****

**EstructuraCampoValor**

Este flujo te ayuda a transformar y visualizar los datos de un archivo PDF de manera organizada por proximidad, a mi criterio sin logica alguna.

Identifica “Cajas de Texto” e introduce en cada Caja el texto identificado. No se que criterio tiene para determinar el contorno de las cajas de texto.

Luego hace una asociación en que las cajas de texto impares son el “campo” y las pares siguientes el “valor” asociado a ese campo. No tiene ninguna lógica y por eso nos da el diccionario final. Ver [archivoEstructuraCampoValor](https://1drv.ms/t/c/da7fecb8f4a79f34/EQKT_i4uOJ5OkJaS89Tx3OABkDluKNrCrIIc39sdQKVgvQ?e=pfYAID).

He utilizado para la prueba un comunicado de Bankinter y obtener los datos relevantes del mismo. Link al pdf con indicacion de los bloques de texto identificados y el orden: [archivomarcado](https://1drv.ms/b/c/da7fecb8f4a79f34/EWcTnw1MSCRKnWD6vmrFGuwB2lTy7PMgAMuj1g2mqiLNHg?e=OYGpw4)

En esta prueba solo he conseguido leer y obtener el texto, sin conseguir un campo: valor con lógica.

Es interesante destacar que con este codigo no consigo leer pdf que venga con “marca de no lectura.

**Importaciones**

from pdfminer.high\_level import extract\_pages

from pdfminer.layout import LTTextBox

1. **Importación de Librerías**:
   * extract\_pages: Función de PDFMiner que extrae páginas del archivo PDF.
   * LTTextBox: Clase de PDFMiner que representa un bloque de texto dentro de una página del PDF.

**Definición de la Ruta del Archivo PDF**

python

pdf\_path = "data/input/archivo2.pdf" # Cambia esto por la ruta a tu archivo PDF

1. **Ruta del Archivo PDF**:
   * pdf\_path: Variable que almacena la ruta al archivo PDF que se va a procesar. Asegúrate de que la ruta sea correcta.

**Función para Extraer Bloques de Texto**

python

def extraer\_bloques\_pdfminer(pdf\_path):

bloques = []

for page\_layout in extract\_pages(pdf\_path):

for element in page\_layout:

if isinstance(element, LTTextBox):

bloques.append(element.get\_text().strip())

return bloques

1. **Función** extraer\_bloques\_pdfminer:
   * **Entrada**: Ruta del archivo PDF (pdf\_path).
   * **Proceso**:
     + Crea una lista vacía bloques.
     + Recorre cada página del PDF.
     + Dentro de cada página, recorre cada elemento de layout.
     + Si el elemento es un LTTextBox (bloque de texto), extrae el texto del bloque, lo limpia de espacios en blanco innecesarios y lo añade a la lista bloques.
   * **Salida**: Lista de bloques de texto extraídos del PDF.

**Función para Estructurar Datos en Formato "Campo: Valor"**

python

def estructurar\_campos(bloques):

campos\_valores = {}

for i in range(0, len(bloques) - 1, 2): # Avanzar en pares (campo, valor)

campo = bloques[i].strip()

valor = bloques[i + 1].strip() if i + 1 < len(bloques) else ""

if campo and valor: # Ambos deben contener texto

campos\_valores[campo] = valor

return campos\_valores

1. **Función** estructurar\_campos:
   * **Entrada**: Lista de bloques de texto (bloques).
   * **Proceso**:
     + Crea un diccionario vacío campos\_valores.
     + Recorre la lista de bloques en pasos de dos (pares de campo y valor).
     + Limpia de espacios en blanco innecesarios cada par de campo y valor.
     + Si ambos contienen texto, añade el par al diccionario campos\_valores. Se asocian por el orden en el que han sido extraidos los bloques, sin criterio logico.
   * **Salida**: Diccionario con datos estructurados en formato campo: valor.

**Procesar el PDF**

python

bloques = extraer\_bloques\_pdfminer(pdf\_path)

campos\_valores = estructurar\_campos(bloques)

1. **Procesamiento del PDF**:
   * **Extraer bloques**: Llama a la función extraer\_bloques\_pdfminer con la ruta del PDF y almacena los bloques de texto en la variable bloques.
   * **Estructurar campos**: Llama a la función estructurar\_campos con los bloques de texto y almacena el diccionario resultante en la variable campos\_valores.

**Mostrar Resultados**

python

print("Datos estructurados:")

for campo, valor in campos\_valores.items():

print(f"{campo}: {valor}")

1. **Mostrar Resultados**:
   * Imprime una cabecera "Datos estructurados:".
   * Recorre el diccionario campos\_valores y imprime cada par campo: valor.

**Resumen**

En resumen, el programa:

1. **Extrae bloques de texto** de un archivo PDF utilizando PDFMiner.
2. **Estructura los bloques de texto** en pares campo: valor.
3. **Imprime los datos estructurados** en la consola.

**PDFMiner** utiliza varios criterios para "construir" cada bloque de texto, dependiendo de la estructura y el contenido del PDF. Aquí te explico algunos de los aspectos principales:

1. **Layout Objects**: PDFMiner clasifica el contenido del PDF en diferentes objetos de layout, como LTTextBox, LTTextLine, LTRect, etc. Para extraer texto, se suele trabajar principalmente con objetos LTTextBox, que representan bloques de texto agrupados lógicamente.
2. **Páginas y Elementos**: Cada página del PDF se descompone en elementos de layout. La función extract\_pages proporciona un iterador sobre estos objetos de layout en cada página del documento.
3. **Detección de Bloques de Texto**: LTTextBox agrupa líneas de texto (LTTextLine) en bloques coherentes, basándose en la proximidad y alineación de las líneas de texto.
4. **Extracción de Texto**: El método get\_text() de LTTextBox se utiliza para extraer el texto contenido dentro de ese bloque. La llamada a strip() se utiliza para eliminar espacios en blanco innecesarios al principio y al final del texto.

En el código, se detectan los bloques de texto con isinstance(element, LTTextBox) y se extraen con element.get\_text().strip(). Estos bloques se almacenan en una lista bloques, que luego se procesan para estructurar la información en pares de campo y valor.

El proceso de construcción de bloques de texto por parte de PDFMiner es bastante detallado, y puede variar según el diseño y formato del PDF. Si tienes un PDF particular con una estructura compleja, es posible que necesites ajustar la lógica de extracción y procesamiento de texto para obtener mejores resultados.