - Optimización del proceso de renombrado de archivos para respetar la numeración existente.
  - Ajuste en la lógica de generación del índice para excluir correctamente archivos no deseados.
  - Mejora en la compatibilidad con diferentes sistemas operativos.
  - Actualización de la documentación y guías de usuario.
- ### 2024-08-09: Actualización mayor (v.1.1.0)
- Implementación de la versión de escritorio utilizando PyQt5.
  - Adición de la funcionalidad para generar índices tanto desde cero como utilizando una plantilla.
  - Mejora en la extracción de metadatos para soportar múltiples tipos de archivos.
  - Implementación de un manejador de Excel para crear y modificar archivos de índice.
  - Actualización de la estructura del proyecto para soportar tanto la versión web como la de escritorio.
  - Adición de pruebas unitarias para las nuevas funcionalidades.
  - Actualización de la documentación para reflejar los nuevos cambios y características.
- ### 2024-08-08: Primera versión (v.1.0.0)
- Lanzamiento inicial del Sistema de Gestión de Expedientes Electrónicos Judiciales.
  - Implementación de la versión web utilizando Streamlit.

## Créditos

Este proyecto es una evolución del trabajo inicial realizado por [HammerDev99 Daniel]([https://github.com/HammerDev99/GestionExpedienteElectronico\\_Version1](https://github.com/HammerDev99/GestionExpedienteElectronico_Version1)), a quien se le reconoce y agradece por sentar las bases de esta herramienta.

Desarrollado y mantenido por Alexander Oviedo Fadul, Profesional Universitario Grado 11 en el Consejo Seccional de la Judicatura de Sucre.

[GitHub](<https://github.com/bladealex9848>) | [Website](<https://alexander.oviedo.isabellaea.com/>) | [Instagram](<https://www.instagram.com/alexander.oviedo.fadul>) | [Twitter](<https://twitter.com/alexanderofadul>) | [Facebook](<https://www.facebook.com/alexanderof/>) | [WhatsApp](<https://api.whatsapp.com/send?phone=573015930519&text;=Hola%20!Quiero%20conversar%20contigo!>) | [LinkedIn](<https://www.linkedin.com/in/alexander-oviedo-fadul-49434b9a/>)

## Licencia

Este proyecto está licenciado bajo la Licencia MIT - vea el archivo [MIT License](<https://opensource.org/licenses/MIT>) para más detalles.

![[Logo de Removedor de Fondo de Imágenes]](<https://raw.githubusercontent.com/bladealex9848/ImageBackgroundRemover/main/logo.png>)

# Removedor de Fondo de Imágenes

Una aplicación web sencilla y fácil de usar que permite a los usuarios eliminar el fondo de las imágenes automáticamente.

## Características

- Subida sencilla de imágenes en formatos JPG, JPEG y PNG.
- Vista previa de la imagen con y sin fondo.
- Opción para descargar la imagen sin fondo.

## Instrucciones de Uso

1. Visita la aplicación web.
2. Carga la imagen de la que deseas eliminar el fondo.
3. Espera a que la aplicación procese la imagen.
4. Descarga la imagen resultante sin fondo.

## Instalación Local

Si prefieres ejecutar la aplicación localmente, sigue estos pasos:

1. Clona este repositorio.
2. Instala las dependencias con `pip install -r requirements.txt`.
3. Ejecuta la aplicación con `streamlit run app.py`.

## Contribuciones

Las contribuciones son siempre bienvenidas. No dudes en enviar un Pull Request o abrir un Issue si tienes alguna sugerencia o identificas algún error.

## Licencia

Este proyecto está licenciado bajo la [Licencia MIT](LICENSE) - mira el archivo LICENSE para más detalles.

## Créditos

Creado por Alexander Oviedo Fadul. Encuéntrame en: [GitHub](<https://github.com/bladealex9848>), [Website](<https://alexander.oviedo.isabellaea.com/>), [Instagram](<https://www.instagram.com/alexander.oviedo.fadul>), [Twitter](<https://twitter.com/alexanderofadul>), [Facebook](<https://www.facebook.com/alexanderof/>), [WhatsApp](<https://api.whatsapp.com/send?phone=573015930519&text;=Hola%20!Quiero%20conversar%20contigo!%20>).

# JudiCalc: Calculadora de Días Hábiles para la Rama Judicial de Colombia

## Tabla de Contenidos

1. [Descripción](#descripción)
2. [Características](#características)
3. [Estructura del Proyecto](#estructura-del-proyecto)
4. [Instalación](#instalación)
5. [Uso](#uso)
6. [Manual Técnico](#manual-técnico)
7. [Manual de Usuario](#manual-de-usuario)
8. [Contribución](#contribución)
9. [Registro de Cambios](#registro-de-cambios)
10. [Créditos](#créditos)
11. [Licencia](#licencia)

## Descripción

JudiCalc es una herramienta web interactiva diseñada específicamente para funcionarios y administrativos de la Rama Judicial de Colombia. Esta aplicación permite calcular con precisión los días hábiles entre dos fechas dadas, teniendo en cuenta los fines de semana, días festivos nacionales y las particularidades de diferentes especialidades judiciales.

## Características Principales

- **Cálculo Preciso de Días Hábiles**: Considera fines de semana y festivos nacionales.
- **Adaptación a Especialidades Judiciales**: Incluye cálculos para Colectiva, Individual Penal, Ejecución de penas y promiscuos de familia.
- **Visualización de Días Hábiles por Mes**: Muestra una tabla detallada por especialidad.
- **Interfaz de Usuario Intuitiva**: Desarrollada con Streamlit para una experiencia web fluida.
- **Capacidad de Adaptación Anual**: Actualmente configurado para 2023, con posibilidad de expansión.

## Estructura del Proyecto

...

JudiCalc/

■

■■■ app.py

■■■ test\_judicialc.py

■

■■■ assets/

■ ■■■ logo.png

■ ■■■ holidays\_2023.json

■ ■■■ dias\_habiles\_2023.csv

■

■■■ requirements.txt

■■■ .gitignore

■■■ README.md



...

## Instalación

1. Clonar el repositorio:

...

```
git clone https://github.com/tu-usuario/JudiCalc.git
```

...

2. Navegar al directorio del proyecto:

...

```
cd JudiCalc
```

...

3. Crear un entorno virtual:

...

```
python -m venv venv
```

```
source venv/bin/activate # En Windows use `venv\Scripts\activate`
```

...

4. Instalar las dependencias:

...

```
pip install -r requirements.txt
```

...

## Uso

Para ejecutar la aplicación:

...

```
streamlit run app.py
```

...

Siga las instrucciones en la interfaz de usuario para seleccionar fechas y calcular días hábiles.

### Componentes Principales

#### 1. Función `load\_data(year)`

- Carga los días festivos desde un archivo JSON.
- Lee el CSV de días hábiles y lo convierte en un diccionario.
- Maneja errores si los archivos no se encuentran.

#### 2. Función `calculate\_working\_days(start\_date, end\_date, specialty, working\_days)`

- Calcula los días hábiles entre dos fechas para una especialidad específica.
- Utiliza los datos del CSV para determinar los días hábiles por mes.
- Ajusta el cálculo para meses parciales al inicio y final del período.

#### 3. Interfaz de Usuario (Streamlit)

- Configura la página y el sidebar.
- Maneja la selección de fechas y especialidad.
- Muestra resultados y tabla de días hábiles.

### Flujo de Datos

1. El usuario selecciona el año, las fechas y la especialidad.
2. Se cargan los datos de días festivos y hábiles.
3. Se calcula el número de días hábiles.
4. Se muestra el resultado y la tabla de días hábiles por mes.

### Manejo de Errores

- Validación de fechas de entrada.
- Manejo de archivos no encontrados con mensajes de advertencia.
- Uso de datos por defecto si no se encuentran los archivos necesarios.

### Consideraciones de Rendimiento

- La aplicación está optimizada para cálculos rápidos utilizando operaciones vectorizadas de Pandas.

- Los datos se cargan una sola vez al inicio, mejorando el tiempo de respuesta para cálculos subsiguientes.

#### ### Seguridad

- No se almacenan datos sensibles del usuario.
- Los archivos de datos están en formato JSON y CSV, que son fáciles de auditar.

#### ### Mantenimiento y Actualización

Para añadir soporte para un nuevo año:

1. Crear `holidays\_[AÑO].json` en la carpeta `assets`.
2. Crear `dias\_habiles\_[AÑO].csv` en la carpeta `assets`.
3. Actualizar la lista de años disponibles en el selectbox del sidebar.

## Manual de Usuario

#### ### Introducción

JudiCalc es una herramienta diseñada para calcular días hábiles en el contexto judicial colombiano. Permite a los usuarios determinar el número de días hábiles entre dos fechas, considerando diferentes especialidades judiciales.

#### ### Acceso a la Aplicación

1. Abra su navegador web.
2. Visite la URL: [https://judicalc.streamlit.app](https://judicalc.streamlit.app)

#### ### Uso de la Aplicación

##### #### 1. Selección del Año

- En el sidebar, elija el año para el cual desea realizar el cálculo.

##### #### 2. Elección de la Especialidad

- También en el sidebar, seleccione la especialidad judicial relevante:

- Colectiva
- Individual Penal
- Individual Ejecución de penas y promiscuos de familia

##### #### 3. Selección de Fechas

- En el área principal, use los selectores de fecha para elegir:

- Fecha de inicio
- Fecha de fin

##### #### 4. Cálculo de Días Hábiles

- Una vez seleccionadas las fechas, la aplicación automáticamente calculará y mostrará el número de días hábiles.

##### #### 5. Visualización de Resultados

- El resultado se muestra en un mensaje de éxito, indicando el número de días hábiles entre las fechas seleccionadas.

- Debajo del resultado, se muestra una tabla con el desglose de días hábiles por mes para el año seleccionado.

#### ### Funcionalidades Adicionales

##### #### Carga de Archivo de Días Festivos Personalizado

- En el sidebar, puede cargar un archivo JSON con días festivos personalizados.
- Útil para casos especiales o para probar diferentes escenarios.

##### #### Recursos Adicionales

- En el sidebar, expanda la sección "Recursos Adicionales" para acceder a:

- Manual de Usuario
- Formulario para Reportar Problemas

#### ### Interpretación de Resultados

- El número de días hábiles mostrado excluye fines de semana y días festivos.
- La tabla de días hábiles por mes muestra el desglose para cada especialidad, útil para comparaciones y verificaciones.

#### ### Solución de Problemas Comunes

1. **\*\*Las fechas no se pueden seleccionar\*\***: Asegúrese de que la fecha de inicio sea anterior a la fecha de fin.
2. **\*\*El cálculo parece incorrecto\*\***: Verifique la especialidad seleccionada y el año, ya que los días hábiles pueden variar.
3. **\*\*Error al cargar archivo personalizado\*\***: Asegúrese de que el archivo esté en formato JSON válido.

### ### Contacto y Soporte

Para obtener ayuda adicional o reportar problemas:

- Visite nuestra [página de GitHub](https://github.com/bladealex9848/JudiCalc/issues)
- Contacte al desarrollador a través de los enlaces proporcionados en el sidebar de la aplicación.

## Contribución

Las contribuciones son bienvenidas. Por favor, abra un issue para discutir cambios mayores antes de hacer un pull request.

## Registro de Cambios

- 2023-08-12: Versión inicial (v1.0.0)
- Implementación del cálculo de días hábiles para el año 2023
- Interfaz web con Streamlit

## Créditos

Desarrollado y mantenido por Alexander Oviedo Fadul, Profesional Universitario Grado 11 en el Consejo Seccional de la Judicatura de Sucre.

[GitHub](https://github.com/bladealex9848) | [Website](https://alexander.oviedo.isabellaea.com/) |

[Instagram](https://www.instagram.com/alexander.oviedo.fadul) |

[Twitter](https://twitter.com/alexanderofadul) | [Facebook](https://www.facebook.com/alexanderof/) | [WhatsApp](https://api.whatsapp.com/send?phone=573015930519&text;=Hola%20!Quiero%20conversar%20contigo!) | [LinkedIn](https://www.linkedin.com/in/alexander-oviedo-fadul-49434b9a/)

## Licencia

Este proyecto está licenciado bajo la Licencia MIT - vea el archivo [MIT License](https://opensource.org/licenses/MIT) para más detalles.

! [Logo de MALLO] (<https://raw.githubusercontent.com/bladealex9848/MALLO/main/assets/logo.jpg>)

# MALLO: MultiAgent LLM Orchestrator

## Tabla de Contenidos

1. [Descripción](#descripción)
2. [Características Principales](#características-principales)
3. [Estructura del Proyecto](#estructura-del-proyecto)
4. [Requisitos Previos](#requisitos-previos)
5. [Instalación](#instalación)
6. [Configuración](#configuración)
7. [Uso](#uso)
8. [Arquitectura del Sistema](#arquitectura-del-sistema)
9. [Componentes Principales](#componentes-principales)
10. [Flujo de Trabajo](#flujo-de-trabajo)
11. [APIs y Servicios Integrados](#apis-y-servicios-integrados)
12. [Manejo de Errores y Logging](#manejo-de-errores-y-logging)
13. [Optimización y Caché](#optimización-y-caché)
14. [Pruebas](#pruebas)
15. [Contribución](#contribución)
16. [Registro de Cambios](#registro-de-cambios)
17. [Roadmap](#roadmap)
18. [Licencia](#licencia)
19. [Contacto](#contacto)

## Descripción

MALLO (MultiAgent LLM Orchestrator) es un sistema avanzado de orquestación de múltiples agentes de Modelos de Lenguaje de Gran Escala (LLMs). Diseñado para manejar consultas complejas, MALLO utiliza una variedad de agentes, desde modelos locales hasta asistentes especializados y APIs de terceros, para proporcionar respuestas precisas y contextuales.

