

# PROJETO FISCALTRON



## Grupo Computron Agent

Outubro de 2025

Integrantes:

André Matos	Maycon Abe
Sidney Wergles	Filipe Andrade
Thiago Souza	Mauricio Abe
Luciano Scagliusi	
Harding Leite	

## Desafio 3 – Entrega final

### Sumário Executivo – Tema

O Projeto FiscalTron representa uma evolução no campo da gestão tributária, unindo tecnologia de ponta à necessidade real de eficiência fiscal. Trata-se de uma plataforma inteligente desenvolvida com base em **agentes autônomos de inteligência artificial**, capazes de operar de forma contínua, realizando análises fiscais detalhadas e automatizadas. Seu foco está na validação de informações tributárias e detecção de inconsistências.

Com o avanço das legislações tributárias e o dinamismo do ambiente regulatório, empresas enfrentam dificuldades para manter a conformidade de forma manual. O FiscalTron surge como uma resposta a esse desafio, atuando como um sistema que aprende com os dados, adapta-se às regras fiscais locais e apresenta resultados de forma contextualizada e compreensível. Sua principal proposta de valor está na automação da verificação fiscal.

## 1. Fluxo de trabalho do FiscalTron



A dinâmica de interação com o usuário no Projeto FiscalTron foi desenhada para ser simultaneamente eficiente e lúdica, conduzida por meio de um personagem chamado FiscalTron, do qual o usuário interage via chatbot, essa interface entrega a sensação de um verdadeiro diálogo com o “FiscalTron”. Desde o primeiro contato, o usuário se depara com uma experiência agradável, pois o chatbot foi pensado para personificar o robô FiscalTron: linguagem amigável, respostas claras e até toques sutis de humor que reforçam a ideia de

estar conversando com um agente inteligente de inteligência artificial. Essa abordagem humanizada faz com que o usuário se sinta acolhido e confiante ao submeter suas notas fiscais para o devido tratamento.

### 1.1. Mini-Agente de Extração de Informações

Logo após o recebimento das notas, este agente inicia o processo de leitura automatizada dos documentos via OCR. Ele é responsável por localizar campos-chave (valores, datas, CNPJ/CPF, itens fiscais etc.) e estruturar esses dados em formulários internos. Essa etapa é crítica para garantir que toda informação seja capturada de maneira precisa, servindo de base para os próximos passos.



### 1.2. Mini-Agente de Interação com LLM



Com os dados extraídos, o segundo mini-agente entra em cena para “consultar” um modelo de linguagem de larga escala (LLM). Sua função é contextualizar cláusulas, regras e doutrinas tributárias que possam influenciar a interpretação dos documentos. Esse agente traduz, por exemplo, normativos complexos em instruções práticas, ajudando a embasar as próximas validações.

### 1.3. Mini-Agente de Validação Específica

Paralelamente, existem outros mini-agentes focados em diferentes tipos de checagens automatizadas. Por exemplo:



- **Validador do estado da federação da nota:** Confere se a nota é referente ao estado de São Paulo, caso encontre não seja, a nota é redirecionada para um repositório específico, por conseguinte, comunica diretamente com o Alert-Tron sobre a inconsistência.
- **Validador da nota (nível 1):** Verifica se a nota segue os padrões exigidos pela legislação, tais como: validade do cpf, cnpj e o cálculo do ISS.

• **Validador de cruzamento de dados:** Realiza a checagem cruzada entre os valores declarados na nota em outros sistemas via API, caso encontre inconsistências, comunica diretamente com o Alert-Tron.



- O **Data-Tron** é o agente responsável por consolidar, organizar e tornar acessíveis todas as informações relevantes para o sistema. Seu trabalho pode ser descrito em três etapas principais, coleta e Pré-processamento; geração de Vetores (Embeddings) e armazenamento em Banco de Dados Vetorial.

O Data-Tron funciona como o armazenador de conhecimento do sistema, mantendo tudo gravado em formato vetorial para que qualquer busca, inferência ou análise posterior seja feita de forma rápida e semanticamente precisa.



