



Programación
Computacional III

Docente: Ing. Willian Montes.

PDFPLUMBER + PYTTX3

Por:

- Edith Claros.
- Emerson Portillo.
- Javier Vargas.



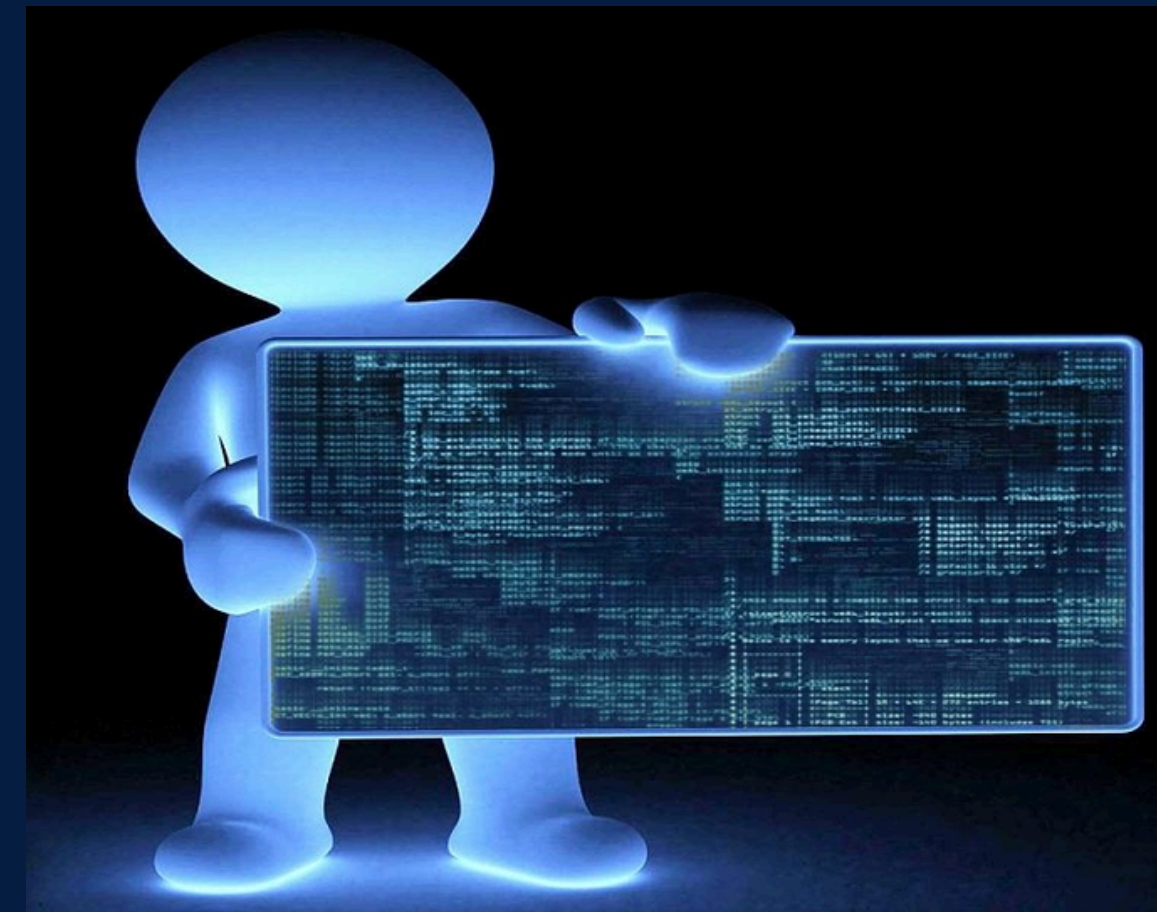
INTRODUCCIÓN

La combinación de librerías como pdfplumber y pyttsx3 ofrece una interesante solución para transformar documentos PDF en audiolibros accesibles. La librería pdfplumber se encarga de la extracción precisa del texto de los archivos PDF, permitiendo a los usuarios acceder a la información contenida en cada página de manera eficiente. Por otro lado, pyttsx3 es una biblioteca de conversión de texto a voz que opera sin conexión, lo que significa que puede generar audio a partir del texto extraído sin necesidad de una conexión a internet.