El sistema está construido con un enfoque modular y flexible, permitiendo la integración de diversos modelos de lenguaje y servicios. MALLO evalúa la complejidad de cada consulta y selecciona dinámicamente el agente o combinación de agentes más apropiados para procesarla, optimizando así la calidad y relevancia de las respuestas.

## Características Principales

- **Integración Multifacética de LLMs**:
  - Modelos locales de Ollama para procesamiento rápido y offline.
  - APIs de OpenAI para tareas que requieren capacidades avanzadas.
  - Integración con otros proveedores de LLM como Groq, Together, DeepInfra, Anthropic, DeepSeek, Mistral y Cohere.
- **Asistentes Especializados**:
  - Conjunto de asistentes entrenados para dominios específicos como derecho, ética, tecnología, entre otros.
- **Búsqueda Web Integrada**:
  - Utilización de DuckDuckGo para enriquecer las respuestas con información actualizada de la web.
- **Procesamiento Avanzado de Texto**:
  - Análisis de complejidad de consultas.
  - Resumen automático de textos largos.

- Extracción de palabras clave y entidades nombradas.
- **\*\*Manejo Inteligente de Recursos\*\***:
- Selección dinámica de agentes basada en la complejidad de la consulta y disponibilidad de recursos.
- Sistema de caché para optimizar el rendimiento y reducir la carga en las APIs.
- **\*\*Interfaz de Usuario Intuitiva\*\***:
- Implementada con Streamlit para una experiencia de usuario fluida y responsive.
- **\*\*Evaluación y Mejora Continua\*\***:
- Sistema de evaluación de respuestas para garantizar la calidad y relevancia.
- Capacidad de aprendizaje y mejora basada en el feedback del usuario.
- **\*\*Logging y Monitoreo\*\***:
- Sistema robusto de logging para seguimiento de errores y rendimiento.
- **\*\*Configuración Flexible\*\***:
- Archivo de configuración YAML para fácil ajuste y personalización del sistema.

## Estructura del Proyecto

```

...
MALLO/
■
■■■ agents.py # Implementación de la clase AgentManager y lógica de agentes
■■■ config.yaml # Configuración general del sistema
■■■ main.py # Punto de entrada principal y UI de Streamlit
■■■ README.md # Documentación del proyecto (este archivo)
■■■ utilities.py # Funciones de utilidad y helpers
■■■ model_speeds.json # Índice de velocidad de modelos locales y en la nube
■■■ CHANGELOG.md # Registro de cambios y versiones
■
■■■ .streamlit/
■ ■■■ secrets.toml # Almacenamiento seguro de claves API (no incluido en el repositorio)
■
■■■ tests/ # Directorio para pruebas unitarias y de integración
■
■■■ logs/ # Directorio para archivos de log
■
■■■ assets/ # Directorio para recursos estáticos (imágenes, estilos, etc.)
|
■■■ tools/ # Directorio utilidades externas
■ ■■■ test_model_speeds.py # Pequeño script para medir la velocidad de los modelos
|
■■■ requirements.txt # Dependencias del proyecto
...

```

## Requisitos Previos

- Python 3.8+
- Ollama instalado localmente (para modelos locales)
- Claves API para servicios externos (OpenAI, Groq, Together, etc.)
- Conexión a Internet (para búsqueda web y APIs externas)

## Instalación

1. Clona el repositorio:

```

...

```

```
git clone https://github.com/bladealex9848/MALLO.git
cd MALLO
...
```

2. Crea y activa un entorno virtual:

```
python3 -m venv venv
source venv/bin/activate # En Windows usa `venv\Scripts\activate`
...
```

3. Instala las dependencias:

```
pip install -r requirements.txt
...
```

4. Configura tus claves API:

Crea un archivo ``.streamlit/secrets.toml`` y añade tus claves API:

```
```toml
OPENAI_API_KEY = "tu_clave_openai_aqui"
GROQ_API_KEY = "tu_clave_groq_aqui"
TOGETHER_API_KEY = "tu_clave_together_aqui"
DEEPINFRA_API_KEY="tu_clave_deepinfra_aqui"
ANTHROPIC_API_KEY="tu_clave_anthropic_aqui"
DEEPSEEK_API_KEY="tu_clave_deepseek_aqui"
MISTRAL_API_KEY="tu_clave_mistral_aqui"
COHERE_API_KEY="tu_clave_cohere_aqui"
# Añade otras claves API según sea necesario
```
```

## Configuración

El archivo `config.yaml` contiene la configuración principal del sistema. Aquí puedes ajustar:

- Modelos disponibles para cada proveedor de LLM
- Prioridades de procesamiento
- Umbrales de complejidad para la selección de agentes
- Configuración de asistentes especializados
- Parámetros de utilidades como búsqueda web y análisis de imágenes

Ejemplo de configuración:

```
```yaml
ollama:
  base_url: "http://localhost:11434"
  default_model: "gemma2:2b"
  models:
    - "gemma2:2b"
    - "llava:latest"
openai:
  default_model: "gpt-4o-mini"
  models:
    - "gpt-4o-mini"
```
```

## ... (otras configuraciones)

```
thresholds:
  moa_complexity: 0.8
  local_complexity: 0.5
  web_search_complexity: 0.4
```

...

## Uso

Para iniciar la aplicación, ejecuta:

...

```
streamlit run main.py
```

...

Luego, abre tu navegador y ve a `http://localhost:8501`.

La interfaz de usuario te permitirá:

1. Ingresar consultas en lenguaje natural.
2. Ver las respuestas generadas por el sistema.
3. Explorar detalles sobre el procesamiento de cada consulta.
4. Acceder a información sobre el estado del sistema y rendimiento de los agentes.

## Arquitectura del Sistema

MALLO utiliza una arquitectura de microservicios basada en agentes, donde cada agente (modelo de lenguaje o servicio especializado) puede procesar consultas de manera independiente. El componente central, `AgentManager`, orquesta estos agentes basándose en la complejidad de la consulta y la disponibilidad de recursos.

### Componentes Clave:

1. **Orquestador de Agentes**: Selecciona y gestiona los agentes apropiados para cada consulta.
2. **Evaluador de Complejidad**: Analiza las consultas para determinar su complejidad y requisitos.
3. **Sistema de Caché**: Almacena y recupera respuestas para consultas frecuentes.
4. **Módulo de Búsqueda Web**: Enriquece las respuestas con información actual de la web.
5. **Interfaz de Usuario**: Proporciona una experiencia interactiva para los usuarios.

## Componentes Principales

### agents.py

- **Clase AgentManager**:
  - Gestiona la inicialización y selección de agentes.
  - Implementa la lógica de procesamiento de consultas.
  - Maneja la integración con diferentes APIs de LLM.

### utilities.py

- Funciones para evaluación de complejidad de consultas.
- Implementación de búsqueda web.
- Sistema de caché para respuestas.
- Funciones de logging y manejo de errores.

### main.py

- Implementa la interfaz de usuario con Streamlit.
- Gestiona el flujo principal de la aplicación.
- Procesa las entradas del usuario y muestra las respuestas.

## Flujo de Trabajo

1. El usuario ingresa una consulta a través de la interfaz de Streamlit.
2. La consulta se evalúa para determinar su complejidad y requisitos.
3. Se selecciona el agente o conjunto de agentes más apropiados.
4. Si es necesario, se realiza una búsqueda web para enriquecer el contexto.
5. Los agentes seleccionados procesan la consulta.
6. Se evalúa y refina la respuesta generada.

7. La respuesta final se presenta al usuario junto con detalles del proceso.

## APIs y Servicios Integrados

MALLO integra varios servicios de LLM y APIs, incluyendo:

- OpenAI
- Groq
- Together
- DeepInfra
- Anthropic
- DeepSeek
- Mistral
- Cohere

Cada servicio se inicializa y gestiona a través de la clase ``AgentManager``, permitiendo una fácil expansión a nuevos proveedores en el futuro.

## Manejo de Errores y Logging

El sistema implementa un robusto sistema de logging y manejo de errores:

- Los errores se registran en el archivo ``mallo.log``.
- Se utilizan diferentes niveles de logging (INFO, WARNING, ERROR) para categorizar los eventos.
- Los errores críticos se muestran al usuario a través de la interfaz de Streamlit.

## Optimización y Caché

Para mejorar el rendimiento y reducir la carga en las APIs externas, MALLO implementa:

- Un sistema de caché para almacenar respuestas frecuentes.
- Evaluación de complejidad para evitar el uso innecesario de recursos en consultas simples.
- Selección inteligente de agentes basada en la disponibilidad y rendimiento histórico.

## Pruebas

El proyecto incluye un conjunto de pruebas unitarias y de integración en el directorio ``tests/``. Para ejecutar las pruebas:

```
'''
```

```
python -m unittest discover tests
```

```
'''
```

## Contribución

Las contribuciones son bienvenidas. Por favor, sigue estos pasos:

1. Haz fork del repositorio.
2. Crea una nueva rama (``git checkout -b feature/AmazingFeature``).
3. Realiza tus cambios y haz commit (``git commit -m 'Add some AmazingFeature'``).
4. Push a la rama (``git push origin feature/AmazingFeature``).
5. Abre un Pull Request.

## Registro de Cambios

Consulta [CHANGELOG.md](CHANGELOG.md) para ver el historial detallado de cambios del proyecto.



## Roadmap

- Implementación de más modelos de lenguaje y APIs.
- Mejora del sistema de evaluación y selección de agentes.
- Desarrollo de una API REST para integración con otros sistemas.
- Implementación de fine-tuning para modelos específicos de dominio.

## Licencia

Este proyecto está licenciado bajo la Licencia MIT - vea el archivo [LICENSE.md](LICENSE.md) para más detalles.

## Contacto

Alexander Oviedo Fadul - alexander.oviedo.fadul@gmail.com

Enlace del proyecto:

[<https://github.com/bladealex9848/MALLO>](https://github.com/bladealex9848/MALLO)

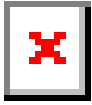
[GitHub](https://github.com/bladealex9848) | [Website](https://alexander.oviedo.isabellaea.com/) |

[Instagram](https://www.instagram.com/alexander.oviedo.fadul) |

[Twitter](https://twitter.com/alexanderofadul) | [Facebook](https://www.facebook.com/alexanderof/) | [W

hatsApp](https://api.whatsapp.com/send?phone=573015930519&text;=Hola%20!Quiero%20conversar%20contigo!) | [LinkedIn](https://www.linkedin.com/in/alexander-oviedo-fadul-49434b9a/)

# OmniChat: Asistente Virtual Todo en Uno



OmniChat es un asistente virtual versátil desarrollado con Langchain y Streamlit. Aprovecha el poder de los Modelos de Lenguaje (LLMs) para ofrecer una amplia gama de funcionalidades, simplificando la interacción con diversos tipos de información y bases de datos.

## Funcionalidades de OmniChat

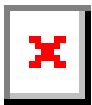
OmniChat ofrece las siguientes capacidades:

- **Chatbot Básico:**  
Mantén conversaciones interactivas con el LLM.
- **Chatbot Consciente del Contexto:**  
Un asistente que recuerda conversaciones previas y proporciona respuestas acordes.
- **Chat con Documentos:**  
Permite al chatbot acceder a documentos personalizados, respondiendo preguntas basadas en la información contenida.
- **Chat con Base de Datos SQL:**  
Interactúa con bases de datos SQL mediante comandos conversacionales simples.
- **Chat con Sitios Web:**  
Permite al chatbot interactuar con contenidos de sitios web.

 `` Aplicación Streamlit

OmniChat es una aplicación multi-página desarrollada con Streamlit, que incluye todas las funcionalidades mencionadas.

Accede a la aplicación aquí: [OmniChat en Streamlit](#)



## Requisitos

- Python 3.10

## Configuración del Entorno Virtual (Python 3.10)

### 1. Instalación de Python 3.10:

#### \* Windows:

- Descarga el instalador de Python 3.10 desde <https://www.python.org/downloads/release/python-3109/> y asegúrate de marcar la casilla "Add Python 3.10 to PATH".

#### \* macOS (usando Homebrew):

- Instala Homebrew si aún no lo tienes: `/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"`
- Luego instala Python 3.10: `brew install python@3.10`

#### \* Linux (usando el gestor de paquetes de tu distribución):

- Ejemplo para Debian/Ubuntu: `sudo apt-get install python3.10 python3.10-venv`

### 2. Creación y Activación del Entorno Virtual:

- \* Abre una terminal en la carpeta de tu proyecto.
- \* Crea el entorno virtual: `python3.10 -m venv mi_entorno_310`
- \* Activa el entorno virtual:
  - Windows: `mi_entorno_310\Scripts\activate`
  - macOS/Linux: `source mi_entorno_310/bin/activate`

### 3. Actualización de pip:

```
pip install --upgrade pip
```

### 4. Instalación de Dependencias:

```
pip install -r requirements.txt
```

## Ejecución Local

### Ejecutar la aplicación principal de Streamlit

```
$ python3 -m streamlit run Inicio.py # Si tienes Python 3.10 instalado en un entorno virtual
```

```
$ streamlit run Inicio.py # Si tienes Streamlit instalado globalmente
```

## Ejecución con Docker

### Generar la imagen

```
$ docker build -t omnichat .
```

## Ejecutar el contenedor Docker

```
$ docker run -p 8501:8501 omnichat
```



## Contribuciones

Planeamos añadir más funcionalidades a OmniChat con el tiempo. Las contribuciones son bienvenidas.