### 1.4. Consciência situacional das ações

Enquanto cada mini-agente executa sua tarefa, o FiscalTron mantém o usuário informado através de mensagens curtas no chatbot, indicando o status ("Estou validando os impostos...", "Consultando orientações fiscais para esse cenário...", "Verificando duplicidade de chave de nota..."). Caso algum mini-agente identifique uma inconsistência grave, por exemplo, divergência de alíquota, nota emitida em duplicidade ou ausência de dados obrigatórios, o FiscalTron notifica o usuário de imediato, exibindo uma mensagem clara e orientações sobre como proceder (corrigir o XML). Por outro lado, se tudo estiver em ordem, o FiscalTron parabeniza o usuário e entrega um relatório resumido das validações realizadas.



## 1.5. O maestro de uma grande orquestração



O **FiscalTron** é o grande maestro de todo o ecossistema de mini-agentes inteligentes, recebendo a missão de orquestrar cada etapa do processo de tratamento de documentos fiscais, de modo que cada componente desde a extração de dados até a geração de alertas encaixe-se de forma coesa e harmônica.

## 1.6. Visão Estratégica e Planejamento Global

Antes de qualquer análise pontual, cabe ao FiscalTron estudar o cenário como um todo: quais tipos de documentos serão processados, quais bases de dados devem ser consultadas, quais regras fiscais precisam ser aplicadas e quais níveis de criticidade devem ser monitorados. A partir dessa apreciação estratégica, o FiscalTron projeta o “mapa de execução”, determinando a sequência ótima de ações.



## 1.7. Alocação de Tarefas aos Mini-Agentes

Uma parte importante da arquitetura são os mini-agentes especializados, cada um responsável por um conjunto específico de funções (extração, pesquisa de doutrina, validação, notificação de alertas, gerenciamento de dados). O FiscalTron, ao receber um lote de notas fiscais, avalia o perfil de cada documento e, então, dispara comandos para:

- Extrator-Tron:
- Researcher-Tron:
- Validador-Tron:
- Alert-Tron:
- Data-Tron:

## 2. Objetivos do FiscalTron

O objetivo principal do FiscalTron é disponibilizar uma plataforma capaz de:

- Detectar divergências e inconsistências em notas fiscais eletrônicas do estado de São Paulo (NF-e, NFS-e);
- Fornecer alertas informativos apontado as inconsistências, antecipando riscos fiscais e prevenindo autuações;

## 3. Público Alvo

O **FiscalTron** é direcionado a empresas de médio e grande porte que atuam em ambientes tributários complexos e lidam com alto volume de emissão e recebimento de documentos fiscais eletrônicos vinculados ao estado de São Paulo. Seu público-alvo inclui departamentos fiscais e contábeis de empresas dos setores de serviços, tecnologia, indústria, saúde, educação e construção civil, que enfrentam desafios recorrentes para garantir a conformidade com normas tributárias municipais, estaduais e federais. Além disso, escritórios de contabilidade, consultorias tributárias que gerenciam múltiplos clientes e operações descentralizadas se beneficiam imensamente da automação e inteligência preditiva oferecidas pela solução.

## 4. Justificativa do tema

A justificativa para o desenvolvimento do Projeto FiscalTron está na necessidade de modernizar os processos fiscais e contábeis frente ao cenário cada vez mais desafiador da administração tributária brasileira. com a proliferação de normas fiscais, variações entre legislações municipais e a digitalização das obrigações acessórias, torna-se praticamente inviável manter a conformidade apenas com processos manuais ou com sistemas tradicionais. Isso gera riscos significativos para as organizações, como autuações, passivos fiscais ocultos, multas por descumprimento e retrabalho administrativo. Além disso, a crescente pressão por transparência, rastreabilidade e eficiência por parte das administrações públicas reforça a demanda por soluções mais inteligentes e responsivas.

Nesse contexto, a inteligência artificial surge como uma aliada estratégica, oferecendo recursos como agentes cognitivos e automação de validações. O FiscalTron utiliza esses recursos para transformar grandes volumes de dados fiscais e contábeis em informações úteis, confiáveis e acionáveis. Ao empregar agentes inteligentes que monitoram, cruzam e interpretam informações em tempo real, o sistema não apenas reduz erros e inconsistências, mas também libera as equipes fiscais e administrativas para tarefas de maior valor agregado. Assim, o projeto

justifica-se não apenas pela inovação tecnológica, mas pela sua capacidade de promover governança tributária, redução de riscos e eficiência operacional, inserindo as organizações de forma definitiva na era da inteligência fiscal.