CONTENIDOS.

1. *Objetivo.*
2. *PdfPlumber.*
3. *Ventajas y desventajas.*
4. *Código base.*
5. *Pytsx3.*
6. *Ventajas y desventajas.*
7. *Código base.*
8. *Código combinado.*
9. *Otras ideas.*
10. *Preguntas.*



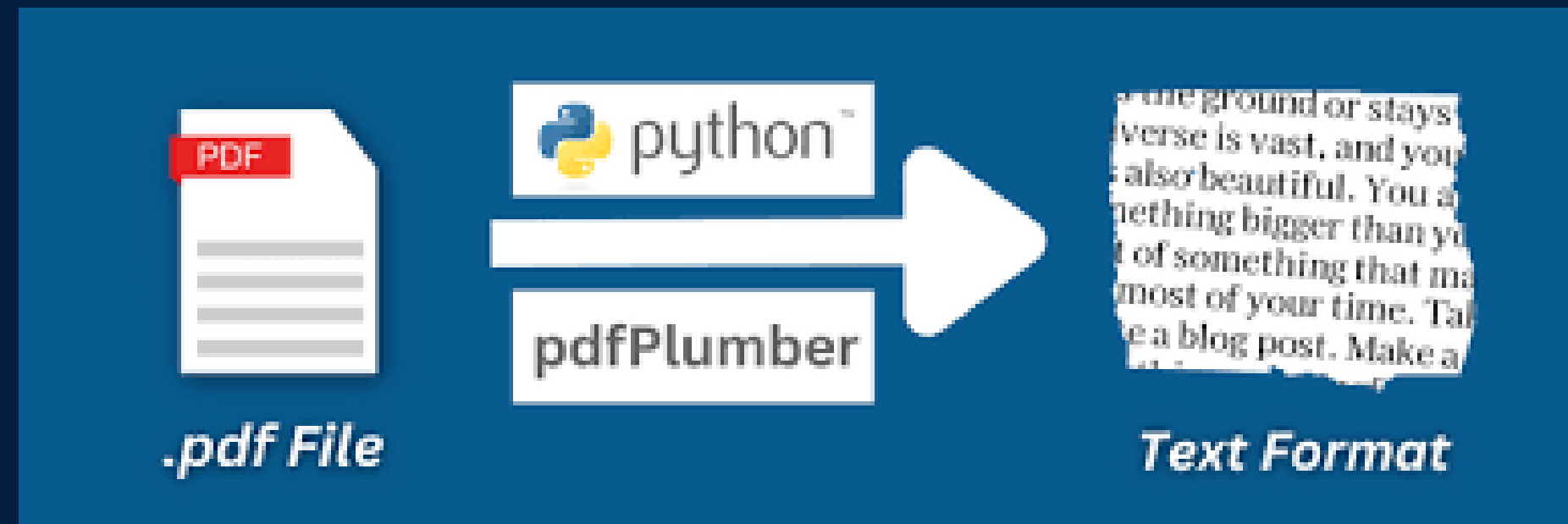
OBJETIVO.

*Conocer diferentes librerías de Python,
que son útiles para facilitar actividades
o resolver problemas en específico.*



Utilidad de pdfplumber:

La librería pdfplumber es una biblioteca de Python que se especializa en la extracción de contenido de archivos PDF, ofreciendo herramientas para acceder a texto, tablas y gráficos. Su utilidad principal radica en la capacidad de convertir documentos PDF en formatos más manejables, como CSV o JSON, facilitando así el análisis de datos. La biblioteca es especialmente útil para tareas de minería de datos, automatización de procesos y análisis de documentos legales o financieros.



VENTAJAS Y DESVENTAJAS DE PDFPLUMBER.

VENTAJAS:

Facilidad de uso: la biblioteca es relativamente fácil de usar y se integra bien con otros paquetes de Python como pandas, lo que permite un análisis más profundo de los datos extraídos.

Documentación completa: la documentación incluye ejemplos claros y una API bien definida, lo que facilita su aprendizaje y uso.

DESVENTAJAS:

Limitaciones con PDFs escaneados: su rendimiento disminuye significativamente con documentos escaneados o aquellos con un formato no estándar, ya que depende de la calidad del texto extraído.