## Licencia

Este proyecto está bajo la licencia MIT. Consulta el archivo [LICENSE](#) para obtener más detalles.

![[Logo de PDFMaster]](<https://raw.githubusercontent.com/bladealex9848/PDFMaster/main/img/logo.png>)

# PDFMaster: Tu asistente de documentos PDF

## Descripción

PDFMaster es una herramienta completa para gestionar documentos PDF. Permite realizar diversas tareas como convertir PDF a Word, generar resúmenes, realizar preguntas y obtener respuestas específicas de un documento, y muchas otras funcionalidades que se están desarrollando.

## Introducción

Con PDFMaster, en menos de 50 líneas de código, aprenderás a aprovechar todo el potencial de chatGPT y convertir tus documentos en conversaciones interactivas. No más lecturas aburridas o búsquedas tediosas, ahora podrás hacer preguntas directamente a tus documentos y obtener respuestas de chatGPT. Para desarrollar esta aplicación necesitaremos:

- OpenAI ChatGPT API
- Streamlit

## ¿Cómo funciona?

1. Divide documento en fragmentos (o chunks)
2. Crea los embeddings de los fragmentos de texto
3. Guarda los fragmentos y los embeddings en una base de conocimiento
4. Busca los fragmentos más similares a la pregunta del usuario gracias a los embeddings.
5. Pasa los fragmentos más similares junto a la pregunta a chatGPT que genera la respuesta

## Funcionalidades

- **Realizar preguntas**: Permite hacer preguntas específicas sobre el contenido de un documento PDF y obtener respuestas precisas.
- **Convertir a Word**: Convierte documentos PDF en documentos Word (.docx).
- **Generar resumen**: Genera un resumen del contenido de un documento PDF.

## Funcionalidades Futuras

- Comprimir PDF
- Aplicar OCR (Reconocimiento óptico de caracteres) a un PDF
- ... y más.

## Instalación

1. Asegúrate de tener Python 3.8 o superior instalado en tu máquina.
2. Clona este repositorio usando `git clone https://github.com/bladealex9848/PDFMaster.git`
3. Navega al directorio del proyecto `cd PDFMaster`
4. Instala las dependencias necesarias usando `pip install -r requirements.txt`
5. Ejecuta `streamlit run app.py` para iniciar la aplicación.
6. Obtén una clave API de OpenAI para usar su API ChatGPT.

## Uso

1. **Realizar preguntas**

- Carga un documento PDF.
- Selecciona la opción 'Realizar preguntas' en el menú de la izquierda.
- Escribe tu pregunta en el cuadro de texto.
- Presiona 'Enter' o haz clic fuera del cuadro de texto para obtener la respuesta.

2. **Convertir a Word**

- Carga un documento PDF.
- Selecciona la opción 'Convertir a Word' en el menú de la izquierda.
- Haz clic en 'Convertir' y espera a que la conversión se complete.
- Haz clic en 'Descargar archivo Word' para descargar el archivo convertido.

3. **Generar resumen**

- Carga un documento PDF.
- Selecciona la opción 'Generar resumen' en el menú de la izquierda.
- El resumen se generará automáticamente y se mostrará en pantalla.

## **Contribuciones**

Si deseas contribuir a este proyecto, por favor, realiza un fork de este repositorio, crea una nueva rama, realiza tus cambios y envía un pull request.

## **Licencia**

Este proyecto está bajo la licencia MIT. Consulta el archivo `LICENSE` para obtener más detalles.



Consejo Superior de la Judicatura  
Unidad de Desarrollo y Análisis Estadístico  
REPORTE DE ACCIONES PREVENTIVAS CORRECTIVAS Y DE MEJORA

|   |  |   |   |  |   |
|---|--|---|---|--|---|
| PROCESO:  | Comunicación Institucional   |   | Fecha:  | 17/07/2023   |   |
| Nivel Central   | <input type="checkbox"/>   | Nivel Seccional   | <input type="checkbox"/>  | Nombre de la dependencia                           | Concejo Seccional del Judicatura de Sucre |
| HALLAZGO  | <input type="checkbox"/>   | OPORTUNIDAD   | <input type="checkbox"/>  | ACCIÓN DE GESTIÓN                                  | <input checked="" type="checkbox"/>       |
| FUENTE DE INFORMACION:  | <input type="checkbox"/>   | Análisis de indicadores                                 | <input checked="" type="checkbox"/>   | Auditoría externa                                  | <input type="checkbox"/>                  |
| Auditoría interna   | <input type="checkbox"/>   | Evaluación de la satisfacción de las partes interesadas | <input checked="" type="checkbox"/>   | Revisión del SIGCMA por parte de la Alta Dirección | <input checked="" type="checkbox"/>       |
| Salida no conforme  | <input type="checkbox"/>   | Quejas, reclamos y sugerencias                          | <input type="checkbox"/>  | Valoración de los riesgos y/o peligros             | <input type="checkbox"/>                  |
| Inspecciones seguridad  | <input type="checkbox"/>   | Análisis vulnerabilidad de emergencias                  | <input type="checkbox"/>  | Otro ¿Cual?: Control Interno                       | <input type="checkbox"/>                  |
| DESCRIPCIÓN DEL HALLAZGO, OPORTUNIDAD O ACCIÓN  |  |   |   |  |   |
| Se ha identificado la oportunidad de implementar un sistema de Inteligencia Artificial (IA) para mejorar el control de legalidad de las solicitudes de vigilancia administrativa judicial. Actualmente, este proceso se realiza manualmente, lo que puede resultar en tiempos de respuesta prolongados y posibles inconsistencias en la evaluación de las solicitudes. La implementación de un sistema de IA, como PDFMaster, permitiría automatizar gran parte de este proceso, mejorando la precisión y la celeridad en la respuesta a estas solicitudes. |  |   |   |  |   |
| CORRECCION (Solo para hallazgos)  |  |   | Aplica  |  | <input type="checkbox"/>                  |
|   |  |   | No Aplica   |  | <input type="checkbox"/>                  |
| Actividades   | Responsable de la actividad  |   | Fecha ejecución   | Seguimiento  |   |
| 1.  |  |   |   |  |   |
| 2.  |  |   |   |  |   |
| 3.  |  |   |   |  |   |
| ACCION PREVENTIVA   | <input type="checkbox"/>   | ACCION CORRECTIVA                                       | <input type="checkbox"/>  | ACCION DE MEJORA                                   | <input checked="" type="checkbox"/>       |
| CAUSAS DEL HALLAZGO, OPORTUNIDAD O ACCIÓN   |  |   |   |  |   |
| 1. Proceso manual de revisión de solicitudes que consume mucho tiempo.<br>2. Posible inconsistencia en la aplicación de criterios de evaluación entre diferentes revisores.<br>3. Volumen creciente de solicitudes que sobrepasa la capacidad de procesamiento manual.<br>4. Necesidad de mejorar la eficiencia y precisión en el control de legalidad.<br>5. Oportunidad de aprovechar las nuevas tecnologías de IA para optimizar procesos administrativos.   |  |   |   |  |   |
| DESCRIPCIÓN DE ACTIVIDADES PARA GESTIONAR LAS CAUSAS  |  |   |   |  |   |
| Actividades a realizar  | Responsable de la actividad  |   | Fecha inicio  | Fecha fin  |   |
| 1. Adaptar PDFMaster para el procesamiento específico de solicitudes de vigilancia administrativa judicial.   | Alexander Oviedo Fadul, Profesional Universitario Grado 11 del Consejo Seccional de la Judicatura de Sucre |   | 3/07/2023   | 29/09/2023   |   |
| 2. Desarrollar un módulo de IA para el análisis de legalidad basado en el ACUERDO No. PSAA11-8716.  | Alexander Oviedo Fadul, Profesional Universitario Grado 11 del Consejo Seccional de la Judicatura de Sucre |   | 2/10/2023   | 1/11/2023  |   |
| 3. Realizar pruebas piloto del sistema en 2 despachos.  | Alexander Oviedo Fadul, Profesional Universitario Grado 11 del Consejo Seccional de la Judicatura de Sucre |   | 2/11/2023   | 31/07/2024   |   |
| 4. Capacitar a 5 empleados en el uso del nuevo sistema.   | Alexander Oviedo Fadul, Profesional Universitario Grado 11 del Consejo Seccional de la Judicatura de Sucre |   | 1/08/2024   | 30/08/2024   |   |
| 5. Implementar el sistema en todos los despachos del Consejo Seccional de Sucre.  | Alexander Oviedo Fadul, Profesional Universitario Grado 11 del Consejo Seccional de la Judicatura de Sucre |   | 2/09/2024   | 30/09/2024   |   |
| SEGUIMIENTO   |  |   |   |  |   |
| Seguimiento de las actividades realizadas   |  |   | Responsable del Seguimiento   |  | Fecha Seguimiento                         |
| 1. Adaptación de PDFMaster: 50% completado al 29/09/2023. Se ha logrado la integración con el sistema de gestión documental existente.  |  |   | Rozana Beatriz Abello Albino<br>Magistrada del Consejo Seccional de la Judicatura de Sucre<br>Despacho 02 |  | 29/09/2023                                |

| PROCESO:   | Comunicación Institucional | Fecha:   | 17/07/2023 |
|--|----------------------------|--|------------|
| 2. Desarrollo del módulo de IA: 30% completado al 01/11/2023. Se ha finalizado el entrenamiento inicial del modelo con el ACUERDO No. PSAA11-8716. |                            | Rozana Beatriz Abello Albino<br>Magistrada del Consejo Seccional de la Judicatura de Sucre<br>Despacho 02          | 1/11/2023  |
| 3. Pruebas piloto: No iniciado. Programado para iniciar el 31/07/2024.   |                            | Maria Claudia Lozano Acosta,<br>Profesional Universitario Grado 11 del Consejo Seccional de la Judicatura de Sucre | 31/07/2024 |
| 4. Capacitación: No iniciado. Programado para iniciar el 30/08/2024.   |                            | Maria Claudia Lozano Acosta,<br>Profesional Universitario Grado 11 del Consejo Seccional de la Judicatura de Sucre | 30/08/2024 |
| 5. Implementación general: No iniciado. Programado para iniciar el 30/09/2024.   |                            | Maria Claudia Lozano Acosta,<br>Profesional Universitario Grado 11 del Consejo Seccional de la Judicatura de Sucre | 30/09/2024 |

| REVISIÓN  |                                     |   |
|---|-------------------------------------|---|
| Las actividades tomadas:  | SI                                  | NO  |
| Fueron adecuadas?   | <input checked="" type="checkbox"/> | <input type="checkbox"/>  |
| Fueron Convenientes?  | <input checked="" type="checkbox"/> | <input type="checkbox"/>  |
| Fueron eficaces?  | <input type="checkbox"/>            | <input type="checkbox"/>  |
| REVISÓ: Responsable del proceso o enlace seccional: Nombre, Cargo y Firma |                                     | Rozana Beatriz Abello Albino Magistrada del Consejo Seccional de la Judicatura de Sucre Despacho 02 |

|                      |   |                                  |
|----------------------|---|----------------------------------|
| Estado de la Acción: | Abierta <input checked="" type="checkbox"/> | Cerrada <input type="checkbox"/> |
|----------------------|---|----------------------------------|

| PARA USO EXCLUSIVO DEL COORDINADOR DE MEJORAMIENTO |                       |                      |
|--|-----------------------|----------------------|
| ACCION PREVENTIVA No.                              | ACCION CORRECTIVA No. | ACCION DE MEJORA No. |

Las actividades se consideran adecuadas y convenientes hasta el momento, aunque es prematuro determinar su eficacia. Son adecuadas porque abordan directamente la necesidad de mejorar el proceso de control de legalidad de las solicitudes de vigilancia administrativa judicial. Son convenientes porque la inversión en tecnología de IA promete una mejora significativa en la eficiencia y precisión del proceso. Sin embargo, la eficacia solo podrá determinarse una vez que el sistema esté completamente implementado y se puedan medir sus resultados en términos de reducción de tiempos de procesamiento y mejora en la precisión de las evaluaciones de legalidad.

| CÓDIGO   | ELABORÓ                      | REVISÓ    | APROBÓ                           |
|----------|------------------------------|-----------|----------------------------------|
| F-ESG-05 | COORDINACIÓN NACIONAL SIGCMA | SIGCMA    | COORDINACIÓN NACIONAL DEL SIGCMA |
| VERSIÓN  | FECHA                        | FECHA     | FECHA                            |
| 5        | 11/06/2022                   | 1/06/2023 | 12/07/2023                       |



![[Logo de Monitoreo de Presión

Arterial]]([https://raw.githubusercontent.com/bladealex9848/Presion\\_Arterial/main/img/logo.png](https://raw.githubusercontent.com/bladealex9848/Presion_Arterial/main/img/logo.png))

# Monitoreo de Presión Arterial

Aplicación para el seguimiento de la presión arterial, permitiendo el registro y visualización de mediciones de presión sistólica y diastólica.

