5. **Extractor de texto de PDF:** Utiliza `PyPDF2` para obtener texto de documentos PDF.

¡Claro! Puedo ayudarte con eso.
Aquí tienes una aplicación en **Python** que utiliza la librería PyPDF2 para extraer texto de un documento PDF.

? Extractor de Texto de PDF con PyPDF2

Este script define una función que toma la ruta de un archivo PDF y opcionalmente un rango de páginas, y luego imprime el texto extraído.

Python

```
import PyPDF2
import os

def extraer_texto_de_pdf(ruta_pdf, pagina_inicio=0, pagina_fin=None):
    """
    Extrae texto de un archivo PDF usando PyPDF2.

    Args:
        ruta_pdf (str): La ruta completa al archivo PDF.
        pagina_inicio (int): El índice de la primera página a extraer (0-indexado).
        pagina_fin (int, optional): El índice de la última página a extraer (exclusivo).
                                    Si es None, se extrae hasta la última página.

    Returns:
        str or None: El texto extraído como una cadena, o None si hay un error.
    """
    # 1. Verificar si el archivo existe
    if not os.path.exists(ruta_pdf):
        print(f"❌ Error: El archivo no se encontró en la ruta: {ruta_pdf}")
        return None

    try:
        # 2. Abrir el archivo PDF en modo binario de solo lectura ('rb')
        with open(ruta_pdf, 'rb') as archivo_pdf:
            # 3. Crear un objeto PdfReader
            lector_pdf = PyPDF2.PdfReader(archivo_pdf)

            # 4. Obtener el número total de páginas
            num_paginas = len(lector_pdf.pages)
            print(f"✅ Documento cargado: '{os.path.basename(ruta_pdf)}' tiene {num_paginas} páginas.")

            # 5. Ajustar el rango de páginas si es necesario
            if pagina_fin is None or pagina_fin > num_paginas:
                pagina_fin = num_paginas

            # 6. Validar que el rango sea lógico
            if pagina_inicio >= pagina_fin:
                print("⚠️ Advertencia: El índice de inicio es mayor o igual que el índice final. No se extrajo texto.")
                return ""

            # 7. Iterar sobre las páginas y extraer el texto
```

```
                texto_extraido = []
                for i in range(pagina_inicio, pagina_fin):
                    pagina = lector_pdf.pages[i]
                    print(f"   -> Extrayendo texto de la página {i + 1}...")
                    texto_extraido.append(pagina.extract_text())

                # 8. Unir todo el texto y retornarlo
                return "\n".join(filter(None, texto_extraido))

        except PyPDF2.errors.PdfReadError:
            print("✕ Error: Parece que el archivo no es un PDF válido o está cifrado.")
            return None
        except Exception as e:
            print(f"✕ Ocurrió un error inesperado: {e}")
            return None

# --- Ejemplo de Uso ---
if __name__ == '__main__':
    # **IMPORTANTE:** Reemplaza 'documento.pdf' con la ruta real de tu archivo PDF.
    RUTA_ARCHIVO_PDF = 'documento.pdf'

    print("--- INICIO DEL EXTRACTOR DE TEXTO ---")

    # Si necesitas crear un PDF de prueba, puedes usar una herramienta en línea
    # y guardarlo con el nombre 'documento.pdf' en la misma carpeta que este script.

    # 1. Ejemplo básico: Extraer texto de todo el documento
    texto_completo = extraer_texto_de_pdf(RUTA_ARCHIVO_PDF)

    if texto_completo:
        print("\n=== TEXTO EXTRAÍDO (Completo) ===")
        print(texto_completo[:500] + '...' if len(texto_completo) > 500 else texto_completo) # Muestra los primeros
500 caracteres
        print("==================================\n")
    else:
        print("\nNo se pudo extraer el texto o el archivo no existe.")

    # 2. Ejemplo avanzado: Extraer texto solo de las páginas 2 a 4 (índices 1 a 3)
    # texto_parcial = extraer_texto_de_pdf(RUTA_ARCHIVO_PDF, pagina_inicio=1, pagina_fin=4)

    # if texto_parcial:
    #     print("\n=== TEXTO EXTRAÍDO (Páginas 2 a 4) ===")
    #     print(texto_parcial[:500] + '...' if len(texto_parcial) > 500 else texto_parcial)
    #     print("======================================\n")
```

Instrucciones de Uso

1. Instalar la librería:

   Asegúrate de tener instalado PyPDF2. Puedes instalarlo usando pip:

   Bash
   ```
   pip install PyPDF2
   ```
2. Guardar el código:

   Guarda el código anterior en un archivo llamado, por ejemplo, extractor.py.
3. Preparar el PDF:

Coloca un archivo PDF en la misma carpeta que el script y renómbralo a documento.pdf, o cambia la línea RUTA_ARCHIVO_PDF = 'documento.pdf' en el script por la ruta correcta a tu archivo.

4. Ejecutar el script:

   Ejecuta el script desde tu terminal:

   Bash
   ```
   python extractor.py
   ```
   El script imprimirá el texto extraído del PDF directamente en la consola.

