MANUAL TÉCNICO

Completo para o seu sistema de monitoramento.

Ele foi desenhado para ser custo zero, utilizando a infraestrutura do GitHub para processar dados e hospedar o site.

---

1. Estrutura de Arquivos do Repositório

Para o sistema funcionar, seu repositório no GitHub deve conter exatamente estes arquivos:

Plaintext

```
├── .github/workflows/monitor.yml  # O "motor" que roda sozinho
├── publicacoes/                   # Pasta onde os PDFs serão salvos
├── index.html                     # A cara do seu site
├── monitor.py                     # O robô que busca os dados
├── requirements.txt               # Lista de bibliotecas necessárias
└── README.md                      # Documentação e Licença
```

---

2. O Robô de Monitoramento (monitor.py)

Este script é responsável por acessar o portal, baixar o PDF e decidir em qual pasta de secretaria ele deve entrar.

Python

```python
import os
import requests
import json
import pdfplumber
from datetime import datetime
from bs4 import BeautifulSoup

# Configurações de Conceição do Araguaia
ENTIDADE_ID = "28642"
URL_BASE = "https://www.diariomunicipal.com.br"
URL_PESQUISA = f"{URL_BASE}/famep/pesquisar"

def identificar_secretaria(caminho_pdf):
    """Analisa o texto do PDF para classificar por secretaria."""
    secretarias = {
        "Saude": ["SAUDE", "SUS", "HOSPITAL", "UBS"],
        "Educacao": ["EDUCACAO", "ESCOLA", "ENSINO", "FUNDEB"],
        "Financas": ["FINANCAS", "TESOURARIA", "TRIBUTOS"],
        "Administracao": ["ADMINISTRACAO", "RH", "PESSOAL"],
        "Gabinete": ["GABINETE", "PREFEITO", "DECRETO"]
```

```
    }
    try:
        with pdfplumber.open(caminho_pdf) as pdf:
            texto_completo = " ".join([page.extract_text() for page in pdf.pages]).upper()
            for sec, termos in secretarias.items():
                if any(termo in texto_completo for termo in termos):
                    return sec
    except:
        pass
    return "Geral"

def buscar_e_salvar():
    hoje = datetime.now()
    data_formatada = hoje.strftime("%d/%m/%Y")

    params = {
        "busca_avancada[entidadeUsuaria]": ENTIDADE_ID,
        "busca_avancada[dataInicio]": data_formatada,
        "busca_avancada[dataFim]": data_formatada,
        "busca_avancada[Enviar]": ""
    }

    print(f"Iniciando busca para {data_formatada}...")
    response = requests.get(URL_PESQUISA, params=params)
    soup = BeautifulSoup(response.text, 'html.parser')

    # Busca links de download (ajuste conforme o HTML da FAMEP)
    links = soup.find_all('a', class_='btn-download-pdf')

    if not links:
        print("Nenhuma publicação nova hoje.")
        return

    for link in links:
        pdf_url = URL_BASE + link.get('href')
        pdf_nome = pdf_url.split('/')[-1] + ".pdf"

        # Download temporário para análise
        r = requests.get(pdf_url)
        temp_path = "temp.pdf"
        with open(temp_path, 'wb') as f:
            f.write(r.content)

        # Define secretaria e caminho
        sec = identificar_secretaria(temp_path)
        pasta_final                                                    =
f"publicacoes/{hoje.year}/{hoje.strftime('%m')}/{hoje.strftime('%d')}/{sec}"
```

```
        os.makedirs(pasta_final, exist_ok=True)

        # Salva o arquivo final
        os.rename(temp_path, f"{pasta_final}/{pdf_nome}")
        print(f"Salvo: {sec}/{pdf_nome}")

    gerar_indice_json()

def gerar_indice_json():
    arvore = {}
    for root, dirs, files in os.walk("publicacoes"):
        pdfs = [f for f in files if f.endswith('.pdf')]
        if pdfs:
            arvore[root] = pdfs
    with open("dados.json", "w") as f:
        json.dump(arvore, f, indent=4)

if __name__ == "__main__":
    buscar_e_salvar()
```

---

3. O Agendador Automático (.github/workflows/monitor.yml)

Crie as pastas .github e depois workflows para colocar este arquivo. Ele fará o robô rodar todo dia às 09:00.