## Características

- Registro de pacientes y mediciones.
- Visualización de datos históricos.
- Generación de gráficos para seguimiento.
- Diagnósticos y recomendaciones basadas en las mediciones.

## Sub-aplicación: Gestor de Base de Datos SQLite

Esta herramienta integrada proporciona un entorno de desarrollo interactivo (IDE) para administrar la base de datos del proyecto. Permite realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) directamente en las tablas de la base de datos SQLite, facilitando la gestión de datos y la resolución de problemas.

### Funcionalidades del Gestor de Base de Datos

- Interfaz intuitiva para realizar operaciones de base de datos.
- Capacidad para insertar, actualizar y eliminar registros de cualquier tabla.
- Visualización de la estructura y datos actuales de la base de datos.

### Uso del Gestor de Base de Datos

Para iniciar el gestor de base de datos, ejecute:

```
'''
```

```
streamlit run db_manager.py
```

```
'''
```

Siga las instrucciones en la interfaz para seleccionar la base de datos y la tabla con la que desea interactuar.

## Instalación

Para instalar las dependencias del proyecto, ejecute:

```
'''
```

```
pip install -r requirements.txt
```

```
'''
```

## Uso

Para iniciar la aplicación, ejecute:

```
'''
```

```
streamlit run app.py
```

```
'''
```

## Contribuciones

Las contribuciones son siempre bienvenidas. No dudes en enviar un Pull Request o abrir un Issue si tienes alguna sugerencia o identificas algún error.

## Licencia

Este proyecto está licenciado bajo la [Licencia MIT](LICENSE) - mira el archivo LICENSE para más detalles.

## Créditos

Creado por Alexander Oviedo Fadul. Encuéntrame en: [GitHub](https://github.com/bladealex9848), [Website](https://alexander.oviedo.isabellaea.com/), [Instagram](https://www.instagram.com/alexander.oviedo.fadul), [Twitter](https://twitter.com/alexanderofadul), [Facebook](https://www.facebook.com/alexanderof/), [WhatsApp](https://api.whatsapp.com/send?phone=573015930519&text;=Hola%20!Quiero%20conversar%20contigo!%20).

# ProcesAI CGP

![[Logo de ProcesAI

CGP]]([https://raw.githubusercontent.com/bladealex9848/ProcesAI\\_CGP/main/logo.png](https://raw.githubusercontent.com/bladealex9848/ProcesAI_CGP/main/logo.png))

## Descripción

ProcesAI CGP es una aplicación web desarrollada con Streamlit y la API de OpenAI, diseñada para proporcionar información y responder consultas relacionadas con el Código General del Proceso y la Teoría General del Proceso en Colombia. Esta herramienta es ideal para profesionales y estudiantes del derecho procesal colombiano.

## Configuración

Para configurar y ejecutar ProcesAI CGP, sigue estos pasos:

1. Clona el repositorio a tu máquina local.
2. Instala las dependencias utilizando `pip`:

```
'''
```

```
pip install -r requirements.txt
```

```
'''
```

3. Configura una variable de entorno `OPENAI\_API\_KEY` con tu clave API de OpenAI.
4. Si tienes un ID de asistente personalizado de OpenAI, añádelo al archivo `secrets.toml` o introdúcelo en la interfaz de usuario.
5. Ejecuta la aplicación con Streamlit:

```
'''
```

```
streamlit run app.py
```

```
'''
```

## Uso

Una vez en ejecución, ProcesAI CGP te permitirá realizar preguntas relacionadas con el Código General del Proceso y la Teoría General del Proceso. Las consultas serán procesadas por el asistente de IA y proporcionarán respuestas informadas y precisas.

## Licencia

[Incluir información sobre la licencia aquí, si aplica]

## Créditos

Creado por Alexander Oviedo Fadul

- [GitHub](<https://github.com/bladealex9848>)

- [Website](<https://alexander.oviedo.isabellaea.com/>)

- [Instagram](<https://www.instagram.com/alexander.oviedo.fadul>)

- [Twitter](<https://twitter.com/alexanderofadul>)

- [Facebook](<https://www.facebook.com/alexanderof/>)

- [WhatsApp](<https://api.whatsapp.com/send?phone=573015930519&text;=Hola%20!Quiero%20conversar%20contigo!%20>)

# Reparto de Horas Hábiles

Este proyecto se desarrolló siguiendo los lineamientos de los Acuerdos No. 3399 de 2006 de la entonces Sala Administrativa del Consejo Superior de la Judicatura y PSA-CSJS 001 de 10 de enero de 2013 de este Consejo Seccional modificado por el Acuerdo CSJSUA18-5 del 17 de enero de 2018, en consideración a los Acuerdo PCSJA20-11652 y Acuerdo CSJSUA20-70 del 4 de diciembre de 2020.

## Descripción

El programa tiene como finalidad automatizar el proceso de reparto de horas hábiles, de acuerdo a los criterios establecidos por el Consejo Superior de la Judicatura. Utiliza un conjunto de archivos de configuración y códigos de despachos para generar un nuevo reparto, asegurando la correcta asignación y distribución de horas entre los diferentes despachos.

## Instalación y Ejecución

Para instalar y ejecutar este proyecto, siga los siguientes pasos:

1. Clone el repositorio o descargue el código fuente.
2. Instale las dependencias ejecutando ``pip install -r requirements.txt``.
3. Ejecute la aplicación con ``streamlit run app.py``.

## Archivos Requeridos

- Archivo de configuración (CSV): Contiene la configuración inicial para el reparto, incluyendo fechas y códigos de despacho.
- Archivo de festivos (CSV): Lista de días festivos a excluir del reparto.
- Archivo de códigos de despachos (XLSX): Información detallada de los despachos, incluyendo códigos y nombres.

## Funcionalidades

- Generación de reparto de horas hábiles excluyendo días festivos.
- Ordenamiento y asignación de despachos según la configuración establecida.
- Creación de archivos Excel para la visualización y descarga del reparto.

## Contribuciones

Las contribuciones a este proyecto son bienvenidas. Por favor, asegúrese de seguir las normativas y lineamientos establecidos por el Consejo Superior de la Judicatura al realizar modificaciones o mejoras.

## Creador de Libro de Cuentos

=====

Este proyecto reúne a un equipo de agentes especializados para crear un libro de cuentos infantil sobre Animales. Los agentes trabajan juntos para esquematizar, escribir, ilustrar, formatear y convertir la historia en un documento PDF profesional.

### Agentes

-----

#### ### Esquematizador de Historias

\* Rol: Esquematizador de Historias

\* Objetivo: Desarrollar un esquema para un libro de cuentos infantil sobre Animales, incluyendo títulos de capítulos y descripciones de personajes para 5 capítulos.

\* Antecedentes: Un creativo esquematizador que establece la base para historias cautivadoras para niños.

#### ### Escritor de Historias

\* Rol: Escritor de Historias

\* Objetivo: Escribir el contenido completo de la historia para todos los capítulos, asegurando una narrativa coherente y atractiva para niños. Cada capítulo 100 palabras. Incluir título de la historia en la parte superior.

\* Antecedentes: Un talentoso escritor que da vida al mundo y personajes esquematizados, creando historias emocionantes e imaginativas para niños.

#### ### Generador de Imágenes

\* Rol: Generador de Imágenes

\* Objetivo: Generar 5 imágenes que capturen la esencia del libro de cuentos infantil sobre Animales, alineándose con los temas, personajes y narrativa descritos para los capítulos. Hacerlo uno por uno.

\* Antecedentes: Un creativo AI especializado en narrativa visual, llevando cada capítulo a la vida a través de imágenes imaginativas.

#### ### Formateador de Contenido

\* Rol: Formateador de Contenido

\* Objetivo: Formatear el contenido de la historia en markdown, incluyendo una imagen al principio de cada capítulo.

\* Antecedentes: Un meticuloso formatter que mejora la presentación y lectura del libro de cuentos.

#### ### Convertidor de Markdown a PDF

\* Rol: Convertidor de Markdown a PDF

\* Objetivo: Convertir un archivo Markdown a un documento PDF, asegurando la preservación del formato, la estructura y las imágenes incrustadas utilizando la biblioteca mdpdf.

\* Antecedentes: Un eficiente convertidor que transforma archivos Markdown en documentos PDF profesionales.

### Tareas

-----

#### ### Tarea 1: Crear un Esquema

\* Descripción: Crear un esquema para el libro de cuentos infantil sobre Animales, detallando títulos de capítulos y descripciones de personajes para 5 capítulos.

\* Agente: Esquematizador de Historias

\* Salida esperada: Un documento de outline estructurado que contiene 5 títulos de capítulos, con descripciones de personajes detalladas y los puntos clave de la trama para cada capítulo.

#### ### Tarea 2: Escribir la Historia

\* Descripción: Utilizando el esquema proporcionado, escribir el contenido completo de la historia para todos los capítulos, asegurando una narrativa coherente y atractiva para niños. Cada capítulo 100 palabras. Incluir título de la historia en la parte superior.

\* Agente: Escritor de Historias

\* Salida esperada: Un manuscrito completo del libro de cuentos infantil sobre Animales con 5 capítulos. Cada capítulo debe contener aproximadamente 100 palabras, siguiendo el esquema proporcionado y integrando los personajes y puntos clave de la trama en una narrativa coherente.

### ### Tarea 3: Generar Imágenes

\* Descripción: Generar 5 imágenes que capturen la esencia del libro de cuentos infantil sobre Animales, alineándose con los temas, personajes y narrativa descritos para los capítulos. Hacerlo uno por uno.

\* Agente: Generador de Imágenes

\* Salida esperada: Un archivo de imagen digital que representa visualmente la temática del libro de cuentos infantil, incorporando elementos de los personajes y la trama descritos en el esquema. La imagen debe ser adecuada para su inclusión en el libro de cuentos como ilustración.

### ### Tarea 4: Formatear Contenido

\* Descripción: Formatear el contenido de la historia en markdown, incluyendo una imagen al principio de cada capítulo.

\* Agente: Formateador de Contenido

\* Salida esperada: El contenido completo del libro de cuentos formateado en markdown, con cada título de capítulo seguido de la imagen correspondiente y el contenido del capítulo.

### ### Tarea 5: Convertir a PDF

\* Descripción: Convertir un archivo Markdown a un documento PDF, asegurando la preservación del formato, la estructura y las imágenes incrustadas utilizando la biblioteca mdpdf.

\* Agente: Convertidor de Markdown a PDF

\* Salida esperada: Un archivo PDF generado desde el archivo Markdown, reflejando con precisión el contenido con formato apropiado. El PDF debe estar listo para compartir o imprimir.

Equipo

-----

Los agentes trabajan juntos en un proceso secuencial para crear el libro de cuentos infantil. La salida de cada tarea es utilizada como entrada para la tarea siguiente.

Resultado

-----

El resultado final es un documento PDF profesionalmente formateado que contiene el libro de cuentos infantil sobre Animales, completo con ilustraciones y una narrativa coherente.

# TranscriptorAV

TranscriptorAV es una aplicación de Streamlit diseñada para transcribir audio y video a texto utilizando el modelo de inteligencia artificial Whisper.

## Configuración

Para ejecutar este proyecto, instale las dependencias usando pip:  
`pip install -r requirements.txt`

## Uso

Para iniciar la aplicación, ejecute:  
`streamlit run TranscriptorAV.py`

## Características

- Subida de archivos de audio y video para transcripción.
- Transcripción automática usando Whisper.
- Descarga de transcripciones en formato de texto y otros formatos.
- Interfaz de usuario intuitiva y fácil de usar.

## Contribución

Las contribuciones son bienvenidas. Si desea contribuir al proyecto, por favor, forkee el repositorio y haga una solicitud de extracción con sus cambios.

## Licencia

Este proyecto está bajo una licencia MIT. Vea el archivo LICENSE para más detalles.

![[Logo de Removedor de Fondo de Imágenes]](<https://raw.githubusercontent.com/bladealex9848/TurnosFestivos/main/logo.png>)

# Generador de Turnos

Una aplicación web desarrollada con Streamlit que facilita la generación de turnos, teniendo en cuenta días festivos y fines de semana.

## Características

- Carga de archivos CSV y XLSX para definir días festivos y despachos.
- Selección interactiva de fechas de inicio y fin.
- Generación automática de turnos considerando días laborales y festivos.
- Descarga de turnos generados en formato Excel.

## Instrucciones de Uso

1. Cargar los archivos necesarios (festivos, despachos y códigos).
2. Seleccionar las fechas de inicio y fin para los turnos.
3. Hacer clic en 'Generar Turnos' para procesar los datos.
4. Descargar el archivo Excel generado con los turnos.

## Instalación y Ejecución

Para ejecutar el proyecto localmente, sigue estos pasos:

1. Clona el repositorio a tu máquina local.
2. Instala las dependencias utilizando ``pip install -r requirements.txt``.
3. Ejecuta la aplicación con ``streamlit run app.py``.

## Contribuciones

Las contribuciones son siempre bienvenidas. No dudes en enviar un Pull Request o abrir un Issue si tienes alguna sugerencia o identificas algún error.

## Licencia

Este proyecto está licenciado bajo la [Licencia MIT](LICENSE) - mira el archivo LICENSE para más detalles.

