DOCUMENTAÇÃO OPERACIONAL – UTFPR Conversor de PDF com IA (n8n)

---

1. OBJETIVO DO PROJETO

Automatizar a extração de tabelas de PDFs de patrimônio da UTFPR, gerando automaticamente arquivos CSV prontos para uso, formatados corretamente, sem acentos nos cabeçalhos, facilitando o uso para qualquer usuário.

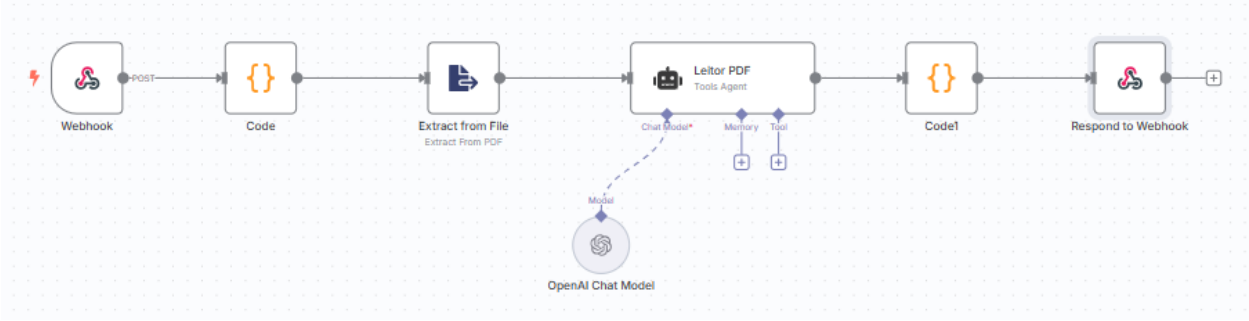
2. ARQUITETURA GERAL DOS FLUXOS

O sistema é composto por dois fluxos principais no n8n:

- **Fluxo de Processamento (POST /conversor):** Recebe o PDF, extrai dados via IA e prepara o resultado.
- **Fluxo de Exibição/Download (GET /utfpr):** Mostra a interface visual, tabela e botão para download do CSV pronto.

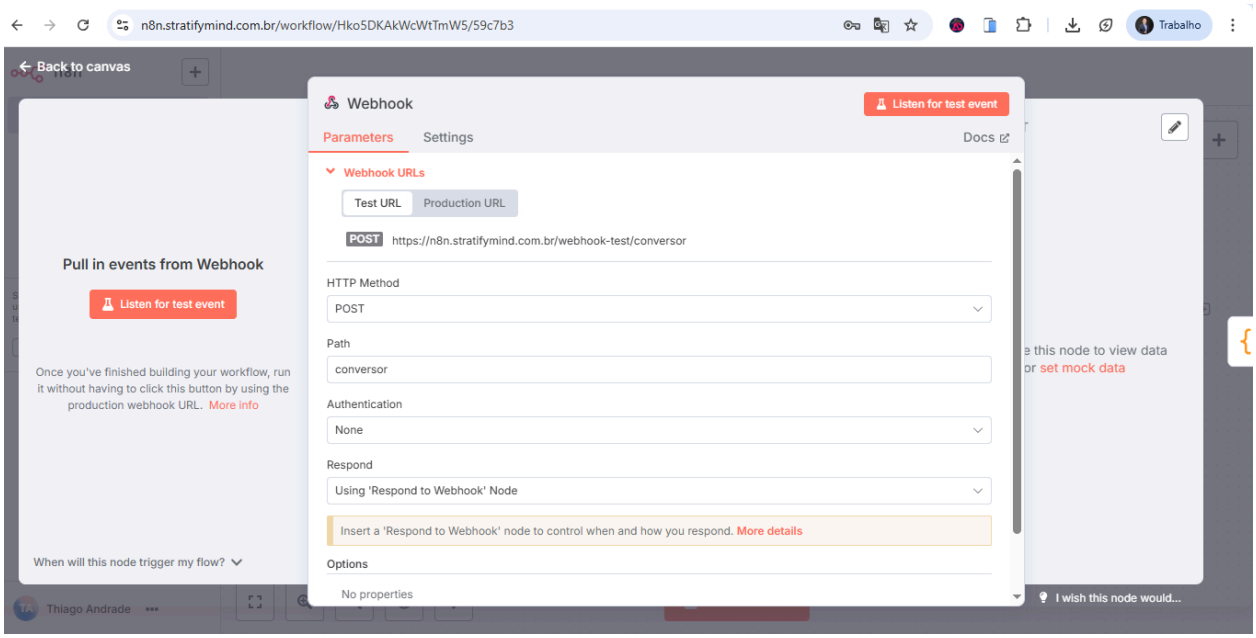
---

3. FLUXO 1: PROCESSAMENTO INTELIGENTE (POST /conversor)



NÓ 1: Webhook (POST)

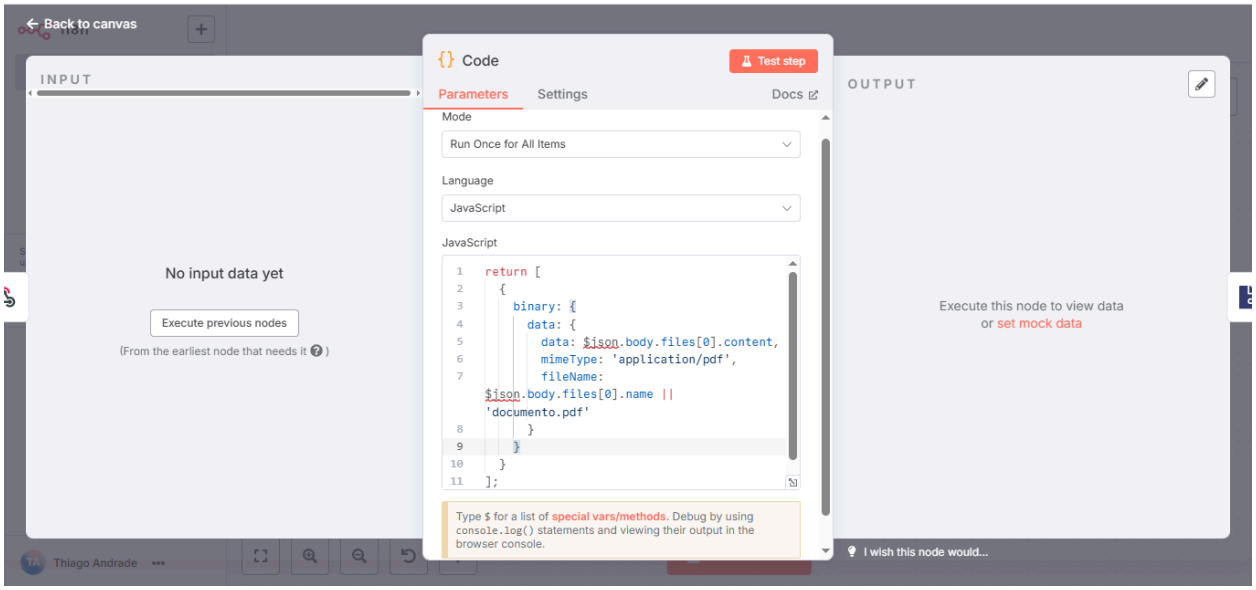
- **Path:** `/conversor`
- **Método:** `POST`
- **Função:** Recebe o PDF enviado pelo usuário para processamento.