No es muy preciso con las imágenes, ya que cuando se encuentra una, puede omitirla sin dar alguna indicación sobre esta.



SINTAXIS BÁSICA DE PDFPLUMBER.

```
import pdfplumber

with pdfplumber.open("documento.pdf") as pdf:
    texto = ""
    for pagina in pdf.pages:
        texto += pagina.extract_text()
```

Comando de instalación: pip install pdfplumber.

Utilidad de pyttsx3:

La librería pyttsx3 es una biblioteca de Python que permite la conversión de texto a voz de manera sencilla y efectiva. Su principal utilidad radica en la capacidad de transformar texto en audio, lo que la hace ideal para aplicaciones que requieren accesibilidad, como lectores de pantalla y audiolibros. A diferencia de otras bibliotecas similares, pyttsx3 funciona sin conexión a internet, lo que garantiza un rendimiento constante y rápido. Soporta varios motores de síntesis de voz, lo que permite a los usuarios elegir entre diferentes voces y acentos.



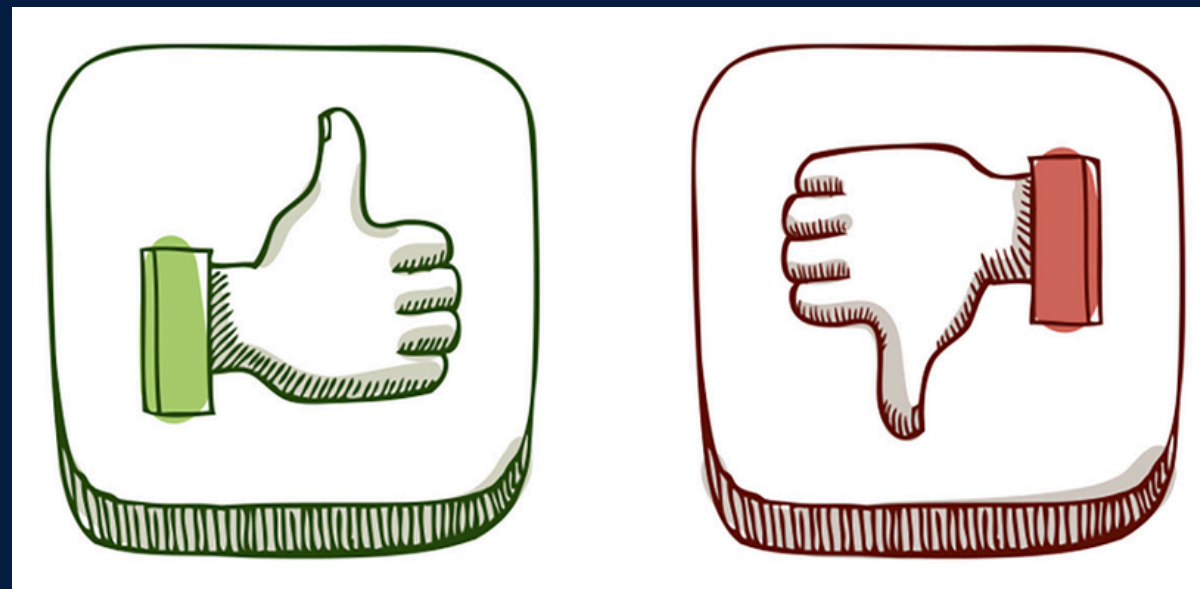
VENTAJAS Y DESVENTAJAS DE PYTTSX3.

Ventajas

- 1 - Offline
- 2- Compatibilidad con múltiples plataformas
- 3- Voz personalizable
- 4- Simplicidad
- 5- Acceso a voces del sistema

Desventajas

- 1- Calidad de voz
- 2- Dependencia del sistema
- 3- Limitaciones en idiomas
- 4- Configuraciones adicionales
- 5- Sin soporte de SSML



SINTAXIS BÁSICA DE PYTTSX3.

```
import pytsx3

engine = pytsx3.init()
engine.setProperty('voice', 'es')

texto = "Esto se convertirá en audio"

engine.save_to_file(texto, 'nombreArchivo.mp3')
engine.runAndWait()
```

Comando de instalación: pip install pytsx3.