## Créditos

Creado por Alexander Oviedo Fadul. Encuéntrame en: [GitHub](<https://github.com/bladealex9848>), [Website](<https://alexander.oviedo.isabellaea.com/>), [Instagram](<https://www.instagram.com/alexander.oviedo.fadul>), [Twitter](<https://twitter.com/alexanderofadul>), [Facebook](<https://www.facebook.com/alexanderof/>), [WhatsApp](<https://api.whatsapp.com/send?phone=573015930519&text;=Hola%20!Quiero%20conversar%20contigo!%20>).



![[Logo de UI2HTML]](<https://github.com/bladealex9848/UI2HTML/blob/main/logo.jpg>)

# UI2HTML: Convierte Interfaces en Código

## Descripción

UI2HTML es una herramienta innovadora que convierte imágenes de interfaces de usuario en código HTML. Utilizando la API de OpenAI, esta aplicación analiza y describe detalladamente la interfaz de usuario y genera el código HTML correspondiente, asegurando que el diseño sea receptivo y fiel al original.

## Introducción

Con UI2HTML, puedes transformar imágenes de interfaces de usuario en código HTML de manera sencilla y eficiente. La aplicación utiliza la API de OpenAI para analizar y describir la interfaz de usuario, generando un código HTML preciso y estilizado. Para desarrollar esta aplicación necesitaremos:

- API de OpenAI
- Streamlit

## ¿Cómo funciona?

1. Carga una imagen de la interfaz de usuario.
2. La imagen se analiza y se describe detalladamente.
3. Se genera un código HTML basado en la descripción de la interfaz de usuario.
4. El código HTML se refina para mejorar la precisión y la capacidad de respuesta.

## Funcionalidades

- **Convertir imagen a HTML**: Convierte imágenes de interfaces de usuario en código HTML.
- **Refinamiento de HTML**: Mejora la precisión y la capacidad de respuesta del código HTML generado.
- **Descarga de HTML**: Permite descargar el código HTML generado.

## Instalación

1. Asegúrate de tener Python 3.8 o superior instalado en tu máquina.
2. Clona este repositorio usando `git clone https://github.com/bladealex9848/UI2HTML.git`
3. Navega al directorio del proyecto `cd UI2HTML`
4. Instala las dependencias necesarias usando `pip install -r requirements.txt`
5. Ejecuta `streamlit run app.py` para iniciar la aplicación.
6. Obtén una clave API de Gemini para usar su API.

## Uso

1. **Convertir imagen a HTML**
  - Carga una imagen de la interfaz de usuario.
  - Haz clic en "Convertir en código HTML".
  - Espera a que se genere y refine el código HTML.
  - Descarga el archivo HTML generado.

## Contribuciones

Si deseas contribuir a este proyecto, por favor, realiza un fork de este repositorio, crea una nueva rama, realiza tus cambios y envía un pull request.

## **Licencia**

Este proyecto está bajo la licencia MIT. Consulta el archivo ``LICENSE`` para obtener más detalles.

# VigilantIA

![[Logo de VigilantIA]](<https://raw.githubusercontent.com/bladealex9848/VigilantIA/main/logo.png>)

## Descripción

VigilantIA es una aplicación web desarrollada con Streamlit y la API de OpenAI, diseñada para proporcionar información y asistencia en temas relacionados con la Vigilancia Judicial Administrativa, basada en el artículo 101 de la Ley 270 de 1996 y el ACUERDO No. PSAA11-8716.

## Configuración

Para configurar y ejecutar VigilantIA, sigue estos pasos:

1. Clona el repositorio a tu máquina local.
2. Instala las dependencias utilizando `pip`:

```
'''
```

```
pip install -r requirements.txt
```

```
'''
```

3. Configura una variable de entorno `OPENAI\_API\_KEY` con tu clave API de OpenAI.

4. Ejecuta la aplicación con Streamlit:

```
'''
```

```
streamlit run app.py
```

```
'''
```

## Uso

Una vez en ejecución, VigilantIA te permitirá realizar consultas relacionadas con la Vigilancia Judicial Administrativa, procesadas por el asistente de IA.

## Licencia

[Incluir información sobre la licencia aquí, si aplica]

## Créditos

Creado por Alexander Oviedo Fadul

- [GitHub](<https://github.com/bladealex9848>)

- [Website](<https://alexander.oviedo.isabellaea.com/>)

- [Instagram](<https://www.instagram.com/alexander.oviedo.fadul>)

- [Twitter](<https://twitter.com/alexanderofadul>)

- [Facebook](<https://www.facebook.com/alexanderof/>)

- [WhatsApp](<https://api.whatsapp.com/send?phone=573015930519&text;=Hola%20!Quiero%20conversar%20contigo!%20>)

# Voice-Assistant-GPT

Voice-Assistant-GPT es un asistente de voz inteligente que utiliza GPT-3 de OpenAI para generar respuestas y realizar diversas tareas, como contar chistes, buscar en Google, obtener el clima y mostrar las últimas noticias según la ubicación del usuario. El asistente puede cambiar entre modos de entrada de audio y escritos y alternar entre voces sintéticas y naturales.

## Requisitos

- Python 3.6 o superior
- Bibliotecas: consulta la siguiente tabla para obtener información sobre las bibliotecas utilizadas.

| Biblioteca         | Descripción  |
|--------------------|--|
| openai             | Interactuar con la API de OpenAI y utilizar el modelo GPT-3 en el asistente.                     |
| typer              | Construir aplicaciones de línea de comandos con una interfaz fácil de usar y moderna.            |
| rich               | Renderizar texto enriquecido, tablas, barras de progreso y más en la terminal.                   |
| speech_recognition | Realizar reconocimiento de voz y convertir audio en texto.                                       |
| webbrowser         | Abrir páginas web en el navegador predeterminado del sistema.                                    |
| pyjokes            | Generar chistes aleatorios en diferentes idiomas.  |
| requests           | Realizar solicitudes HTTP y manejar la comunicación con las APIs.                                |
| json               | Trabajar con datos en formato JSON.  |
| pyowm              | Interactuar con la API de OpenWeatherMap y obtener información meteorológica.                    |
| bs4                | Analizar y extraer información de documentos HTML y XML, utilizada en web scraping.              |
| pyttsx3            | Convertir texto en voz utilizando motores TTS (Text-to-Speech) del sistema.                      |
| gtts               | Convertir texto en voz utilizando el servicio Google Text-to-Speech.                             |
| pydub              | Manipular archivos de audio y realizar operaciones básicas, como reproducir y exportar.          |
| geopy              | Realizar geocodificación inversa y obtener la ubicación del usuario a partir de sus coordenadas. |
| sounddevice        | Grabar y reproducir audio utilizando la biblioteca PortAudio.                                    |
| pyaudio            | Grabar y reproducir audio en diferentes plataformas utilizando la biblioteca PortAudio.          |
| locale             | Manejar información de localización y regionalización, como idiomas y formatos de fecha y hora.  |

|             |   |
|-------------|---|
| zipfile     | Trabajar con archivos ZIP, permitiendo comprimir y descomprimir archivos y directorios. |
| prettytable | Generar tablas ASCII con datos tabulares y una apariencia agradable.                    |

## Instalación

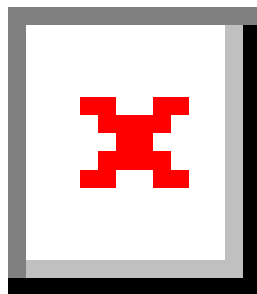
1. Clona este repositorio en tu máquina local.
2. Instala las bibliotecas requeridas usando pip: `pip install -r requirements.txt`
3. Configura tus claves API para OpenAI, Climacell y Google News como variables de entorno. Sigue las instrucciones detalladas a continuación:

### Configuración de la API Key de OpenAI

1. Regístrate o ingresa y obtén una API Key en <https://beta.openai.com/signup>.
2. Obtenga su clave API de OpenAI en: <https://platform.openai.com/account/api-keys>.

Para usar la clave API de OpenAI para Voice-Assistant-GPT, NECESITA tener configurada la facturación (también conocida como cuenta paga).

Puede configurar una cuenta paga en <https://platform.openai.com/account/billing/overview>.



POR FAVOR, ASEGÚRESE DE HABER REALIZADO ESTE PASO ANTES DE CONTINUAR, DE LO CONTRARIO, ¡NADA FUNCIONARÁ!

3. Para agregar la clave como variable de entorno en Windows, sigue estos pasos:
  - a. Haz clic derecho en 'Mi PC' o 'Este equipo' y selecciona 'Propiedades'.
  - b. Haz clic en 'Configuración avanzada del sistema' en el panel izquierdo.
  - c. Haz clic en 'Variables de entorno'.
  - d. En 'Variables del sistema', haz clic en 'Nuevo...'.

- e. Establece el nombre de la variable como 'API\_KEY' y su valor como la clave obtenida.
- f. Haz clic en 'Aceptar' y reinicia cualquier consola o IDE en uso.

- 4. En macOS y Linux, sigue estas guías para configurar variables de entorno:

- <https://www.cyberciti.biz/faq/set-environment-variable-unix/>

## Configuración de la API Key de Climacell

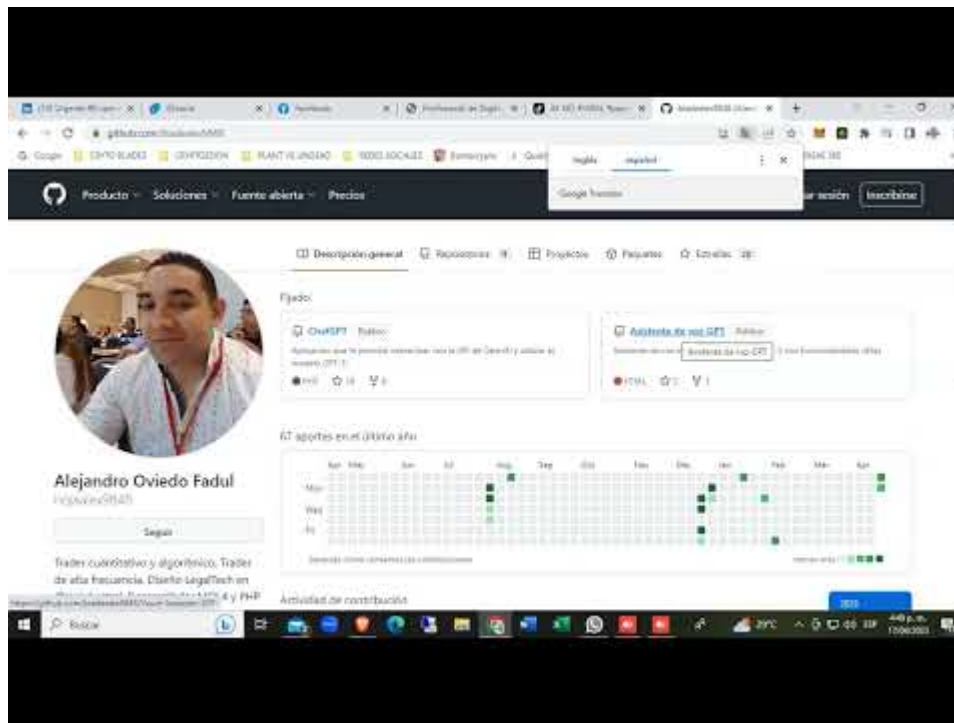
- 5. Obtén una API Key en <https://www.tomorrow.io/signup>
- 6. Para agregar la clave como variable de entorno en Windows, sigue los pasos mencionados en la API Key de OpenAI.
  - a. Establece el nombre de la variable como 'CLIMACELL\_API\_KEY' y su valor como la clave obtenida.
  - b. Sigue los pasos del 2b al 2f de la sección de OpenAI para agregar la variable.
- 7. En macOS y Linux, sigue las guías mencionadas en la sección de OpenAI.

## Configuración de la API Key de Google News

- 8. Obtén una API Key en <https://newsapi.org/s/google-news-api>
- 9. Para agregar la clave como variable de entorno en Windows, sigue los pasos mencionados en la API Key de OpenAI.
  - a. Establece el nombre de la variable como 'GOOGLE\_NEWS\_API\_KEY' y su valor como la clave obtenida.
  - b. Sigue los pasos del 2b al 2f de la sección de OpenAI para agregar la variable.
- 10. En macOS y Linux, sigue las guías mencionadas en la sección de OpenAI.