¿Te gustaría que modificara la aplicación para **guardar el texto extraído** en un archivo de texto (.txt) en lugar de imprimirlo en la consola?
¡Excelente! Aquí tienes la versión modificada que **guarda el texto extraído en un archivo .txt**.
He renombrado la función y añadido un argumento para la ruta de salida.

? Extractor de Texto de PDF que Guarda en TXT

Python
```python
import PyPDF2
import os

def extraer_y_guardar_texto_de_pdf(ruta_pdf, ruta_salida_txt, pagina_inicio=0, pagina_fin=None):
    """
    Extrae texto de un archivo PDF y lo guarda en un archivo de texto (.txt).

    Args:
        ruta_pdf (str): La ruta completa al archivo PDF de entrada.
        ruta_salida_txt (str): La ruta donde se guardará el archivo de texto de salida.
        pagina_inicio (int): El índice de la primera página a extraer (0-indexado).
        pagina_fin (int, optional): El índice de la última página a extraer (exclusivo).
                                    Si es None, se extrae hasta la última página.

    Returns:
        bool: True si la extracción y el guardado fueron exitosos, False en caso contrario.
    """
    # 1. Verificar si el archivo PDF existe
    if not os.path.exists(ruta_pdf):
        print(f"❌ Error: El archivo PDF no se encontró en la ruta: {ruta_pdf}")
        return False

    try:
        # 2. Abrir el archivo PDF en modo binario de solo lectura ('rb')
        with open(ruta_pdf, 'rb') as archivo_pdf:
            lector_pdf = PyPDF2.PdfReader(archivo_pdf)
            num_paginas = len(lector_pdf.pages)
            print(f"✅ Documento cargado: '{os.path.basename(ruta_pdf)}' tiene {num_paginas} páginas.")

            # 3. Ajustar y validar el rango de páginas
            if pagina_fin is None or pagina_fin > num_paginas:
                pagina_fin = num_paginas

            if pagina_inicio >= pagina_fin:
                print(" Advertencia: El índice de inicio es mayor o igual que el índice final. No se extrajo texto.")
```

```
                return False

            # 4. Iterar sobre las páginas y extraer el texto
            texto_extraido = []
            for i in range(pagina_inicio, pagina_fin):
                pagina = lector_pdf.pages[i]
                print(f"   -> Extrayendo texto de la página {i + 1}...")
                texto_extraido.append(pagina.extract_text())

            texto_final = "\n".join(filter(None, texto_extraido))

            # 5. Guardar el texto en el archivo de salida
            with open(ruta_salida_txt, 'w', encoding='utf-8') as archivo_txt:
                archivo_txt.write(texto_final)

            print(f"\n✨ ¡Éxito! El texto extraído se ha guardado en: {ruta_salida_txt}")
            print(f"   -> Caracteres guardados: {len(texto_final)}")
            return True

    except PyPDF2.errors.PdfReadError:
        print("❌ Error: Parece que el archivo no es un PDF válido o está cifrado.")
        return False
    except Exception as e:
        print(f"❌ Ocurrió un error inesperado: {e}")
        return False

# --- Ejemplo de Uso ---
if __name__ == '__main__':
    # **IMPORTANTE:** Reemplaza 'documento.pdf' con la ruta real de tu archivo PDF.
    RUTA_PDF_ENTRADA = 'documento_ejemplo.pdf'
    RUTA_TXT_SALIDA = 'texto_extraido.txt'

    print("--- INICIO DEL EXTRACTOR DE TEXTO Y GUARDADO EN TXT ---")

    # Si no tienes un PDF, puedes cambiar 'documento_ejemplo.pdf' por la ruta de uno que tengas.

    # Llamar a la función para extraer y guardar todo el documento
    exito = extraer_y_guardar_texto_de_pdf(RUTA_PDF_ENTRADA, RUTA_TXT_SALIDA)

    if exito:
        print("\nProceso de extracción finalizado con éxito.")
    else:
        print("\nEl proceso de extracción falló.")

    # Puedes intentar un segundo ejemplo, solo de algunas páginas:
    # RUTA_TXT_PARCIAL = 'texto_parcial.txt'
    # print("\n--- Extrayendo páginas específicas (e.g., de la 2 a la 4) ---")
    # extraer_y_guardar_texto_de_pdf(RUTA_PDF_ENTRADA, RUTA_TXT_PARCIAL, pagina_inicio=1,
pagina_fin=4)
```

? Resumen de la Modificación

- **Nuevo Parámetro:** La función ahora requiere ruta_salida_txt para especificar dónde guardar el resultado.

- **Manejo de Archivo de Texto:** Una vez que se extrae todo el texto (texto_final), se utiliza un segundo bloque with open(ruta_salida_txt, 'w', encoding='utf-8') as archivo_txt: para **escribir** el contenido en el archivo de texto.
- **Codificación:** Se usa encoding='utf-8' para asegurar que caracteres especiales, acentos y emojis se guarden correctamente.
  Simplemente guarda este código, asegúrate de que el archivo PDF exista en la ruta especificada, ¡y ejecútalo! El archivo texto_extraido.txt aparecerá en la misma carpeta.
  ¿Hay alguna otra funcionalidad que te gustaría añadir, como un manejo especial para documentos cifrados?