- 

NÓ 2: Code (JavaScript)

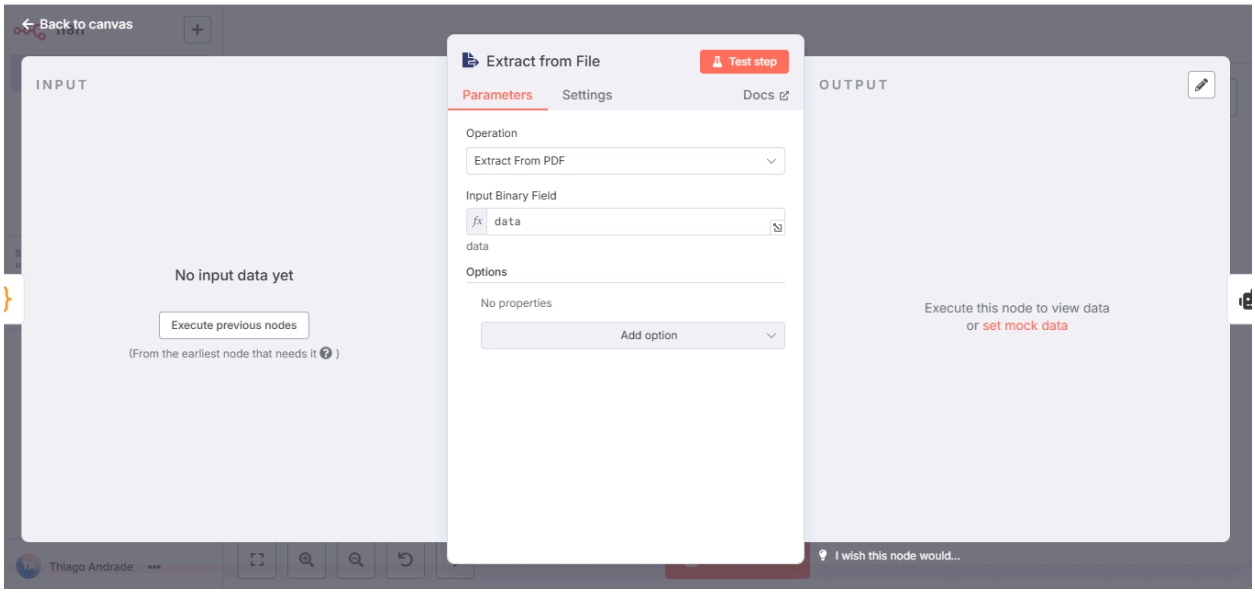
- **Função:** Prepara o arquivo PDF em formato binário para o próximo nó.
- **Código:**

```
return [
  {
    binary: {
      data: {
        data: $json.body.files[0].content,
        mimeType: 'application/pdf',
        fileName: $json.body.files[0].name || 'documento.pdf'
      }
    }
  }
];
```



NÓ 3: Extract from File

- **Operation:** Extract From PDF
- **Input Binary Field:** `data`
- **Função:** Extrai o texto das tabelas do PDF enviado.

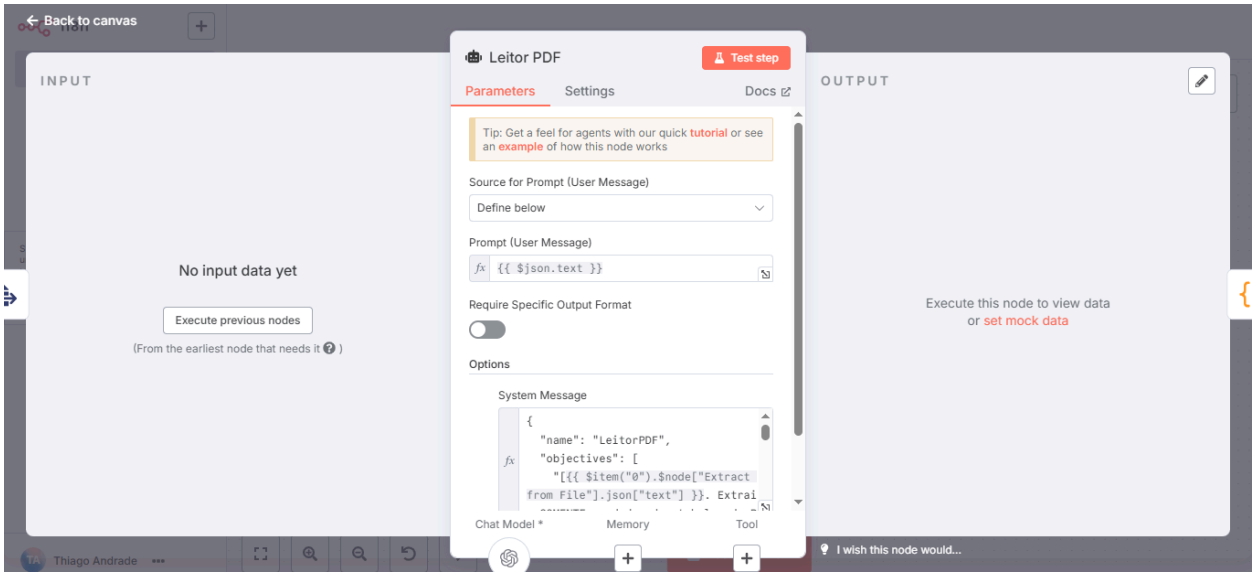


NÓ 4: Leitor PDF (Agent/IA)

- **Modelo:** OpenAI Chat Model + Tools Agent
- **Prompt (exemplo):**

```
{

  "name": "LeitorPDF",

  "objectives": [
```

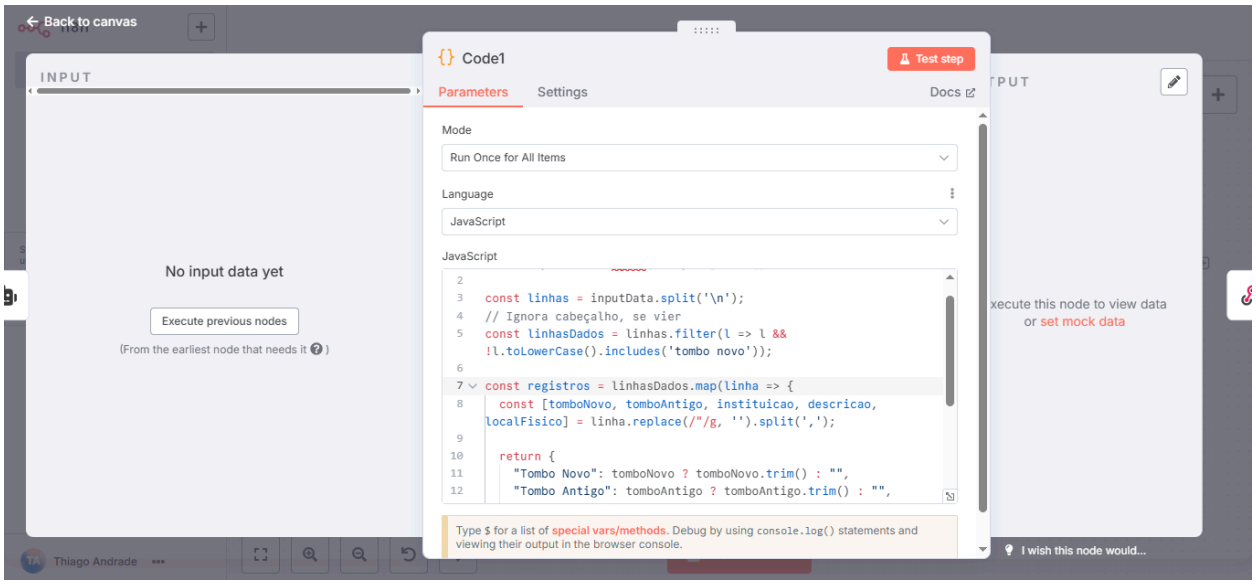
```
    "[{{ $item("0").$node["Extract from File"].json["text"] }}. Extraia
SOMENTE os dados das tabelas do PDF conforme o padrão de colunas abaixo e
retorne EXCLUSIVAMENTE no formato CSV, SEM linhas ou células N/A, sem títulos
extras, sem comentários, exatamente no seguinte formato e separando cada
campo por vírgula, uma linha para cada registro:",

    "",

    "TOMBO NOVO,TOMBO ANTIGO,INSTITUICAO,DESCRICAO DO BEM,LOCAL FISICO",

    "",

    "Exemplo:",

    "834,1333,UTFPR,Cadeira,Q-108",

    "11134,20421,UTFPR,Cadeira,Q-104",

    "",

    "IMPORTANTE:",

    "- Na coluna 'DESCRICAO DO BEM', coloque APENAS o valor que aparece na
coluna 'Descrição' da tabela do PDF.",

    "- NÃO inclua informações de 'Estado de Conservação', 'Situação' ou
qualquer outro campo nesta coluna.",

    "- Cada coluna do CSV deve conter exclusivamente o valor correspondente
da tabela original, sem concatenações ou misturas.",

    "Se algum campo estiver vazio, apenas deixe em branco entre as vírgulas,
sem colocar N/A, null ou aspas.",

    "Não inclua nenhum texto extra, títulos duplicados ou comentários fora do
padrão CSV solicitado."

  ],

  "general_rules": [

    "Seja preciso e rigoroso ao preencher as colunas.",

    "Não coloque aspas em campos vazios ou inteiros.",

    "Não acrescente títulos, comentários, legendas ou qualquer texto fora do
padrão CSV solicitado."

  ]

}
```