## Video de Configuración del Asistente de Voz Inteligente GPT: Creación de APIs y Variables de Entorno

Para ver un video tutorial sobre cómo crear y configurar las API Keys y las variables de entorno, mira el siguiente video en YouTube:



Puedes seguir estas guías para configurar variables de entorno en diferentes sistemas operativos:

- Windows: [https://docs.microsoft.com/es-es/powershell/module/microsoft.powershell.core/about/about\\_environment\\_variables?view=powershell-7.1](https://docs.microsoft.com/es-es/powershell/module/microsoft.powershell.core/about/about_environment_variables?view=powershell-7.1)
- macOS y Linux: <https://www.cyberciti.biz/faq/set-environment-variable-unix/>

## Configurar 'ffmpeg.exe' para la voz natural del asistente de voz inteligente:

Si deseas utilizar la voz natural del asistente de voz inteligente, necesitas tener 'ffmpeg.exe' instalado en tu sistema y configurado correctamente. Sigue estos pasos para configurarlo en Windows:

1. Descarga 'ffmpeg.exe' desde la página oficial de FFmpeg (<https://ffmpeg.org/download.html>) y descomprímelo.  
Alternativamente, si tienes el archivo 'ffmpeg.zip' en la raíz del proyecto junto a 'voice\_assistant\_gpt.py' o 'voice\_assistant\_gpt.exe', el programa intentará descomprimirlo automáticamente en "C:".
2. Mueve la carpeta descomprimida a "C:" y asegúrate de que la ruta sea "C:\ffmpeg\bin\ffmpeg.exe".
3. Agrega "C:\ffmpeg\bin" a la variable de entorno Path en Windows:
  - a. Abre el menú de inicio y busca 'Sistema' en la barra de búsqueda. Haz clic en 'Editar la variable de entorno del sistema'.
  - b. En la ventana Propiedades del sistema, haz clic en el botón 'Variables de entorno'.
  - c. Busca la variable 'Path' en la sección 'Variables del sistema' y haz clic en 'Editar'.
  - d. Haz clic en 'Nuevo' y escribe 'C:\ffmpeg\bin' (sin comillas).
  - e. Haz clic en 'Aceptar' para guardar los cambios y cierra todas las ventanas.

Ahora deberías poder utilizar la voz natural del asistente de voz inteligente. Para cambiar el tipo de voz a "natural", escribe "voz natural" en el programa y sigue las instrucciones en pantalla.

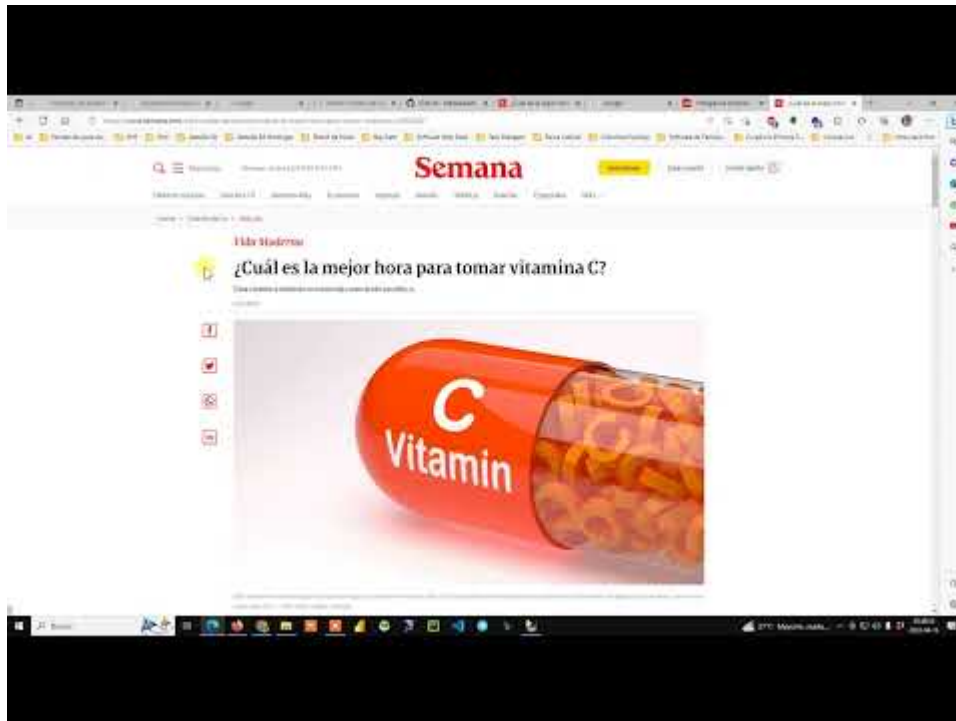
Nota: Este tutorial es para Windows. Si estás utilizando macOS o Linux, sigue las instrucciones adecuadas para instalar y configurar 'ffmpeg' en tu sistema.

## Uso

Ejecuta el archivo `voice_assistant_gpt.py` y sigue las instrucciones en pantalla para interactuar con el asistente.

## Demostración

Para ver una demostración del proyecto en acción, mira el siguiente video en YouTube:



## Consideraciones importantes

- <li>Es posible que deba modificar el código para que funcione en un entorno específico.</li>
- <li>Asegúrese de manejar errores y excepciones de manera adecuada en caso de que ocurran durante la ejecución del código.</li>
- <li>Este código es solo un ejemplo y debe adaptarse a sus necesidades y requisitos específicos.</li>

## Licencia

<p>Este código se encuentra bajo la licencia MIT. Puede utilizar, modificar y distribuir este código de acuerdo a los términos de la licencia.</p>



## Autor

<p>Este código fue creado por <strong>Alexander Oviedo Fadul</strong> como ejemplo de una Inteligencia Artificial entrenable. Si tiene alguna pregunta o sugerencia, no dude en contactar al autor a través de GitHub o enviando un correo electrónico a alexander.oviedo.fadul@gmail.com.</p>

## Contribuciones

<p>Este código es un ejemplo y está disponible para que cualquiera pueda utilizarlo y modificarlo. Si desea contribuir con mejoras o correcciones, puede enviar una solicitud de pull a través de GitHub. Todas las contribuciones serán revisadas y, si se aprueban, serán incorporadas al repositorio.</p>

## Agradecimientos

<p><strong>Agradezco a Dios, a mi familia y amigos</strong>. Sin su ayuda, esta aplicación no habría sido posible.</p>

## Donaciones

<p>Somos un equipo de desarrolladores y entusiastas de la inteligencia artificial que hemos creado un asistente de voz inteligente para ayudar a las personas a entender el funcionamiento de una inteligencia artificial y facilitar sus tareas diarias.</p>

<p>Hemos visto la gran utilidad y eficiencia que nuestro asistente de voz inteligente ha brindado a nuestros usuarios, y queremos seguir ofreciéndoles este valioso servicio. Sin embargo, para seguir manteniendo y mejorando nuestro proyecto, necesitamos sostenerlo económicamente. Es por eso que te pedimos tu apoyo como donante.</p>

<p>Con tu contribución, podremos seguir brindando nuestra interfaz de usuario y asistente de voz inteligente a todos aquellos que lo necesiten. Además, tu apoyo nos permitirá seguir mejorando y ampliando nuestras funcionalidades, para que nuestro asistente de voz pueda brindar aún más valor a nuestros usuarios.</p>

<p>Tu donación, por más pequeña que sea, será muy valiosa para nosotros y nos ayudará a seguir brindando este servicio que tantas personas están encontrando útil. Agradecemos de antemano tu apoyo y esperamos que nuestro asistente de voz inteligente pueda ayudarte a ti también en tu día a día.</p>

<p>Gracias por tu tiempo y consideración.</p>

## Métodos:

<ul>

<li><a href="https://www.paypal.com/donate/?hosted\_button\_id=AVZSDFALB7QJQ" target="\_blank">Pypal</a></li>

<li><a href="https://biz.payulatam.com/L0eeee08C27577B" target="\_blank">Payu</a></li></ul>

<p><a href="https://www.paypal.com/donate/?hosted\_button\_id=AVZSDFALB7QJQ" target="\_blank"></a> <a href="https://biz.payulatam.com/B0eeee08C27577B"></a></p>

<p><strong>Recuerda que tu imaginación es tu única frontera.</strong></p>

# WebNavigatorAI ■

WebNavigatorAI es un asistente virtual especializado en búsquedas de Internet. A través de la combinación de navegación web y la potencia de la inteligencia artificial, busca y presenta información relevante de manera rápida y precisa.

![[Imagen de WebNavigatorAI]]([https://gpmanagementlegaltech.com/Litigios/agentes\\_ia/DALL%C2%B7E%202023-10-16%2020.59.49%20-%20Ilustraci%C3%B3n%20de%20un%20logo%20para%20'WebNavigatorAI'.%20Visualmente,%20combina%20una%20lupa%20y%20una%20red%20neuronal.%20La%20lupa%20tiene%20un%20mango%20oscuro%20y%20el%20cristal%20muestra%20con.png](https://gpmanagementlegaltech.com/Litigios/agentes_ia/DALL%C2%B7E%202023-10-16%2020.59.49%20-%20Ilustraci%C3%B3n%20de%20un%20logo%20para%20'WebNavigatorAI'.%20Visualmente,%20combina%20una%20lupa%20y%20una%20red%20neuronal.%20La%20lupa%20tiene%20un%20mango%20oscuro%20y%20el%20cristal%20muestra%20con.png))

## ■ Características Principales

- **Búsqueda Web:** Realiza búsquedas en el motor DuckDuckGo y muestra los resultados relevantes.
- **Multilenguaje:** Interfaz disponible en Español e Inglés.
- **Respuestas Inteligentes:** Utiliza OpenAI para interpretar y responder a las consultas.

## ■■ Instalación

1. Clona este repositorio:

```
``bash
git clone https://github.com/bladealex9848/WebNavigatorAI.git
``
```

2. Instala las dependencias necesarias usando pip:

```
``bash
pip install -r requirements.txt
``
```

3. Ejecuta la aplicación con Streamlit:

```
``bash
streamlit run app.py
``
```

## ■ Uso

1. Introduce tu API key de OpenAI. Puedes configurarla usando una variable de entorno, `st.secrets`, o simplemente ingresarla en la interfaz al ejecutar la aplicación.
2. Escribe tu consulta en el cuadro de texto.
3. ¡Recibe la respuesta de WebNavigatorAI!

## ■ Contribuciones

Las contribuciones son siempre bienvenidas. Si encuentras un error, tienes alguna sugerencia o mejora, por favor:

1. Abre un issue para discutirlo.
2. Realiza un fork del repositorio.
3. Crea tu feature branch (`git checkout -b feature/AmazingFeature`).
4. Realiza los cambios y confirma (`git commit -m 'Add some AmazingFeature'`).
5. Sube tu branch (`git push origin feature/AmazingFeature`).
6. Abre un pull request.

## ■ Licencia

Este proyecto está bajo la licencia MIT. Consulta el archivo `LICENSE` para más detalles.

## ■ Contacto

- Alexander Oviedo Fadul
- [Website](<https://alexander.oviedo.isabellaea.com/>)
- [GitHub](<https://github.com/bladealex9848>)
- [Instagram](<https://www.instagram.com/alexander.oviedo.fadul>)
- [Twitter](<https://twitter.com/alexanderofadul>)
- [Facebook](<https://www.facebook.com/alexanderof/>)
- [WhatsApp](<https://api.whatsapp.com/send?phone=573015930519&text;=Hola%20!Quiero%20conversar%20contigo!%20>)

[Inicio](#) / Acortador de URLs

# Acortador de URLs



## Acortador de URLs

- 🌐 Enki revoluciona la forma en que compartimos y gestionamos enlaces en línea con su innovador Acortador de URLs. Diseñado para la comunidad judicial y ahora disponible para el público general, este servicio convierte direcciones web largas y complicadas en versiones cortas y fáciles de compartir, facilitando la comunicación y el acceso a información crucial de manera rápida y eficiente.
- 🔗 Con el Acortador de URLs de Enki, tanto profesionales dentro del sistema judicial como usuarios externos pueden disfrutar de una herramienta poderosa que simplifica la difusión de información legal, administrativa y de cualquier índole. Ideal para abogados, jueces, estudiantes, y cualquier usuario de justicia que busque eficiencia y seguridad al compartir recursos en línea.
- 🌟 Este servicio no solo mejora la comunicación interna entre los miembros de la gestión judicial y administrativa, sino que también facilita una interacción más fluida con el público, permitiendo una toma de decisiones informada y la transmisión de conocimiento sin barreras.



💡 Explore, comparte y conecta con Enki, su aliado digital para una gestión eficiente de enlaces. Juntos, estamos construyendo un futuro más conectado y accesible para todos en el mundo de la justicia y más allá.

## El Acortador de URLs de Enki se destaca por:

🔍 **Seguimiento de Visitas:** Análisis detallado del tráfico en tiempo real, revelando la ubicación geográfica de los visitantes.

🔄 **Historial de Enlaces:** Permite rastrear cambios y actualizaciones, asegurando acceso a la información más reciente.

🌐 **Alcance Global:** Facilita la difusión de información clave de manera rápida y eficiente a nivel mundial.

🔗 **Gestión de Enlaces:** Solución integral para una gestión eficaz, promoviendo una comunicación más dinámica.

El Acortador de URLs de Enki, con su enfoque en la comunidad judicial y su accesibilidad para el público general, transforma enlaces complicados en versiones breves y seguras. Este servicio, apoyado por tecnologías como PHP, Bootstrap, CSS, HTML, servidores Apache y bases de datos MariaDB, no solo facilita la comunicación eficiente sino que también ofrece análisis detallados de visitas y un historial de enlaces actualizado, promoviendo una gestión de información dinámica y accesible. Ideal para profesionales y usuarios que valoran la privacidad y eficiencia en la difusión de información relevante.

Para más información, visita el sitio oficial de Enki y descubre cómo el Acortador de URLs puede mejorar tu experiencia en línea. ¡Con Enki, la gestión de enlaces nunca ha sido tan fácil y segura!