Al unir las dos librerías:

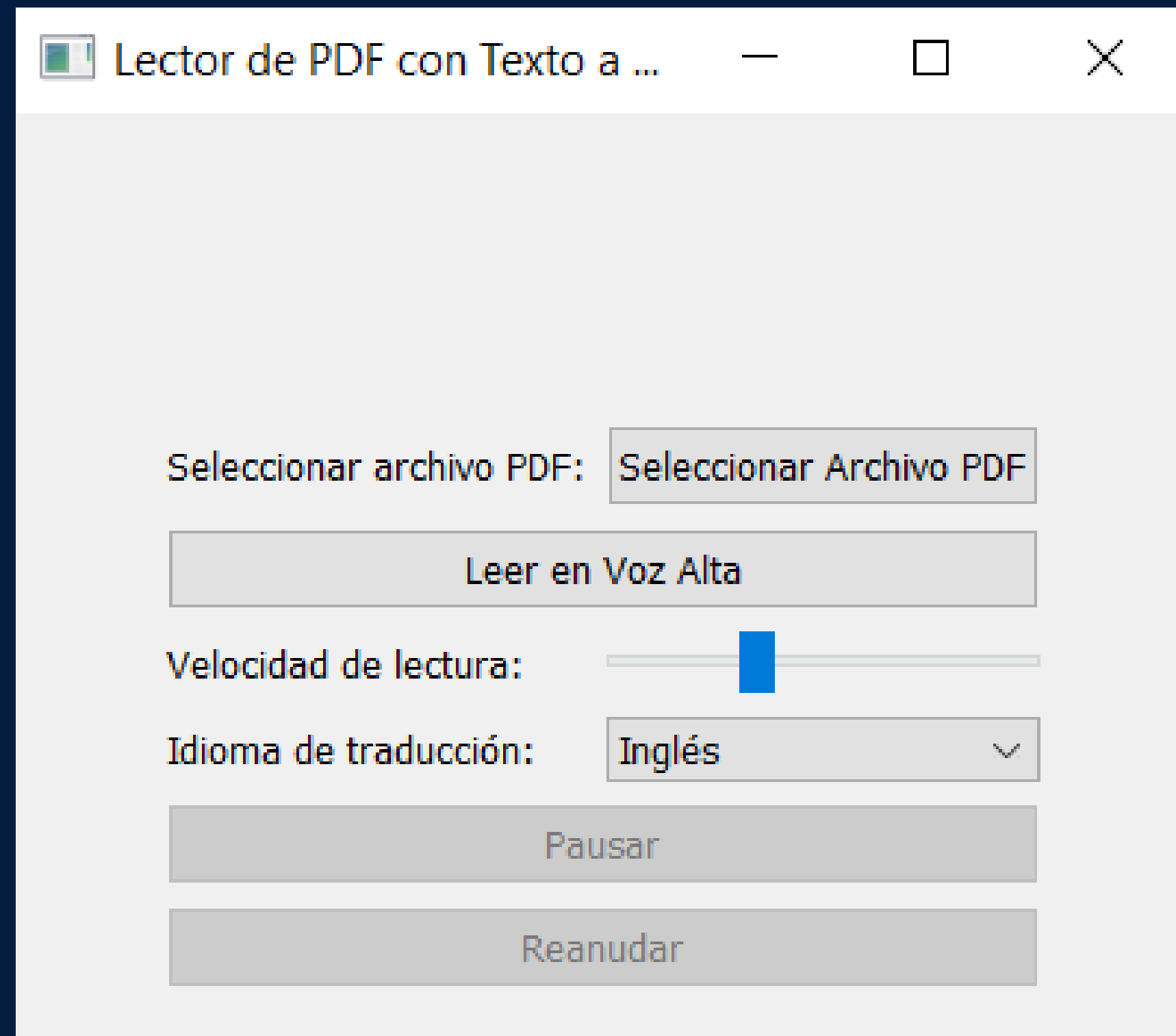
```
import pdfplumber
import pyttsx3

# Función que extrae el texto del pdf.
def ExtraerTexto(docPDF):
    with pdfplumber.open(docPDF) as pdf:
        texto = ""
        for pagina in pdf.pages:
            texto += pagina.extract_text()
    return texto