NÓ 5: Code1 (JavaScript)

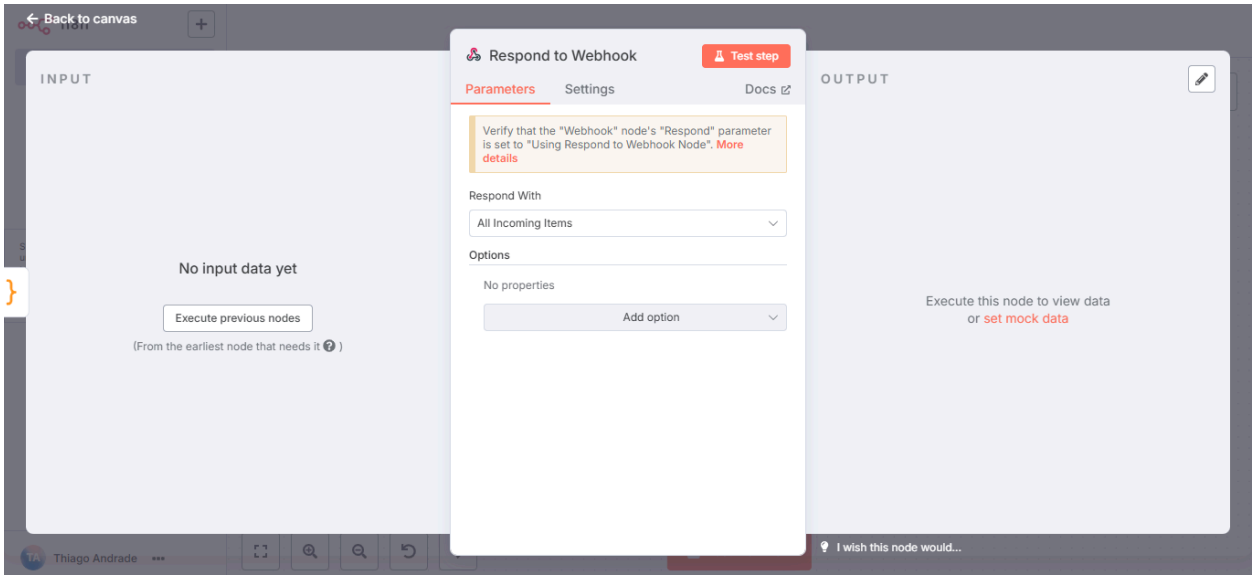
- **Função:** Transforma o texto CSV cru retornado pela IA em objeto estruturado e remove acentos dos cabeçalhos.
- **Exemplo:**

```
const inputData = $json["output"].trim();
const linhas = inputData.split('\n');
const linhasDados = linhas.filter(l => l && !l.toLowerCase().includes('tombo
novo'));
const registros = linhasDados.map(linha => {
  const [tomboNovo, tomboAntigo, instituicao, descricao, localFisico] =
linha.replace(/"/g, '').split(',');
  return {
    "Tombo Novo": tomboNovo ? tomboNovo.trim() : "",
    "Tombo Antigo": tomboAntigo ? tomboAntigo.trim() : "",
    "Instituicao": instituicao ? instituicao.trim() : "",
    "Descricao do Bem": descricao ? descricao.trim() : "",
    "Local Fisico": localFisico ? localFisico.trim() : ""
  };
});
return registros.map(item => ({ json: item }));
```

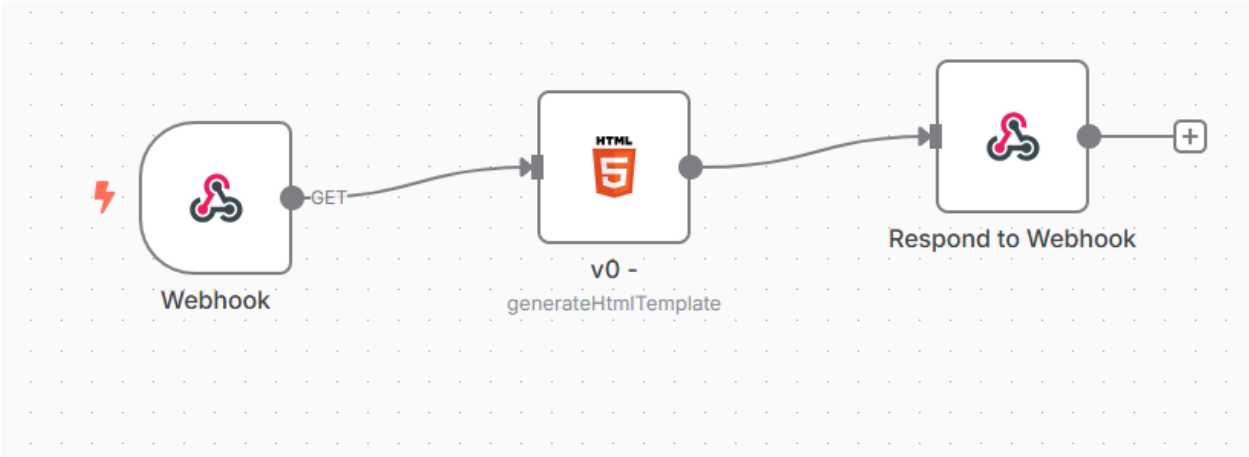


NÓ 6: Respond to Webhook

- **Função:** Retorna todos os registros processados em formato JSON para o frontend ou API integradora.
- **Dica:** Retornar sempre um array de objetos.