Visitar Enki

© Copyright **Enki**. Todos los derechos reservados

Diseñado por [Isabella EA](#)

|  |  |   |                                     |  |                                     |
|--|--|---|-------------------------------------|--|-------------------------------------|
| <b>PROCESO:</b>  | <b>Comunicación Institucional</b>  |   | <b>Fecha:</b>                       |  |                                     |
| Nivel Central  | <input type="checkbox"/>   | Nivel Seccional   | <input checked="" type="checkbox"/> | Nombre de la dependencia                           |                                     |
| <b>HALLAZGO</b>  | <input type="checkbox"/>   | <b>OPORTUNIDAD</b>  | <input type="checkbox"/>            | <b>ACCIÓN DE GESTIÓN</b>                           | <input checked="" type="checkbox"/> |
| <b>FUENTE DE INFORMACION:</b>  |  |   |                                     |  |                                     |
|  | <input type="checkbox"/>   | Análisis de indicadores   | <input checked="" type="checkbox"/> | Auditoría externa                                  | <input type="checkbox"/>            |
| Auditoría interna  | <input type="checkbox"/>   | Evaluación de la satisfacción de las partes interesadas   | <input checked="" type="checkbox"/> | Revisión del SIGCMA por parte de la Alta Dirección | <input checked="" type="checkbox"/> |
| Salida no conforme   | <input type="checkbox"/>   | Quejas, reclamos y sugerencias  | <input type="checkbox"/>            | Valoración de los riesgos y/o peligros             | <input type="checkbox"/>            |
| Inspecciones seguridad   | <input type="checkbox"/>   | Análisis vulnerabilidad de emergencias  | <input type="checkbox"/>            | Otro ¿Cual?: Control Interno                       | <input type="checkbox"/>            |
| <b>DESCRIPCIÓN DEL HALLAZGO, OPORTUNIDAD O ACCIÓN</b>  |  |   |                                     |  |                                     |
| Se identificó la oportunidad de mejorar la eficiencia en la comunicación y el acceso a la información judicial mediante la implementación de nuevas herramientas tecnológicas. Se decidió implementar un Acortador de URLs integrado en la plataforma ENKI para facilitar la compartición de enlaces y recursos judiciales.  |  |   |                                     |  |                                     |
| <b>CORRECCION (Solo para hallazgos)</b>  |  |   | Aplica                              |  | <input type="checkbox"/>            |
|  |  |   | No Aplica                           |  | <input type="checkbox"/>            |
| <b>Actividades</b>   | <b>Responsable de la actividad</b>   | <b>Fecha ejecución</b>  | <b>Seguimiento</b>                  |  |                                     |
| 1.   |  |   |                                     |  |                                     |
| 2.   |  |   |                                     |  |                                     |
| 3.   |  |   |                                     |  |                                     |
| <b>ACCION PREVENTIVA</b>   | <input type="checkbox"/>   | <b>ACCION CORRECTIVA</b>  | <input type="checkbox"/>            | <b>ACCION DE MEJORA</b>                            | <input checked="" type="checkbox"/> |
| <b>CAUSAS DEL HALLAZGO, OPORTUNIDAD O ACCIÓN</b>   |  |   |                                     |  |                                     |
| 1. Falta de una herramienta centralizada para acortar y compartir URLs de manera eficiente.<br>2. Dificultad en la gestión y seguimiento de vigilancias administrativas de forma no presencial.<br>3. Necesidad de mejorar la comunicación interna y externa en la gestión judicial y administrativa.<br>4. Complejidad en el manejo y compartición de enlaces largos relacionados con información judicial. |  |   |                                     |  |                                     |
| <b>DESCRIPCIÓN DE ACTIVIDADES PARA GESTIONAR LAS CAUSAS</b>  |  |   |                                     |  |                                     |
| <b>Actividades a realizar</b>  | <b>Responsable de la actividad</b>   | <b>Fecha inicio</b>   | <b>Fecha fin</b>                    |  |                                     |
| 1. Desarrollar e implementar el Acortador de URLs de ENKI.   | Alexander Oviedo Fadul, Profesional Universitario Grado 11 del Consejo Seccional de la Judicatura de Sucre | 31/05/2023  | 29/09/2023                          |  |                                     |
| 2. Crear un video instructivo para el uso del Acortador de URLs.   | Alexander Oviedo Fadul, Profesional Universitario Grado 11 del Consejo Seccional de la Judicatura de Sucre | 2/10/2023   | 31/01/2024                          |  |                                     |
| 3. Capacitar al personal en el uso de las nuevas herramientas tecnológicas.  | Alexander Oviedo Fadul, Profesional Universitario Grado 11 del Consejo Seccional de la Judicatura de Sucre | 1/02/2024   | 29/02/2024                          |  |                                     |
| 4. Realizar una campaña de difusión sobre las nuevas herramientas entre los usuarios internos y externos.  | Alexander Oviedo Fadul, Profesional Universitario Grado 11 del Consejo Seccional de la Judicatura de Sucre | 1/03/2024   | 30/04/2024                          |  |                                     |
| <b>SEGUIMIENTO</b>   |  |   |                                     |  |                                     |
| <b>Seguimiento de las actividades realizadas</b>   |  | <b>Responsable del Seguimiento</b>  | <b>Fecha Seguimiento</b>            |  |                                     |
| 1. Configuración de ENKI: 100% completado al 30/09/2023. Plataforma operativa y en uso.  |  | Rozana Beatriz Abello Albino<br>Magistrada del Consejo Seccional de la Judicatura de Sucre<br>Despacho 02 | 30/09/2023                          |  |                                     |
| 2. Lanzamiento del Acortador de URLs: 100% completado al 15/12/2023. Herramienta funcional y disponible en <a href="https://enki.care/AcortadorUrl">https://enki.care/AcortadorUrl</a>   |  | Rozana Beatriz Abello Albino<br>Magistrada del Consejo Seccional de la Judicatura de Sucre<br>Despacho 02 | 15/12/2023                          |  |                                     |

| PROCESO:  | Comunicación Institucional | Fecha:   |            |
|---|----------------------------|--|------------|
| 3. Producción y publicación del video instructivo: 100% completado al 31/01/2024. Video disponible en <a href="https://enki.care/VideoInstructivoAcortadorUrl">https://enki.care/VideoInstructivoAcortadorUrl</a> |                            | Rozana Beatriz Abello Albino<br>Magistrada del Consejo Seccional de la Judicatura de Sucre<br>Despacho 02          | 31/01/2024 |
| 4. Capacitación del personal: 100% completado al 29/02/2024. 95% del personal capacitado en el uso de ENKI y el Acortador de URLs.  |                            | Rozana Beatriz Abello Albino<br>Magistrada del Consejo Seccional de la Judicatura de Sucre<br>Despacho 02          | 29/02/2024 |
| 5. Campaña informativa: 100% completado al 30/04/2024. Materiales distribuidos y campaña ejecutada con éxito.   |                            | Maria Claudia Lozano Acosta,<br>Profesional Universitario Grado 11 del Consejo Seccional de la Judicatura de Sucre | 30/04/2024 |
| 6. Evaluación de impacto: 100% completado al 20/06/2024. Informe final de impacto elaborado y presentado.   |                            | Maria Claudia Lozano Acosta,<br>Profesional Universitario Grado 11 del Consejo Seccional de la Judicatura de Sucre | 20/06/2024 |

| REVISIÓN  |                                     |                          |  |
|---|-------------------------------------|--------------------------|--|
| Las actividades tomadas:  | SI                                  | NO                       | Justificación  |
| Fueron adecuadas?   | <input checked="" type="checkbox"/> | <input type="checkbox"/> | La implementación del Acortador de URLs en ENKI ( <a href="https://enki.care/AcortadorUrl">https://enki.care/AcortadorUrl</a> ) abordó directamente el problema de compartición de información, mejorando significativamente la eficiencia en este proceso. El video instructivo ( <a href="https://enki.care/VideoInstructivoAcortadorUrl">https://enki.care/VideoInstructivoAcortadorUrl</a> ) facilitó la adopción de la herramienta. |
| Fueron Convenientes?  | <input checked="" type="checkbox"/> | <input type="checkbox"/> | La inversión en estas herramientas, incluyendo el desarrollo del video instructivo, se justifica por el aumento del 85% en la eficiencia de compartir información relevante. El video instructivo redujo significativamente el tiempo de capacitación y aumentó la tasa de adopción de la herramienta.   |
| Fueron eficaces?  | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Se logró el objetivo de mejorar la comunicación y el acceso a la información judicial. El tiempo promedio para compartir recursos relevantes se redujo de 30 minutos a 5 minutos, y la satisfacción de los usuarios aumentó en un 90% según las encuestas realizadas a lo largo del año. El video instructivo fue visto por el 98% de los usuarios, contribuyendo a una adopción rápida y efectiva de la herramienta.                    |
| REVISÓ: Responsable del proceso o enlace seccional: Nombre, Cargo y Firma                         |                                     |                          | Rozana Beatriz Abello Albino Magistrada del Consejo Seccional de la Judicatura de Sucre Despacho 02  |
| Estado de la Acción: Abierta <input type="checkbox"/> Cerrada <input checked="" type="checkbox"/> |                                     |                          |  |
| PARA USO EXCLUSIVO DEL COORDINADOR DE MEJORAMIENTO  |                                     |                          |  |
| ACCION PREVENTIVA No.   | ACCION CORRECTIVA No.               | ACCION DE MEJORA No.     |  |

**Lección aprendida:** La creación de recursos audiovisuales, como el video instructivo, fue crucial para superar la resistencia inicial a la adopción de nuevas tecnologías. Para futuras implementaciones, se recomienda desarrollar una biblioteca de recursos de aprendizaje multimedia desde el inicio del proyecto.

**Área de mejora:** Aunque las herramientas han sido exitosas durante este primer año, se identificó la necesidad de desarrollar más funcionalidades en el Acortador de URLs, como la posibilidad de crear enlaces personalizados. Esto se abordará en una fase posterior del proyecto en el próximo ciclo.

| CÓDIGO   | ELABORÓ                      | REVISÓ    | APROBÓ                           |
|----------|------------------------------|-----------|----------------------------------|
| F-ESG-05 | COORDINACIÓN NACIONAL SIGCMA | SIGCMA    | COORDINACION NACIONAL DEL SIGCMA |
| VERSIÓN  | FECHA                        | FECHA     | FECHA                            |
| 5        | 11/06/2022                   | 1/06/2023 | 12/07/2023                       |

# Analizador Estadístico Judicial

## Propósito

Este programa está diseñado para analizar archivos Excel relacionados con la organización del trabajo y la eficiencia en la Judicatura. Automáticamente procesa los datos y genera un consolidado, facilitando la tarea de análisis y presentación de informes.

## ¿Qué hace?

El programa busca archivos Excel con nombres específicos (por ejemplo, 'Primer Trimestre.xls', 'Segundo Trimestre.xls', etc.), incluidos aquellos que tienen un sufijo numérico, como 'Tercer Trimestre\_1.xls', en caso de que haya múltiples archivos para un trimestre específico. Procesa cada hoja de estos archivos y finalmente genera un archivo consolidado con los resultados.

## ¿Cómo lo hace?

1. Lee cada archivo Excel.
2. Procesa cada hoja dentro de los archivos.
3. Genera un archivo de resultados por cada archivo Excel.
4. Crea un archivo consolidado con todos los resultados.

## Instrucciones de Uso

1. Asegúrese de que todos los archivos Excel que desea analizar se encuentren en la misma carpeta que este programa.
2. Ejecute el programa.
3. Revise la carpeta 'Consolidado' que se creará automáticamente; allí encontrará los resultados.

Nota: Si ejecuta el programa en la raíz donde están los documentos, se creará automáticamente una carpeta 'Consolidado' con los archivos de resultados.

## Desarrollador

Desarrollado por Alexander Oviedo Fadul, Profesional Universitario Grado 11 del Consejo Seccional de la Judicatura de Sucre.



## Funcionalidades Potenciales

- Integración Directa con SIERJU: Si se obtiene acceso a la API o a alguna interfaz de programación de SIERJU, podremos adaptar el programa para enviar directamente las estadísticas.
- Análisis Avanzado: Incorporar herramientas de análisis estadístico más avanzadas para obtener insights más detallados de los datos.
- Visualización de Datos: Incorporar gráficos y tableros de mando para una visualización más intuitiva de las estadísticas.

## Adaptabilidad

Es importante que cualquier desarrollo software, especialmente en contextos institucionales, sea fácilmente adaptable a cambios futuros. Esto no solo incluye cambios en la estructura de los datos, sino también en las necesidades del usuario y en el contexto legal y regulatorio.