YAML

```
name: Monitoramento Diario DOM
on:
  schedule:
    - cron: '0 12 * * *' # 12:00 UTC é 09:00 em Brasília
  workflow_dispatch:

jobs:
  job_monitor:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout do Codigo
        uses: actions/checkout@v3

      - name: Configurar Python
        uses: actions/setup-python@v4
        with:
          python-version: '3.9'

      - name: Instalar Dependencias
        run: pip install requests beautifulsoup4 pdfplumber

      - name: Executar Scraper
```

```
      run: python monitor.py

    - name: Salvar no Github
      run: |
        git config --global user.name "KM Projetos Bot"
        git config --global user.email "bot@kmprojetos.com.br"
        git add .
        git commit -m "Sincronização Automática $(date +'%d/%m/%Y')" || exit 0
        git push
```

---

4. Interface do Site (index.html)

Este código cria uma página moderna para listar os arquivos sem precisar de banco de dados.

HTML

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Monitor DOM - Conceição do Araguaia</title>
    <style>
        body { font-family: 'Segoe UI', sans-serif; background: #f0f2f5; margin: 0; padding: 20px; }
        .container { max-width: 900px; margin: auto; background: white; padding: 30px; border-radius: 12px; box-shadow: 0 4px 6px rgba(0,0,0,0.1); }
        h1 { color: #1a73e8; border-bottom: 3px solid #1a73e8; padding-bottom: 10px; }
        .folder-path { background: #e8f0fe; padding: 10px; margin-top: 20px; font-weight: bold; border-radius: 5px; color: #1967d2; }
        .file-item { display: block; padding: 10px 20px; border-bottom: 1px solid #eee; text-decoration: none; color: #3c4043; transition: 0.2s; }
        .file-item:hover { background: #f8f9fa; color: #1a73e8; }
        footer { margin-top: 50px; text-align: center; font-size: 0.85em; color: #70757a; border-top: 1px solid #eee; padding-top: 20px; }
    </style>
</head>
<body>
    <div class="container">
        <h1>Monitoramento DOM - Conceição do Araguaia</h1>
        <div id="app">Carregando diários...</div>

        <footer>
            <p>Licença: Sistema de Transparência Pública</p>
            <p><strong>Desenvolvido por: Márcio Rodrigues de Oliveira da KM PROJETOS</strong></p>
        </footer>
    </div>
```

```
    <script>
        async function carregarDados() {
            try {
                const res = await fetch('dados.json');
                const data = await res.json();
                const app = document.getElementById('app');
                app.innerHTML = '';

                for (const [path, files] of Object.entries(data)) {
                    const divPath = document.createElement('div');
                    divPath.className = 'folder-path';
                    divPath.innerHTML = '" + path.replace('publicacoes/', '').replace(/\//g, ' > ');
                    app.appendChild(divPath);

                    files.forEach(file => {
                        const a = document.createElement('a');
                        a.className = 'file-item';
                        a.href = path + '/' + file;
                        a.target = '_blank';
                        a.innerHTML = '" + file;
                        app.appendChild(a);
                    });
                }
            } catch (e) {
                document.getElementById('app').innerHTML = 'Nenhum dado encontrado. O robô rodará em breve.';
            }
        }
        carregarDados();
    </script>
</body>
</html>
```

---

5. Como Ativar o Sistema (Passo a Passo)

1. **Crie o Repositório:** Vá no seu GitHub e crie um novo repositório chamado monitor-dom-cda.
2. **Suba os Arquivos:** Envie todos os arquivos acima para o repositório.
3. **Ative o GitHub Pages:**
    - Vá em **Settings** > **Pages**.
    - Em "Build and deployment", selecione a branch main e a pasta root. Clique em Save.
4. **Dê Permissões ao Robô:**
    - Vá em **Settings** > **Actions** > **General**.

    - Role até "Workflow permissions" e marque **"Read and write permissions"**. Isso permite que o robô salve os PDFs no repositório.
5. **Primeira Execução:** Vá na aba **Actions**, clique em "Monitoramento Diario DOM" e clique no botão **Run workflow**.

Licença e Créditos

Este sistema foi projetado para garantir a transparência das publicações municipais de forma automatizada.

Desenvolvido por: Márcio Rodrigues de Oliveira da KM PROJETOS