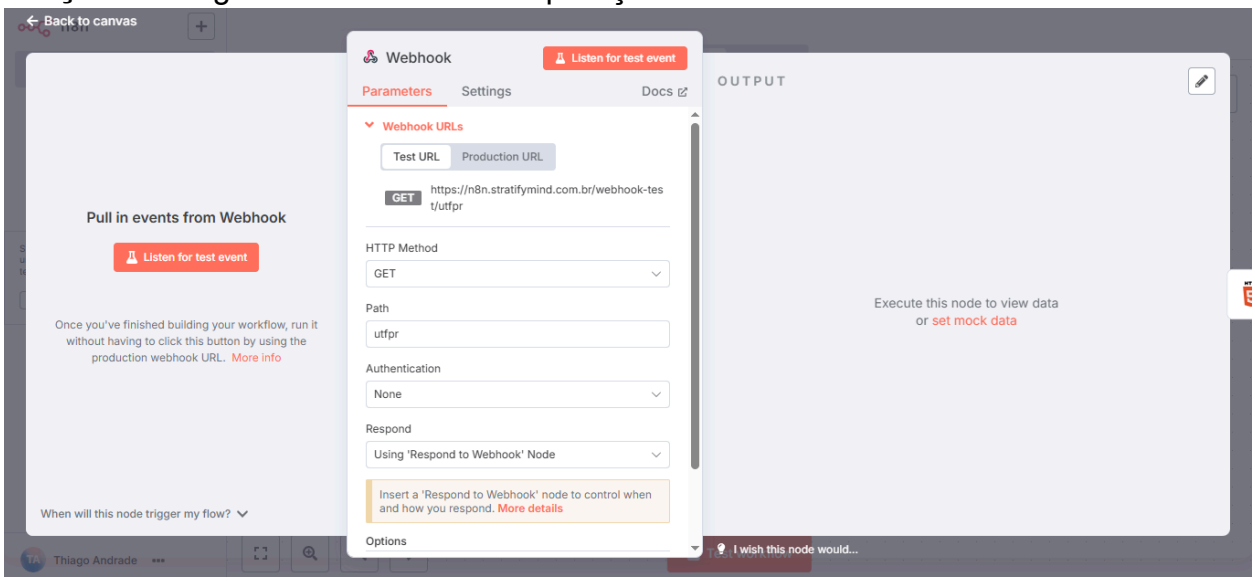


4. FLUXO 2: INTERFACE E DOWNLOAD (GET /utfpr)



NÓ 1: Webhook (GET)

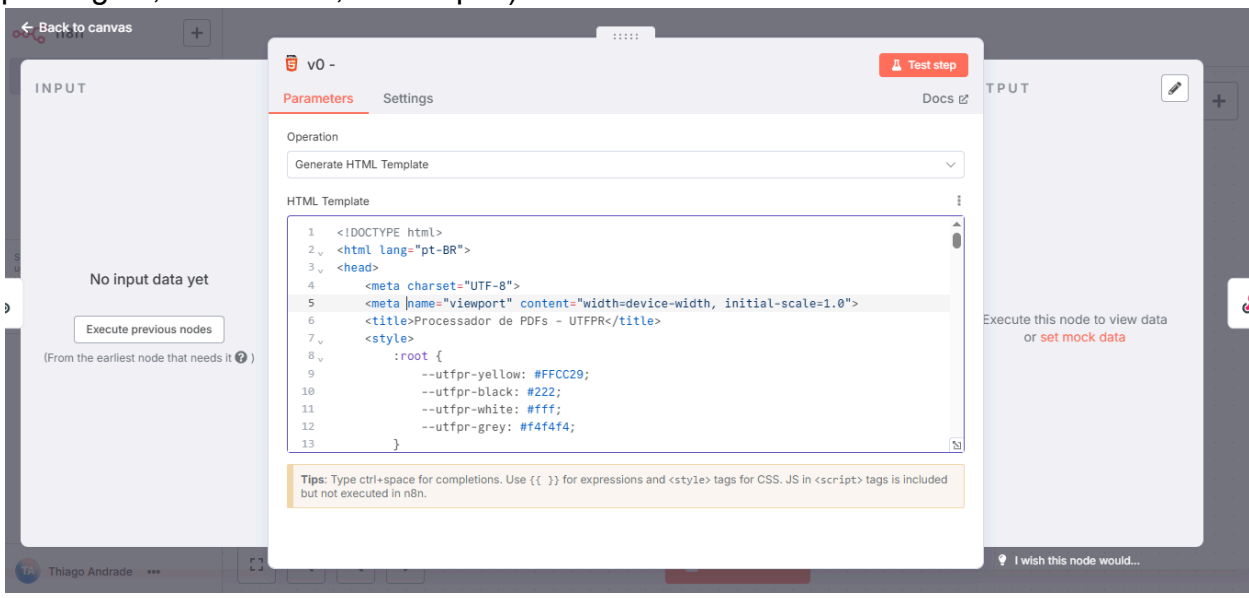
- **Path:** `/utfpr`
- **Método:** GET
- **Função:** Carrega a interface web da aplicação.



NÓ 2: HTML Template

- **Função:**
  - Renderiza a interface visual (painel de sucesso, tabela com colunas limpas e botão Download CSV).

- Utiliza JS interno para gerar o arquivo CSV pronto para Excel (separado por vírgula, sem acento, sem aspas).