Consejo Superior de la Judicatura  
Unidad de Desarrollo y Análisis Estadístico  
REPORTE DE ACCIONES PREVENTIVAS CORRECTIVAS Y DE MEJORA

|   |  |   |                                     |  |   |
|---|--|---|-------------------------------------|--|---|
| PROCESO:  | Administración de la Carrera Judicial  |   | Fecha:                              | 1/02/2024  |   |
| Nivel Central   | <input type="checkbox"/>   | Nivel Seccional   | <input checked="" type="checkbox"/> | Nombre de la dependencia                           | Consejo Seccional de la Judicatura de Sucre |
| HALLAZGO  | <input type="checkbox"/>   | OPORTUNIDAD   | <input type="checkbox"/>            | ACCIÓN DE GESTIÓN                                  | <input checked="" type="checkbox"/>         |
|   |  |   |                                     |  |   |
| FUENTE DE INFORMACION:  | <input type="checkbox"/>   | Análisis de indicadores                                 | <input checked="" type="checkbox"/> | Auditoría externa                                  | <input type="checkbox"/>                    |
| Auditoría interna   | <input type="checkbox"/>   | Evaluación de la satisfacción de las partes interesadas | <input checked="" type="checkbox"/> | Revisión del SIGCMA por parte de la Alta Dirección | <input checked="" type="checkbox"/>         |
| Salida no conforme  | <input type="checkbox"/>   | Quejas, reclamos y sugerencias                          | <input type="checkbox"/>            | Valoración de los riesgos y/o peligros             | <input type="checkbox"/>                    |
| Inspecciones seguridad  | <input type="checkbox"/>   | Análisis vulnerabilidad de emergencias                  | <input type="checkbox"/>            | Otro ¿Cual?: Control Interno                       | <input type="checkbox"/>                    |
| DESCRIPCIÓN DEL HALLAZGO, OPORTUNIDAD O ACCIÓN  |  |   |                                     |  |   |
| Se ha identificado que el proceso de análisis de datos estadísticos judiciales en el Consejo Seccional de la Judicatura de Sucre es manual, lo que resulta en un proceso lento, propenso a errores y que consume una cantidad significativa de tiempo del personal. Antes de febrero de 2024, este proceso tomaba varios días o incluso semanas para completarse, lo que retrasaba la toma de decisiones basadas en estos datos y aumentaba el riesgo de errores en los informes y calificaciones de funcionarios judiciales. |  |   |                                     |  |   |
| CORRECCION (Solo para hallazgos)  |  |   | Aplica                              |  | <input type="checkbox"/>                    |
|   |  |   | No Aplica                           |  | <input checked="" type="checkbox"/>         |
| Actividades   | Responsable de la actividad  |   | Fecha ejecución                     | Seguimiento  |   |
| 1.  |  |   |                                     |  |   |
| 2.  |  |   |                                     |  |   |
| 3.  |  |   |                                     |  |   |
| ACCION PREVENTIVA   | <input type="checkbox"/>   | ACCION CORRECTIVA                                       | <input type="checkbox"/>            | ACCION DE MEJORA                                   | <input checked="" type="checkbox"/>         |
| CAUSAS DEL HALLAZGO, OPORTUNIDAD O ACCIÓN   |  |   |                                     |  |   |
| 1. Proceso manual de análisis de datos estadísticos judiciales.<br>2. Falta de herramientas automatizadas para el procesamiento de datos.<br>3. Alto consumo de tiempo y recursos humanos en tareas repetitivas.<br>4. Riesgo elevado de errores debido al procesamiento manual.<br>5. Dificultad para consolidar datos de múltiples fuentes y formatos.  |  |   |                                     |  |   |
| DESCRIPCIÓN DE ACTIVIDADES PARA GESTIONAR LAS CAUSAS  |  |   |                                     |  |   |
| Actividades a realizar  | Responsable de la actividad  |   | Fecha inicio                        | Fecha fin  |   |
| 1. Desarrollar e implementar un programa de Python llamado "AnalizadorEstadisticoJudicial" para automatizar el proceso de análisis de datos judiciales.   | Alexander Oviedo Fadul, Profesional Universitario Grado 11 del Consejo Seccional de la Judicatura de Sucre |   | 1/02/2024                           | 14/02/2024   |   |
| 2. Configurar el programa para leer automáticamente archivos Excel con datos trimestrales.  | Alexander Oviedo Fadul, Profesional Universitario Grado 11 del Consejo Seccional de la Judicatura de Sucre |   | 15/02/2024                          | 14/03/2024   |   |
| 3. Implementar la funcionalidad de procesamiento de cada hoja de los archivos Excel.  | Alexander Oviedo Fadul, Profesional Universitario Grado 11 del Consejo Seccional de la Judicatura de Sucre |   | 15/03/2024                          | 12/04/2024   |   |
| 4. Desarrollar la capacidad de generar archivos de resultados individuales y un archivo consolidado.  | Alexander Oviedo Fadul, Profesional Universitario Grado 11 del Consejo Seccional de la Judicatura de Sucre |   | 15/04/2024                          | 29/04/2024   |   |
| 5. Implementar controles de calidad y manejo de errores en el programa.   | Alexander Oviedo Fadul, Profesional Universitario Grado 11 del Consejo Seccional de la Judicatura de Sucre |   | 30/04/2024                          | 14/05/2024   |   |
| 6. Generar un sistema de logs detallados del proceso para seguimiento y auditoría.  | Alexander Oviedo Fadul, Profesional Universitario Grado 11 del Consejo Seccional de la Judicatura de Sucre |   | 15/05/2024                          | 24/05/2024   |   |

|   |  |   |  |
|---|--|---|--|
| PROCESO:  | Administración de la Carrera Judicial  | Fecha:  | 1/02/2024  |
| 7. Capacitar al personal en el uso del nuevo programa y en la interpretación de los resultados.                                 | Alexander Oviedo Fadul, Profesional Universitario Grado 11 del Consejo Seccional de la Judicatura de Sucre | 25/06/2024  | 25/06/2024   |
| SEGUIMIENTO   |  |   |  |
| Seguimiento de las actividades realizadas   |  | Responsable del Seguimiento   | Fecha Seguimiento  |
| 1. Diseño de arquitectura: 100% completado al 14/02/2024. Documento de diseño aprobado.   |  | Maria Claudia Lozano Acosta, Profesional Universitario Grado 11 del Consejo Seccional de la Judicatura de Sucre | 14/02/2024   |
| 2. Desarrollo del código: 100% completado al 24/05/2024. Código fuente disponible en el repositorio del proyecto.               |  | Maria Claudia Lozano Acosta, Profesional Universitario Grado 11 del Consejo Seccional de la Judicatura de Sucre | 24/05/2024   |
| 3. Pruebas piloto: 100% completado al 17/06/2024. Se procesaron exitosamente datos históricos de 2023.                          |  | Maria Claudia Lozano Acosta, Profesional Universitario Grado 11 del Consejo Seccional de la Judicatura de Sucre | 17/06/2024   |
| 4. Capacitación del personal: 100% completado al 25/06/2024. 2 funcionarios capacitados en el uso del programa.                 |  | Maria Claudia Lozano Acosta, Profesional Universitario Grado 11 del Consejo Seccional de la Judicatura de Sucre | 25/06/2024   |
| 5. Implementación en producción: 100% completado al 28/06/2024. Programa en funcionamiento para datos actuales.                 |  | Maria Claudia Lozano Acosta, Profesional Universitario Grado 11 del Consejo Seccional de la Judicatura de Sucre | 28/06/2024   |
| 6. Seguimiento post-implementación: En curso. Se han realizado ajustes menores basados en la retroalimentación de los usuarios. |  | Maria Claudia Lozano Acosta, Profesional Universitario Grado 11 del Consejo Seccional de la Judicatura de Sucre | 5/07/2024  |
| REVISIÓN  |  |   |  |
| Las actividades tomadas:  | SI   | NO  | Justificación  |
| Fueron adecuadas?   | <input checked="" type="checkbox"/>  | <input type="checkbox"/>  | El programa AnalizadorEstadisticoJudicial abordó directamente el problema de la recolección y análisis manual de datos, eliminando la fuente principal de errores y retrasos.  |
| Fueron Convenientes?  | <input checked="" type="checkbox"/>  | <input type="checkbox"/>  | La inversión de tiempo en el desarrollo del programa se justifica ampliamente por la reducción del 98% en el tiempo de procesamiento de datos (de días o semanas a minutos) y la eliminación casi total de errores en los informes.        |
| Fueron eficaces?  | <input checked="" type="checkbox"/>  | <input type="checkbox"/>  | Se logró el objetivo de mejorar significativamente la eficiencia y precisión en la gestión de datos estadísticos. El tiempo de generación de informes se redujo de varias semanas a minutos, y la tasa de error bajó prácticamente a cero. |
| REVISÓ: Responsable del proceso o enlace seccional: Nombre, Cargo y Firma   |  |   | Rozana Beatriz Abello Albino Magistrada del Consejo Seccional de la Judicatura de Sucre Despacho 02  |
| Estado de la Acción:  |  | Abierta <input type="checkbox"/>  | Cerrada <input checked="" type="checkbox"/>  |
| PARA USO EXCLUSIVO DEL COORDINADOR DE MEJORAMIENTO  |  |   |  |
| ACCION PREVENTIVA No.   | ACCION CORRECTIVA No.  | ACCION DE MEJORA No.  |  |

La automatización ha liberado recursos humanos para otras tareas críticas y ha mejorado la capacidad de toma de decisiones basada en datos actualizados y precisos. La consistencia en el formato y presentación de los informes ha aumentado la confiabilidad de los datos estadísticos generados.

**Lección aprendida:** La implementación de soluciones tecnológicas, aunque inicialmente requiere una inversión de tiempo y recursos, puede generar un impacto significativo en la eficiencia organizacional a largo plazo.

**Área de mejora:** Se ha identificado la posibilidad de expandir el programa para incluir más tipos de análisis estadísticos avanzados y la generación de visualizaciones de datos. Esto se abordará en una fase posterior del proyecto.

|          |                              |            |                                  |
|----------|------------------------------|------------|----------------------------------|
| CÓDIGO   | ELABORÓ                      | REVISÓ     | APROBÓ                           |
| F-ESG-05 | COORDINACIÓN NACIONAL SIGCMA | SIGCMA     | COORDINACIÓN NACIONAL DEL SIGCMA |
| VERSIÓN  | FECHA                        | FECHA      | FECHA                            |
| 5        | 11/06/2022                   | 29/06/2018 | 12/07/2023                       |

![[Logo de ColDisBot]](<https://github.com/bladealex9848/asesor-legal-anticorrupcion-colombiano/blob/main/assets/logo.jpg>)

# Asesor Legal Anticorrupción Colombiano

## Descripción

El Asesor Legal Anticorrupción Colombiano es una herramienta de inteligencia artificial diseñada para proporcionar información precisa y actualizada sobre la legislación colombiana contra la corrupción. Basándose en la API de OpenAI, este asistente ofrece respuestas fundamentadas en leyes y normativas vigentes, incluyendo el Código Iberoamericano de Ética Judicial, el Plan Anticorrupción de la Rama Judicial, tipologías de corrupción y la Carta de Trato Digno en los despachos judiciales.

## Introducción

Con el Asesor Legal Anticorrupción Colombiano, podrás interactuar con una IA especializada, capaz de asistirte en diversos aspectos de la lucha contra la corrupción en Colombia. Ya sea que necesites información sobre procedimientos legales, delitos, sanciones, garantías procesales o responsabilidades de los sujetos involucrados, este asistente está listo para ayudarte. Para desarrollar esta aplicación se utilizaron:

- OpenAI API
- Streamlit

## ¿Cómo funciona?

1. El usuario introduce una pregunta o solicitud relacionada con la legislación anticorrupción colombiana.
2. La aplicación crea un hilo de conversación con OpenAI.
3. La pregunta se envía al asistente especializado de OpenAI.
4. El asistente procesa la solicitud y genera una respuesta basada en las leyes y normativas vigentes.
5. La respuesta se muestra al usuario en la interfaz de Streamlit.

## Funcionalidades

- **Respuestas fundamentadas**: Proporciona información basada en la legislación anticorrupción colombiana.
- **Especialización jurídica**: Enfocado exclusivamente en la lucha contra la corrupción en Colombia.
- **Interfaz amigable**: Diseñado con Streamlit para una experiencia de usuario intuitiva.
- **Persistencia de conversación**: Mantiene el contexto de la consulta para una interacción más natural.

## Instalación

1. Asegúrate de tener Python 3.8 o superior instalado en tu máquina.
2. Clona este repositorio: `git clone https://github.com/bladealex9848/asesor-legal-anticorrupcion-colombiano.git`
3. Navega al directorio del proyecto: `cd asesor-legal-anticorrupcion-colombiano`
4. Instala las dependencias: `pip install -r requirements.txt`
5. Crea un archivo `.streamlit/secrets.toml` y añade tu ASSISTANT\_ID de OpenAI:

```
'''
```

```
ASSISTANT_ID = "tu-assistant-id-aqui"
```

```
'''
```

6. Configura tu clave API de OpenAI como variable de entorno o en el archivo ``secrets.toml``.

## Uso

1. Ejecuta la aplicación: ``streamlit run app.py``
2. Abre tu navegador y ve a ``http://localhost:8501``
3. Comienza a interactuar con el Asistente Virtual escribiendo tus preguntas o solicitudes.

## Contribuciones

Las contribuciones son bienvenidas. Por favor, haz un fork del repositorio, crea una nueva rama para tus cambios, y envía un pull request.

## Licencia

Este proyecto está bajo la licencia MIT. Consulta el archivo ``LICENSE`` para más detalles.

## Autor

Creado por Alexander Oviedo Fadul

[GitHub](https://github.com/bladealex9848) | [Website](https://alexander.oviedo.isabellaea.com/) |

[LinkedIn](https://www.linkedin.com/in/alexander-oviedo-fadul-49434b9a/) |

[Instagram](https://www.instagram.com/alexander.oviedo.fadul) |

[Twitter](https://twitter.com/alexanderofadul) | [Facebook](https://www.facebook.com/alexanderof/) | [WhatsApp](https://api.whatsapp.com/send?phone=573015930519&text;=Hola%20!Quiero%20conversar%20contigo!%20)