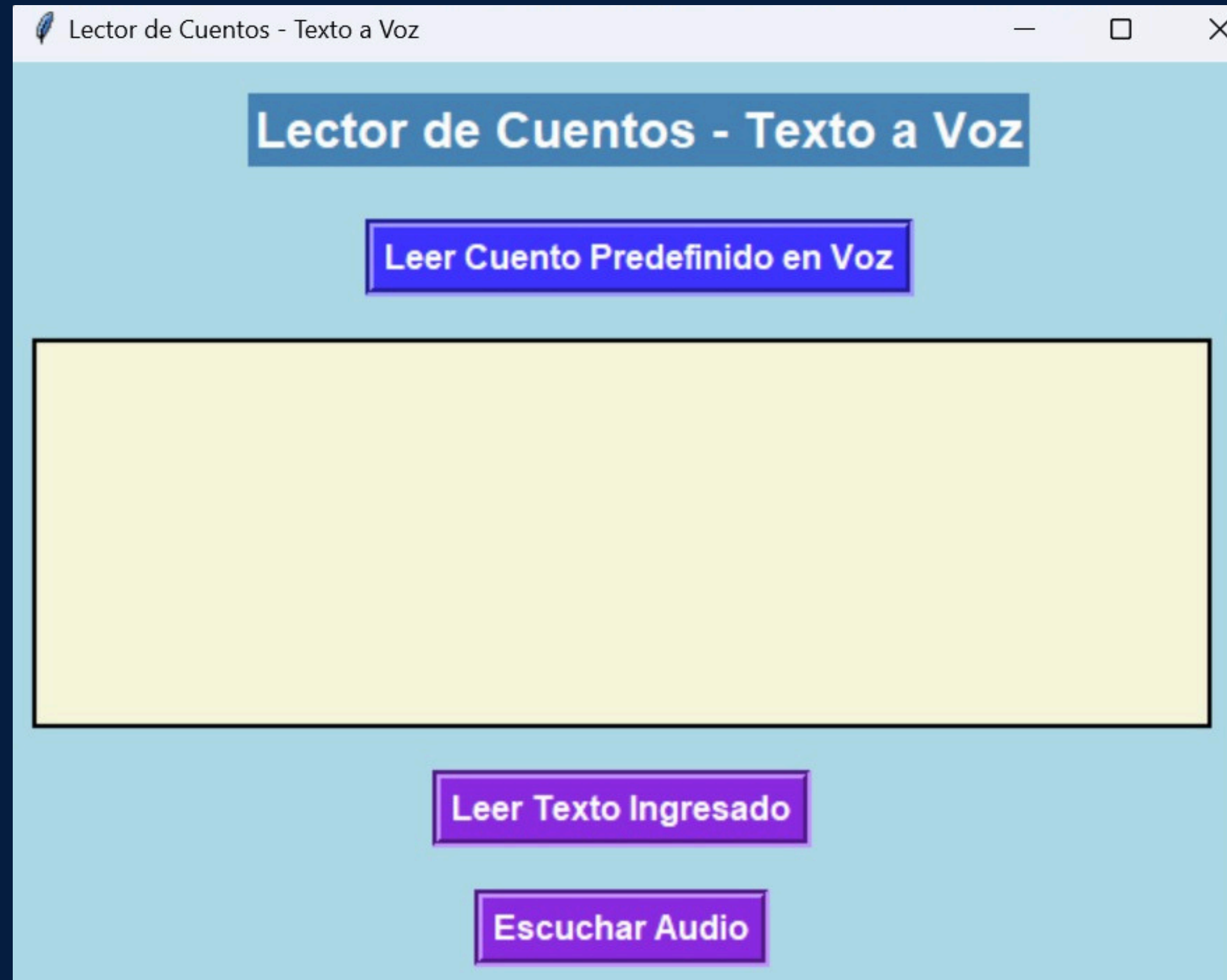
# Función que lee el texto en voz alta.
def LeerTexto(texto):
    engine = pyttsx3.init()
    engine.say(texto)
    engine.runAndWait()

# Pasar lo necesario a las funciones.
texto = ExtraerTexto("pdf_prueba1.pdf")
LeerTexto(texto)
```

OTRAS IDEAS.



OTRAS IDEAS.



¿PREGUNTAS?