- <!DOCTYPE html>
- <html lang="pt-BR">
- <head>
- <meta charset="UTF-8">
- <meta name="viewport" content="width=device-width, initial-scale=1.0">
- <title>Processador de PDFs - UTFPR</title>
- <style>
- :root {
- --utfpr-yellow: #FFCC29;
- --utfpr-black: #222;
- --utfpr-white: #fff;
- --utfpr-grey: #f4f4f4;
- }
- * { margin: 0; padding: 0; box-sizing: border-box; }
- body {
- font-family: 'Segoe UI', Arial, Helvetica, sans-serif;
- background: var(--utfpr-grey);
- min-height: 100vh;
- display: flex;
- align-items: center;
- justify-content: center;
- padding: 20px;

```
        }
        .container {
            max-width: 850px;
            width: 100%;
            background: var(--utfpr-white);
            border-radius: 18px;
            padding: 40px 28px 32px 28px;
            box-shadow: 0 16px 40px rgba(34,34,34,0.13);
            text-align: center;
            border: 2px solid var(--utfpr-yellow);
        }
        h1 {
            color: var(--utfpr-black);
            margin-bottom: 8px;
            font-size: 2.3em;
            font-weight: bold;
            letter-spacing: 1.2px;
        }
        .subtitle {
            color: #555;
            margin-bottom: 30px;
            font-size: 1.08em;
        }
        .upload-area {
            border: 2.5px dashed var(--utfpr-black);
            border-radius: 12px;
            padding: 60px 16px;
            background: linear-gradient(145deg, #fff, #fffbe8 90%);
            transition: all 0.3s ease;
            cursor: pointer;
            margin-bottom: 30px;
            position: relative;
        }
        .upload-area.dragover {
            border-color: var(--utfpr-yellow) !important;
            background: linear-gradient(145deg, #fffde2, #fffad1) !important;
        }
        .upload-icon {
            font-size: 3em;
```

```
            margin-bottom: 12px;
            color: var(--utfpr-yellow);
        }
        .upload-text {
            font-size: 1.15em;
            color: var(--utfpr-black);
            margin-bottom: 7px;
            font-weight: 600;
        }
        .upload-subtext {
            color: #333;
            font-size: 0.98em;
        }
        #fileInput { position: absolute; left: -9999px; opacity: 0; }
        .file-list { margin-bottom: 20px; display: none; }
        .file-item {
            display: flex; align-items: center; justify-content: space-between;
            background: #fffde7;
            padding: 13px 20px; margin-bottom: 9px;
            border-radius: 9px;
            border-left: 4px solid var(--utfpr-yellow);
        }
        .file-info { display: flex; align-items: center; flex: 1; }
        .file-icon { color: var(--utfpr-black); font-size: 1.5em; margin-right:
14px; }
        .file-details h4 { color: var(--utfpr-black); font-size: 1em;
margin-bottom: 2px; font-weight: 600;}
        .file-size { color: #888; font-size: 0.93em; }
        .remove-btn {
            background: var(--utfpr-black); color: var(--utfpr-yellow);
            border: none; padding: 7px 12px; border-radius: 6px;
            cursor: pointer; font-size: 0.92em; font-weight: 500;
        }
        .remove-btn:hover { background: var(--utfpr-yellow); color:
var(--utfpr-black);}
        .process-btn {
            background: var(--utfpr-yellow); color: var(--utfpr-black);
            border: none; padding: 16px 40px; font-size: 1.18em;
```

```
            border-radius: 10px; cursor: pointer; width: 100%; margin-bottom: 18px;
            font-weight: bold; text-transform: uppercase; letter-spacing: 1px;
            box-shadow: 0 2px 12px rgba(34,34,34,0.08);
        }
        .process-btn:hover:not(:disabled) {
            background: var(--utfpr-black); color: var(--utfpr-yellow);
        }
        .process-btn:disabled {
            background: #f1f1f1; color: #aaa;
            cursor: not-allowed; box-shadow: none;
        }
        .progress-section { display: none; background: #fffde2; padding: 30px; border-radius: 13px; margin-bottom: 16px; }
        .progress-bar {
            width: 100%; height: 12px; background: #efefef; border-radius: 7px;
            overflow: hidden; margin: 19px 0 10px 0;
        }
        .progress-fill {
            height: 100%;
            background: linear-gradient(90deg, var(--utfpr-yellow), #fffde2 80%);
            width: 0%; transition: width 0.5s ease;
            position: relative;
        }
        .progress-text { color: var(--utfpr-black); font-size: 1.07em; font-weight: 500;}
        .status-indicator { display: inline-flex; align-items: center; margin-top: 15px; padding: 10px 15px;
            background: #fff6c9; border-radius: 8px; font-size: 0.93em; color: var(--utfpr-black);}
        .status-dot { width: 8px; height: 8px; background: var(--utfpr-yellow); border-radius: 50%; margin-right: 8px; animation: pulse 2s infinite;}
        @keyframes pulse {0%{opacity:1;}50%{opacity:.4;}100%{opacity:1;}}
        .results-section { display: none; text-align: left;}
        .results-title { color: var(--utfpr-black); margin-bottom: 20px; font-size: 1.3em; text-align: center; font-weight: 700;}
        .results-stats {
```

```
            display: grid; grid-template-columns: repeat(auto-fit, minmax(140px, 1fr));
            gap: 13px; margin-bottom: 18px;
        }
        .stat-card {
            background: #fffbe8; padding: 16px; border-radius: 11px;
            text-align: center; box-shadow: 0 2px 10px rgba(34,34,34,0.06);
        }
        .stat-number { font-size: 1.55em; font-weight: 700; color: var(--utfpr-yellow); margin-bottom: 5px;}
        .stat-label { color: #222; font-size: 0.9em; text-transform: uppercase; letter-spacing: 0.3px;}
        .results-table {
            width: 100%; border-collapse: collapse; background: white;
            border-radius: 12px; overflow: hidden;
            box-shadow: 0 2px 13px rgba(34,34,34,0.09); margin-bottom: 17px;
        }
        .results-table th {
            background: var(--utfpr-black); color: var(--utfpr-yellow);
            padding: 14px 11px; text-align: left; font-weight: 600; font-size: 0.95em; text-transform: uppercase;
        }
        .results-table td {
            padding: 13px 11px; border-bottom: 1px solid #eee; font-size: 0.97em; color: #212121;
        }
        .results-table tr:last-child td { border-bottom: none; }
        .results-table tr:nth-child(even) { background: #fffde7; }
        .results-table tr:hover { background: #fffcc7; }
        .error-section {
            display: none; background: #ffe9e9; border: 2px solid #ffb4b4; color: #b00;
            padding: 18px; border-radius: 12px; margin-top: 15px;
        }
        .success-message {
            background: #e5ffdf; border: 2px solid #97c16e; color: #184202;
            padding: 16px; border-radius: 10px; margin-bottom: 22px; text-align: center; font-weight: 500; font-size: 1.07em;
        }
```

```
        .action-buttons { display: flex; gap: 15px; justify-content: center; margin-top: 22px;}
        .new-upload-btn, .download-btn {
            background: var(--utfpr-black); color: var(--utfpr-yellow); border: none; padding: 10px 22px; border-radius: 7px;
            cursor: pointer; font-weight: 500; font-size: 1em; text-decoration: none;
        }
        .download-btn { background: var(--utfpr-yellow); color: var(--utfpr-black);}
        .new-upload-btn:hover, .download-btn:hover { filter: brightness(1.12);}
        .notification {
            position: fixed; top: 20px; right: 20px; padding: 14px 22px; border-radius: 8px; color: white;
            font-weight: 500; z-index: 1000; transform: translateX(400px); transition: transform 0.3s ease;
        }
        .notification.show { transform: translateX(0); }
        .notification.success { background: linear-gradient(135deg, #56cc42, #28a745);}
        .notification.error { background: linear-gradient(135deg, #dc3545, #a00418);}
        .notification.info { background: linear-gradient(135deg, #1c1c1c, #888);}
        .raw-data {
            background: #f9f9f9; border: 1px solid #eee; border-radius: 8px; padding: 13px; margin: 17px 0; font-family: 'Courier New', monospace; font-size: 0.96em;
            max-height: 220px; overflow-y: auto; white-space: pre-wrap; word-break: break-all;
        }
        .debug-section {
            background: #f0f8ff; border: 1px solid #ddd; border-radius: 8px; padding: 15px; margin: 15px 0;
            font-family: 'Courier New', monospace; font-size: 0.9em;
        }
        @media (max-width: 800px) { .container {padding: 18px;} }
    </style>
</head>
<body>
    <div class="container">
```

