

Fase 1 — Coleta e Preparação dos Arquivos

1.1. Criar um módulo de upload de PDF (backend ou web)

- Ferramenta: Flask ou FastAPI + React
- O front envia o PDF via API para o backend.

1.2. Criar uma pasta temporária para armazenar os arquivos enviados.

```
python
CopiarEditar
import os
from fastapi import UploadFile

UPLOAD_FOLDER = "temp_pdfs"
os.makedirs(UPLOAD_FOLDER, exist_ok=True)

def save_pdf(uploaded_file: UploadFile):
    filepath = os.path.join(UPLOAD_FOLDER, uploaded_file.filename)
    with open(filepath, "wb") as f:
        f.write(uploaded_file.file.read())
    return filepath
```

Fase 2 — Leitura de PDFs

2.1. Detectar se o PDF é texto ou imagem

- Use [PyMuPDF](#) (fitz) ou [PyPDF2](#) para tentar extrair texto diretamente.
- Se não extrair nada, então é escaneado (imagem) → usar OCR.

2.2. Extração de texto direto (PDF com texto embutido)

```
python
CopiarEditar
import fitz # PyMuPDF

def extract_text_from_pdf(path):
    text = ""
```

```
doc = fitz.open(path)
for page in doc:
    text += page.get_text()
return text
```

2.3. OCR para PDFs com imagem

- Use `pdf2image` para converter cada página em imagem.
- Aplique `Tesseract` OCR nas imagens.

```
python
CopiarEditar
from pdf2image import convert_from_path
import pytesseract

def extract_text_from_image_pdf(path):
    images = convert_from_path(path)
    text = ""
    for image in images:
        text += pytesseract.image_to_string(image, lang='por')
    return text
```

Fase 3 — Extração de Informações com NLP

3.1. Criar extratores baseados em regex + NLP

Exemplo de padrões comuns:

- CNPJ: `\d{2}\.\d{3}\.\d{3}/\d{4}-\d{2}`
- Data: `\d{2}/\d{2}/\d{4}`
- Valor total: `R\$?\d+,\d{2}`

```
python
CopiarEditar
import re
```

```
def extract_fields(text):
    data = {
        "cnpj": re.findall(r'\d{2}\.\d{3}\.\d{3}/\d{4}-\d{2}',
text),
        "valor_total": re.findall(r'R\$ ?\d+,\d{2}', text),
        "data": re.findall(r'\d{2}/\d{2}/\d{4}', text)
    }
    return data
```

3.2. NLP para contratos ou notas não estruturadas

- Use [spaCy](#) para fazer Named Entity Recognition (NER).
 - Pode treinar modelos personalizados depois para encontrar campos como “cliente”, “fornecedor”, “produto”.
-



Fase 4 — Banco de Dados

4.1. Estrutura da Tabela

- Tabela `documentos` com: `id`, `cnpj`, `data`, `valor_total`, `json_extraido`, `caminho_pdf`, `data_upload`.

4.2. Inserção via SQLAlchemy ou ORM de sua escolha

python

CopiarEditar

```
import sqlite3
```

```
def save_to_db(data, caminho_pdf):
    conn = sqlite3.connect("docs.db")
    c = conn.cursor()
    c.execute("""
        CREATE TABLE IF NOT EXISTS documentos (
            id INTEGER PRIMARY KEY AUTOINCREMENT,
            cnpj TEXT,
            data TEXT,
            valor_total TEXT,
            json_extraido TEXT,
```

```
        caminho_pdf TEXT
    )
    """)
    c.execute("INSERT INTO documentos (cnpj, data, valor_total,
json_extraido, caminho_pdf) VALUES (?, ?, ?, ?, ?)",
            (data["cnpj"][0] if data["cnpj"] else None,
             data["data"][0] if data["data"] else None,
             data["valor_total"][0] if data["valor_total"] else
None,
             str(data),
             caminho_pdf))
    conn.commit()
    conn.close()
```

Fase 5 — Interface Web com React

Funcionalidades:

- Upload de arquivos
- Lista de documentos extraídos
- Visualização detalhada dos dados
- Filtros por CNPJ, valor, data

Você pode usar uma stack como:

- Vite + React + Tailwind
 - Chamada para APIs do backend (FastAPI/Flask)
 - Exibição de tabelas com bibliotecas como `react-table` ou `MUI DataGrid`
-

Fase 6 — Treinamento de IA para Notas Diversas (Avançado)





6.1. Armazene exemplos com anotações manuais

- Salve as posições e campos de diferentes formatos de nota (rotulagem).

6.2. Treine um modelo com **LayoutLMv3** ou **Donut** (para layout + OCR)

- Use o dataset **FUNSD**, ou crie seu próprio dataset de NFs.

Extras que Impressionam

-  Exibição visual do PDF com campos destacados
-  Exportação em Excel dos dados extraídos
-  Login e histórico por usuário (Firebase Auth ou JWT)
-  Testes unitários (Pytest)