## **5. Proposta preliminar de como o projeto será desenvolvido**

### **5.1. A Equipe Computron Agents**

O desenvolvimento do Projeto FiscalTron será conduzido pela equipe Computron, um grupo técnico multidisciplinar formado por nove integrantes capacitados, com formação em tecnologia da informação, ciência de dados e inteligência artificial aplicada. Cada membro da equipe traz uma bagagem significativa em projetos.

A atuação dos membros da equipe será distribuída por especialidade: alguns focarão na concepção da solução, estrutura da orquestração das ações, outros na implementação das automações, e em apoio ao trabalho de validação fiscal, além da integração com os sistemas públicos. Essa sinergia garante um time equilibrado entre tecnologia, modelagem fiscal e aplicação de inteligência artificial, permitindo a entrega de uma solução funcional desde os primeiros ciclos de desenvolvimento.

### **5.2. A arquitetura da solução**

A arquitetura do FiscalTron foi concebida com base nos princípios da modularidade. O núcleo do sistema será sustentado por agentes inteligentes conectados a modelos de linguagem de larga escala (LLM), que fornecerão a capacidade de extração das informações, compreensão semântica e contextual dos documentos fiscais. Esses agentes atuarão como copilotos tributários, capazes de validar, correlacionar e emitir alertas com base em divergências.

Para orquestrar esses fluxos, será utilizado a plataforma n8n, que funcionará como a espinha dorsal da comunicação entre os diversos módulos da solução. O n8n permitirá a criação de pipelines automatizados para ingestão, pré-processamento e validação de dados, bem como o acionamento de agentes de IA e das rotinas de verificação fiscal. Dessa forma, cada etapa — desde a submissão das notas fiscais até a emissão de alertas — poderá ser rastreada.



Figura 1 – Arquitetura funcional

A Figura 1 representa a arquitetura funcional do FiscalTron, um sistema fiscal inteligente orquestrado por automações e agentes baseados em IA, com objetivo de garantir conformidade tributária de forma escalável, transparente e autônoma. A seguir, cada componente do FiscalTron será descrito, com ênfase na função e na relação com os agentes inteligentes e o orquestrador n8n.

### 5.3. XML/Documentos Fiscais

A entrada do sistema ocorre por meio da recepção de documentos fiscais eletrônicos, como XMLs e .pdf de NF-e, NFS-e. Esses arquivos são oriundos de sistemas de gestão empresarial, prefeituras ou outras fontes e contêm os dados brutos das operações comerciais. Os agentes inteligentes do FiscalTron iniciam sua atuação com a leitura e categorização automatizada desses documentos, ativada por gatilhos no orquestrador n8n.

### 5.4. Extrator de Informações

Este componente é responsável pelo pré-processamento e extração das informações fiscais essenciais dos documentos recebidos. Utilizando automações desenvolvidas no n8n e apoio de modelos semânticos, o extrator organiza os dados

em estruturas compreensíveis para os agentes de IA. Campos como CNPJ, valores, alíquotas e outras informações.

## **5.5. FiscalTron (Maestro)**

O coração do sistema é o agente FiscalTron, representado por uma instância cognitiva capaz de coordenar as ações dos demais componentes. Ele opera como o copiloto tributário digital, ativando LLM local. O FiscalTron é acionado e gerenciado pelo n8n, que define o fluxo de trabalho baseado em eventos ou rotinas definidas.

## **5.6. LLM**

Aqui reside a inteligência cognitiva do FiscalTron: um modelo de linguagem de larga escala (LLM). Ele fornece capacidade de interpretação contextual, geração de explicações. Ao receber dados processados, o LLM valida se há conformidade e divergência ou inconsistência. É essa camada que transforma o sistema em algo mais do que um simples verificador.

## **5.7. Validador**

O validador é o componente responsável pela checagem formal e legal dos dados extraídos e processados. Ele aplica as regras fiscais parametrizadas no projeto. A interação entre o FiscalTron, o LLM e o validador é mediada pelo n8n e resulta em um processo de verificação contínua, em tempo real.

## **5.8. Alertas**

Quando o sistema detecta divergências, omissões ou inconsistências fiscais, os agentes geram alertas informativos personalizados. Esses alertas podem ser direcionados enviados para os responsáveis via chatbot. Cada alerta contém explicações geradas pelo LLM, recomendações de ação e links rastreáveis para evidências.