- <h1>Processador de PDFs - UTFPR</h1>
- Powered by AI - Extração Inteligente de Dados
-
-
- 📁
- Clique ou arraste seus PDFs aqui
- A IA irá extrair dados automaticamente
-
- <input type="file" id="fileInput" multiple accept=".pdf">
-
- <button class="process-btn" id="processBtn" onclick="processFiles()" disabled>🚀 Processar com IA</button>
-
-
- Preparando processamento...
-
-
-
-
-
- Aguardando resposta da IA...
-
-
-
- ✅ Processamento concluído com sucesso pela IA!
-
- <h3 class="results-title">Dados Extraídos pela IA</h3>
-
-
- <button class="download-btn" onclick="downloadResults()">📊 Download CSV</button>
- <button class="new-upload-btn" onclick="resetInterface()">📄 Processar Novos PDFs</button>
-
-

```
        <div class="error-section" id="errorSection">
            <strong>⚠️ Erro no processamento:</strong>
            <div id="errorMessage"></div>
            <div class="action-buttons">
                <button class="new-upload-btn" onclick="resetInterface()">🔄 Tentar Novamente</button>
            </div>
        </div>
    </div>
<script>
let selectedFiles = [];
let extractedData = [];
const WEBHOOK_URL = 'https://n8nwebhook.stratifymind.com.br/webhook/conversor';

document.addEventListener('DOMContentLoaded', function() { initializeEventListeners(); });

function initializeEventListeners() {
    const uploadArea = document.getElementById('uploadArea');
    const fileInput = document.getElementById('fileInput');
    uploadArea.addEventListener('click', function(e) { e.preventDefault(); e.stopPropagation(); triggerFileInput(); });
    ['dragenter','dragover','dragleave','drop'].forEach(ev=>{ uploadArea.addEventListener(ev, preventDefaults, false); document.body.addEventListener(ev, preventDefaults, false); });
    ['dragenter','dragover'].forEach(ev=>uploadArea.addEventListener(ev, highlight, false));
    ['dragleave','drop'].forEach(ev=>uploadArea.addEventListener(ev, unhighlight, false));
    uploadArea.addEventListener('drop', handleDrop, false);
    fileInput.addEventListener('change', function(e){ handleFiles(e.target.files); });
}

function triggerFileInput() {
    const fileInput = document.getElementById('fileInput'); fileInput.click();
}

function preventDefaults(e){e.preventDefault();e.stopPropagation();}
```

```
function highlight(e){document.getElementById('uploadArea').classList.add('dragover');}
function unhighlight(e){document.getElementById('uploadArea').classList.remove('dragover');}
function handleDrop(e){ const dt = e.dataTransfer; handleFiles(dt.files);}

function showNotification(msg, type='info'){
    const exs = document.querySelectorAll('.notification'); exs.forEach(n=>{ if(document.body.contains(n)){document.body.removeChild(n);} });
    const n = document.createElement('div'); n.className = `notification ${type}`; n.textContent = msg; document.body.appendChild(n);
    setTimeout(()=>n.classList.add('show'),100);
    setTimeout(()=>{ n.classList.remove('show'); setTimeout(()=>{ if(document.body.contains(n)){document.body.removeChild(n);} },300); },3000);
}

function handleFiles(files) {
    if(!files||files.length===0){showNotification('Nenhum arquivo selecionado','error');return;}
    const arr=[...files]; let added=0;
    arr.forEach(f=>{
        if(f.type==='application/pdf'){
            if(!selectedFiles.find(sf=>sf.name===f.name&&sf.size===f.size)){
                selectedFiles.push(f); added++;
            }
        } else showNotification(`${f.name} não é um PDF válido`, 'error');
    });
    if(added>0){showNotification(`${added} arquivo(s) adicionado(s)`,'success'); updateFileList();}
    else if(arr.some(f=>f.type==='application/pdf')) showNotification('Todos os PDFs já foram adicionados','info');
}

function updateFileList(){
    const fileList = document.getElementById('fileList');
    const fileItems = document.getElementById('fileItems');
    const processBtn = document.getElementById('processBtn');
```

```
if(selectedFiles.length===0){fileList.style.display='none';processBtn.disabled=true;return;}
    fileList.style.display='block'; fileItems.innerHTML='';
    selectedFiles.forEach((file,index)=>{
        const el=document.createElement('div');
        el.className='file-item';
        el.innerHTML=`<div class="file-info"><div class="file-icon">📄</div><div class="file-details"><h4>${file.name}</h4><div class="file-size">${formatFileSize(file.size)}</div></div></div><button class="remove-btn" onclick="removeFile(${index})">×</button>`;
        fileItems.appendChild(el);
    });
    processBtn.disabled=false;
}

function removeFile(index){ const fileName=selectedFiles[index].name; selectedFiles.splice(index,1); updateFileList(); showNotification(`${fileName} removido`,'info'); }

function formatFileSize(bytes){if(bytes===0)return'0 Bytes';const k=1024,sizes=['Bytes','KB','MB','GB'],i=Math.floor(Math.log(bytes)/Math.log(k));return parseFloat((bytes/Math.pow(k,i)).toFixed(2))+' '+sizes[i];}

function fileToBase64(file){return new Promise((resolve,reject)=>{const r=new FileReader();r.readAsDataURL(file);r.onload=()=>resolve(r.result.split(',')[1]);r.onerror=e=>reject(e);});}

async function processFiles(){
    if(selectedFiles.length===0){showNotification('Selecione pelo menos um PDF','error');return;}

    document.getElementById('uploadSection').style.display='none';
    document.getElementById('progressSection').style.display='block';
    document.getElementById('resultsSection').style.display='none';
    document.getElementById('errorSection').style.display='none';

    const progressFill=document.getElementById('progressFill');
    const progressText=document.getElementById('progressText');
```

```
    const statusText=document.getElementById('statusText');

    try{
        const uploadedFiles=[];
        for(let i=0;i<selectedFiles.length;i++){
            const file=selectedFiles[i],progress=((i+1)/selectedFiles.length)*30;
            progressFill.style.width=progress+'%';
            progressText.textContent=`Preparando ${file.name} (${i+1}/${selectedFiles.length})`;
            const base64=await fileToBase64(file);

uploadedFiles.push({name:file.name,size:file.size,type:file.type,content:base64,uploadTime:new Date().toISOString()});
            await new Promise(r=>setTimeout(r,120));
        }

        const processId='proc_'+Date.now()+'_'+Math.random().toString(36).substr(2,9);
        progressText.textContent='Enviando para IA...'; progressFill.style.width='45%';
        statusText.textContent='Conectando com IA...';

        const payload={
            processId, files: uploadedFiles, configuration: {
                extractionType: 'structured',
                fieldsToExtract: ['tombo_novo', 'tombo_antigo', 'instituicao', 'descricao_bem', 'local_fisico'],
                customPrompt: `Analise cada página dos PDFs e extraia TODOS os dados de inventário de bens patrimoniais encontrados.
                Para cada bem, identifique:
                - Tombo Novo: número de tombamento atual
                - Tombo Antigo: número de tombamento anterior (se houver)
                - Instituição: UTFPR, CEFET, ETFPR ou similar
                - Descrição do Bem: descrição completa do item
                - Local Físico: localização, sala, código do local
                Retorne os dados em formato JSON estruturado, um objeto para cada bem encontrado.
                Seja minucioso e extraia TODOS os bens listados em cada página.`,
                outputFormat: 'json',
```

```
                useAdvancedAI: true
            },
            totalFiles: selectedFiles.length,
            timestamp: new Date().toISOString(),
            source: 'utfpr_pdf_processor'
        };

        let currentProgress=45;
        const progressInterval=setInterval(()=>{
if(currentProgress<92){currentProgress+=Math.random()*6;progressFill.st
yle.width=Math.min(currentProgress,92)+'%';}},1000);

        console.log('Enviando payload:', payload);

        const response=await
fetch(WEBHOOK_URL,{method:'POST',headers:{'Content-Type':'applicati
on/json','Accept':'application/json'},body:JSON.stringify(payload)});

        clearInterval(progressInterval);

        if(response.ok){
            const responseData=await response.json();
            console.log('Resposta recebida:', responseData);

            progressFill.style.width='100%';
            progressText.textContent='Processamento concluído!';
            statusText.textContent='Dados extraídos com sucesso!';

            await new Promise(r=>setTimeout(r,800));
            showResults(responseData);
        } else {
            const errorText=await response.text();
            console.error('Erro na resposta:', response.status, errorText);
            throw new Error(`Erro do servidor: ${response.status} -
${errorText}`);
        }
    }catch(error){
        console.error('Erro no processamento:', error);
        showError(error.message);
```

```
        showNotification('Erro no processamento','error');
    }
}

function showResults(data){
    document.getElementById('progressSection').style.display='none';
    document.getElementById('resultsSection').style.display='block';
    const resultsContainer=document.getElementById('resultsContainer');
    const resultsStats=document.getElementById('resultsStats');
    extractedData = [];
    // Processar diferentes tipos de resposta
    if (Array.isArray(data)) {
        extractedData = data;
    } else if (data && typeof data === 'object') {
        if (data.output) {
            if (typeof data.output === 'string') {
                try { extractedData = JSON.parse(data.output); }
                catch (e) { extractedData = []; }
            } else if (Array.isArray(data.output)) {
                extractedData = data.output;
            } else if (typeof data.output === 'object') {
                extractedData = [data.output];
            }
        } else if (data.data && Array.isArray(data.data)) {
            extractedData = data.data;
        } else if (data.results && Array.isArray(data.results)) {
            extractedData = data.results;
        } else {
            const hasExpectedFields = Object.keys(data).some(key =>
                key.toLowerCase().includes('tombo') ||
                key.toLowerCase().includes('instituicao') ||
                key.toLowerCase().includes('descricao')
            );
            if (hasExpectedFields) {
                extractedData = [data];
            } else {
                for (const [key, value] of Object.entries(data)) {
                    if (Array.isArray(value) && value.length > 0) {
                        extractedData = value;
```

```
                    break;
                }
            }
        }
    }
    if (!extractedData || extractedData.length === 0) {
        resultsContainer.innerHTML = `
            <div class="debug-section">
                <h4>Debug - Resposta recebida:</h4>
                <pre>${JSON.stringify(data, null, 2)}</pre>
            </div>
            <p style="text-align: center; color: #666; margin: 20px 0;">
                Nenhum dado estruturado foi extraído. Verifique o formato da resposta acima.
            </p>
        `;
        resultsStats.innerHTML = `
            <div class="stat-card"><div class="stat-number">${selectedFiles.length}</div><div class="stat-label">Arquivos</div></div>
            <div class="stat-card"><div class="stat-number">0</div><div class="stat-label">Itens Extraídos</div></div>
            <div class="stat-card"><div class="stat-number">0</div><div class="stat-label">Instituições</div></div>
        `;
    } else {
        displayStructuredData();
        const totalItems = extractedData.length;
        const uniqueInstitutions = [...new Set(extractedData.map(item => {
            return item.instituicao || item.Instituição || item.INSTITUICAO || item.institution || '';
        }).filter(Boolean))];
        resultsStats.innerHTML = `
            <div class="stat-card"><div class="stat-number">${selectedFiles.length}</div><div class="stat-label">Arquivos</div></div>
            <div class="stat-card"><div class="stat-number">${totalItems}</div><div class="stat-label">Itens Extraídos</div></div>
```

```
            <div class="stat-card"><div class="stat-number">${uniqueInstitutions.length}</div><div class="stat-label">Instituições</div></div>
        `;
    }
}

/** AJUSTE AQUI: função fiel aos campos do seu JSON **/
function displayStructuredData(){
    const resultsContainer = document.getElementById('resultsContainer');
    if (extractedData.length > 0) {
        let tableHTML = `
            <table class="results-table">
                <thead>
                    <tr>
                        <th>Tombo Novo</th>
                        <th>Tombo Antigo</th>
                        <th>Instituicao</th>
                        <th>Descricao do Bem</th>
                        <th>Local Fisico</th>
                    </tr>
                </thead>
                <tbody>
        `;
        extractedData.forEach(item => {
            // Use apenas os campos exatos do JSON retornado pelo Code1!
            const tomboNovo = item["Tombo Novo"] || '-';
            const tomboAntigo = item["Tombo Antigo"] || '-';
            const instituicao = item["Instituição"] || '-';
            const descricao = item["Descricao do Bem"] || item["Descrição do Bem"] || '-'; // cobre ambos os casos
            const local = item["Local Físico"] || '-';

            tableHTML += `
                <tr>
                    <td>${tomboNovo}</td>
                    <td>${tomboAntigo}</td>
                    <td>${instituicao}</td>
                    <td>${descricao}</td>
```

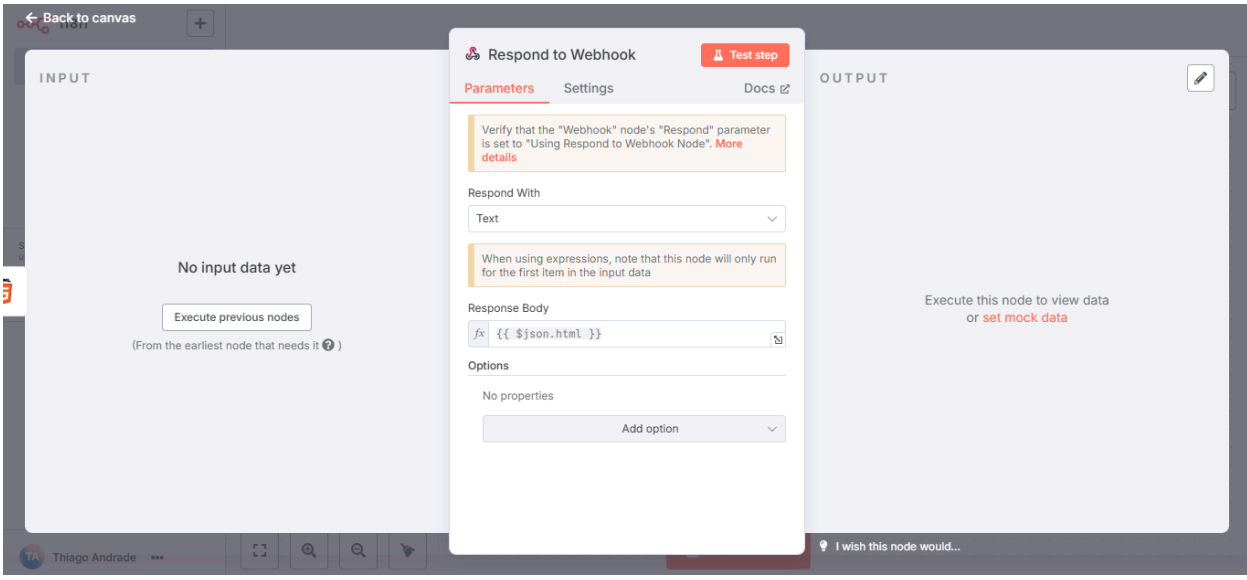
```
                    <td>${local}</td>
                </tr>
            `;
        });
        tableHTML += '</tbody></table>';
        resultsContainer.innerHTML = tableHTML;
    } else {
        resultsContainer.innerHTML = '<p style="text-align: center; color:
#666;">Nenhum dado foi extraído dos PDFs.</p>';
    }
}

function showError(msg){
    document.getElementById('progressSection').style.display='none';
    document.getElementById('errorSection').style.display='block';
    document.getElementById('errorMessage').textContent=msg;
}

function downloadResults(){
    if(!extractedData||extractedData.length===0){
        showNotification('Nenhum dado para download','error');
        return;
    }
    const headers = [
        'Tombo Novo',
        'Tombo Antigo',
        'Instituicao',
        'Descricao do Bem',
        'Local Fisico'
    ];
    const csvContent = [
        headers.join(';'),
        ...extractedData.map(item => {
            const row = [
                item["Tombo Novo"] || '',
                item["Tombo Antigo"] || '',
                item["Instituição"] || '',
                item["Descricao do Bem"] || item["Descrição do Bem"] || '',
                item["Local Físico"] || ''
```

```
            ];
            return row.map(field => {
                let valor = String(field).replace(/"/g, '""');
                if (valor.includes(';') || valor.includes('\n')) {
                    return `"${valor}"`;
                }
                return valor;
            }).join(';');
        })
    ].join('\r\n');
    const blob = new Blob([csvContent], { type: 'text/csv;charset=utf-8;' });
    const link = document.createElement('a');
    const url = URL.createObjectURL(blob);
    link.setAttribute('href', url);
    link.setAttribute('download', `dados_utfpr_${new
Date().toISOString().split('T')[0]}.csv`);
    link.style.visibility = 'hidden';
    document.body.appendChild(link);
    link.click();
    document.body.removeChild(link);
    showNotification('Download iniciado!', 'success');
}

function resetInterface() {
    selectedFiles = [];
    extractedData = [];
    document.getElementById('uploadSection').style.display = 'block';
    document.getElementById('progressSection').style.display = 'none';
    document.getElementById('resultsSection').style.display = 'none';
    document.getElementById('errorSection').style.display = 'none';
    document.getElementById('fileList').style.display = 'none';
    document.getElementById('fileItems').innerHTML = '';
    document.getElementById('processBtn').disabled = true;
    document.getElementById('progressFill').style.width = '0%';
    const fileInput = document.getElementById('fileInput');
    fileInput.value = '';
    showNotification('Interface resetada', 'info');
}
</script>
```

- </body>
- </html>
-
-

NÓ 3: Respond to Webhook

- **Função:**
  - Retorna a página para o navegador.
  - Dica: Sempre use o tipo de resposta `Text` para o HTML renderizar.



---

5. PRINCIPAIS BOAS PRÁTICAS E DICAS

- Utilize sempre PDFs “nativos” (não apenas imagem) para extração de tabelas.
- Simplifique o prompt da IA, pedindo “apenas dados CSV, nada além”.
- Valide o arquivo CSV baixado, testando no Excel.
- Se mudar o layout das tabelas no PDF, ajuste prompt da IA e parsing do JS.
- Se adicionar colunas novas, lembre de atualizar o JS e o HTML da tabela/CSV.
- Sempre remova acentos e caracteres especiais dos títulos para evitar problemas no Excel.

---

6. MANUTENÇÃO E TROUBLESHOOTING

- **Problema de extração?**
  - Verifique se o PDF não está apenas como imagem/escaneado.
- **IA devolvendo texto errado?**
  - Reforce o prompt (“nada além dos dados CSV, sem título, sem comentário”).

- **CSV baixa com colunas bagunçadas?**
    - Ajuste o JS de geração do CSV para garantir separação só por vírgula e sem aspas nos campos.
- **Erros HTTP:**
    - Use debug/log do n8n e revise se os dados não estão chegando como undefined.

---

7. FLUXO DE USO – PASSO A PASSO

1. Usuário faz upload do PDF via formulário integrado ao webhook POST `/conversor`.
2. PDF é processado, IA extrai as tabelas, dados são limpos no JavaScript e devolvidos como JSON.
3. Frontend recebe, monta a visualização na rota GET `/utfpr`.
4. Usuário visualiza tabela e pode baixar CSV já pronto, direto para o Excel, sem etapas técnicas.

---

8. NOTAS FINAIS

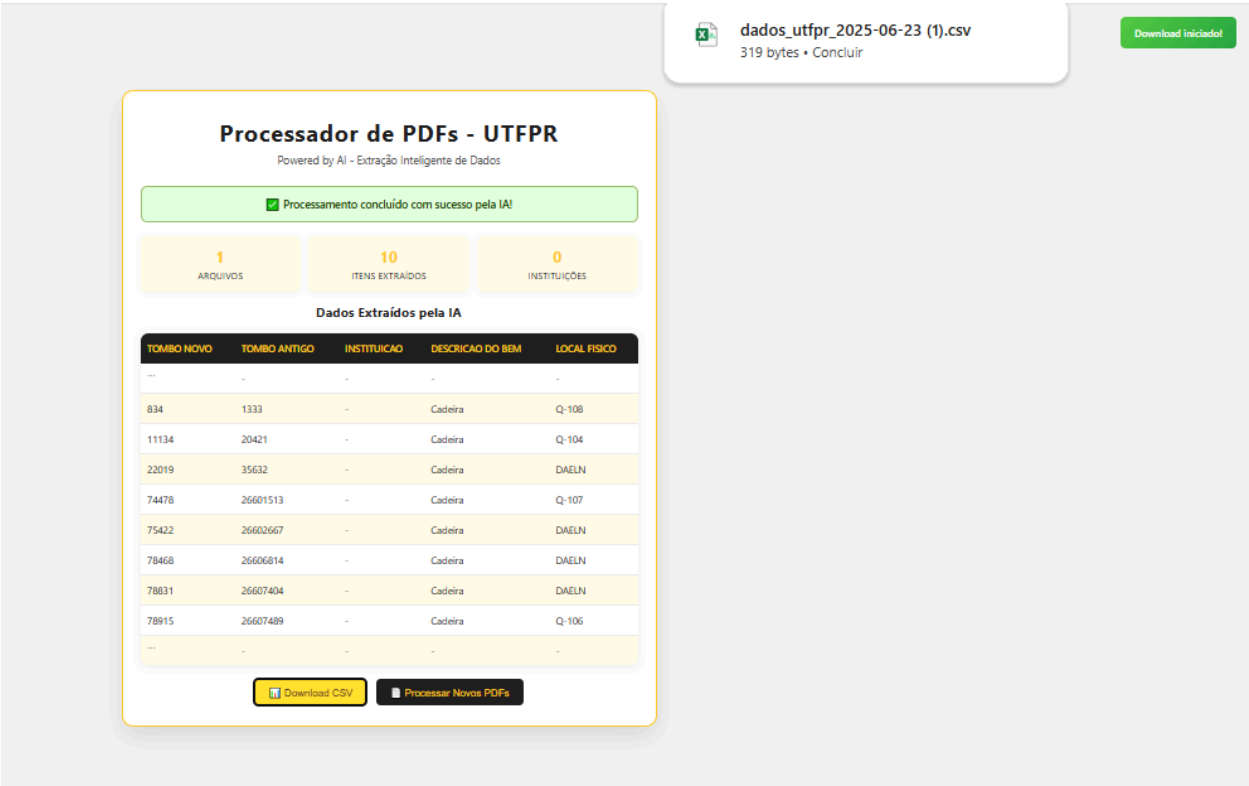
- O sistema pode ser facilmente ajustado para novas necessidades (mais campos, novas regras, outros formatos).
- Para dúvidas ou manutenção, sempre revise o prompt da IA e os scripts de parsing.

---

9. INTERFACE USUARIO


Processador de PDFs - UTFPR
Powered by AI - Extração Inteligente de Dados
Enviando para IA...
Conectando com IA...


Processador de PDFs - UTFPR
Powered by AI - Extração Inteligente de Dados
Processamento concluído com sucesso pela IA!
1
ARQUIVOS
10
ITENS EXTRAÍDOS
0
INSTITUIÇÕES
Dados Extraídos pela IA
TOMBO NOVO
TOMBO ANTIGO
INSTITUICAO
DESCRICAO DO BEM
LOCAL FISICO
...
-
-
-
-
834
1333
-
Cadeira
Q-108
11134
20421
-
Cadeira
Q-104
22019
35632
-
Cadeira
DAELN
74478
26601513
-
Cadeira
Q-107
75422
26602667
-
Cadeira
DAELN
78468
26606814
-
Cadeira
DAELN
78831
26607404
-
Cadeira
DAELN
78915
26607489
-
Cadeira
Q-106
...
-
-
-
-
Download CSV
Processar Novos PDFs



Csv Gerado:

A	B	C	D	E
Tombo Novo	Tombo Antigo	Instituicao	Descricao do Bem	Local Fisico
``				
834	1333		Cadeira	Q-108
11134	20421		Cadeira	Q-104
22019	35632		Cadeira	DAELN
74478	26601513		Cadeira	Q-107
75422	26602667		Cadeira	DAELN
78468	26606814		Cadeira	DAELN
78831	26607404		Cadeira	DAELN
78915	26607489		Cadeira	Q-106
``				