## **5.9. Banco de Dados Vetorial**

O sistema armazena os documentos processados, embeddings semânticos gerados pelo LLM e histórico de interações em um banco de dados vetorial.

## **5.10. Plataforma orquestradora n8n**

Por fim, todo o fluxo operacional — desde a ingestão de arquivos, extração de dados, ativação dos agentes, geração de alertas é orquestrado pelo n8n. Este orquestrador atua como a "espinha dorsal" da automação, permitindo modularidade,

rastreabilidade e facilidade de integração com sistemas externos. Ele é o componente que assegura que os agentes trabalhem em harmonia, dentro de regras, e com autonomia controlada.

## **6. Conclusão**

O projeto FiscalTron materializa uma solução inovadora para os desafios contemporâneos de conformidade tributária, ao empregar uma arquitetura baseada em agentes autônomos de inteligência artificial voltados para o estado de São Paulo. Essa plataforma, ainda em fase de desenvolvimento, propõe a automatização de processos que até então eram executados de forma manual e suscetíveis a falhas humanas: extração de dados de notas fiscais, validação de cálculos e alíquotas, geração de alertas personalizados e armazenamento organizado das informações em bases de dados. Ao integrar esses componentes em um fluxo orquestrado pelo próprio FiscalTron, garante-se não apenas a consistência de cada etapa, mas também a sinergia entre elas, resultando em um ciclo contínuo de aprendizado e adaptação.

## 7. Status Report – Evolução do Projeto FiscalTron

### 7.1. Introdução

Este relatório apresenta o andamento do projeto de automação da leitura de Notas Fiscais utilizando OCR e integração com agentes inteligentes, implementado na plataforma n8n. O objetivo principal é automatizar o processo de captura, leitura e estruturação de dados extraídos de arquivos PDF de notas fiscais, promovendo eficiência, rastreabilidade e integração com outros sistemas.

### 7.2. Descrição Técnica da Solução

A solução foi construída utilizando o **n8n**, uma plataforma de automação baseada em workflows visuais. O fluxo principal é dividido em três macroetapas:

**Upload e Monitoramento:** Um agente inteligente monitora um repositório de arquivos onde notas fiscais em formato PDF são disponibilizadas por diferentes fornecedores.

**Leitura e Extração OCR:** Utilizando o módulo OCR do Tesseract embutido no n8n, o fluxo extrai os dados contidos nas notas, incluindo CNPJ, data de emissão, valor, nome da empresa e número da nota.

**Estruturação e Envio:** Os dados extraídos são padronizados e enviados para uma base de dados relacional ou sistema ERP via API RESTful.

Os principais nós utilizados incluem:

- **HTTP Request** para integração com APIs externas;
- **Set e Function** para manipulação de dados;
- **If e Switch** para controle de fluxo;
- **Read PDF e OCR** para extração textual.

### 7.3. Inconsistências e Ações Corretivas

Durante a execução, a equipe identificou **inconsistências no processo de leitura dos arquivos**, principalmente em notas com má qualidade de digitalização ou layout não padronizado. Os principais problemas detectados foram:

- Erros de reconhecimento de campos críticos, como número da nota ou CNPJ;
- Falhas na extração de textos em campos com múltiplas linhas;

- Dificuldades na padronização dos dados para posterior integração.

Para enfrentar essas questões, a equipe técnica realizou **três reuniões consecutivas**, com foco na análise das falhas e proposição de melhorias. As ações corretivas envolvem:

- Reconfiguração dos parâmetros do OCR para melhor sensibilidade;
- Criação de um bloco de limpeza e normalização textual;
- Treinamento do modelo OCR com amostras específicas do contexto das notas recebidas.

## 8. Progresso do Projeto

Na data atual o **projeto foi concluído**, com as etapas principais de upload, leitura e estruturação funcionalmente implementadas. Os testes parciais realizados indicam um desempenho satisfatório em grande parte dos documentos processados.

No entanto, o sistema ainda requer:

- Ajustes em funções de limpeza e validação de dados;
- Inserção de logs estruturados para auditoria;
- Realização de **testes sistêmicos e integrados**, com diferentes lotes de notas fiscais.

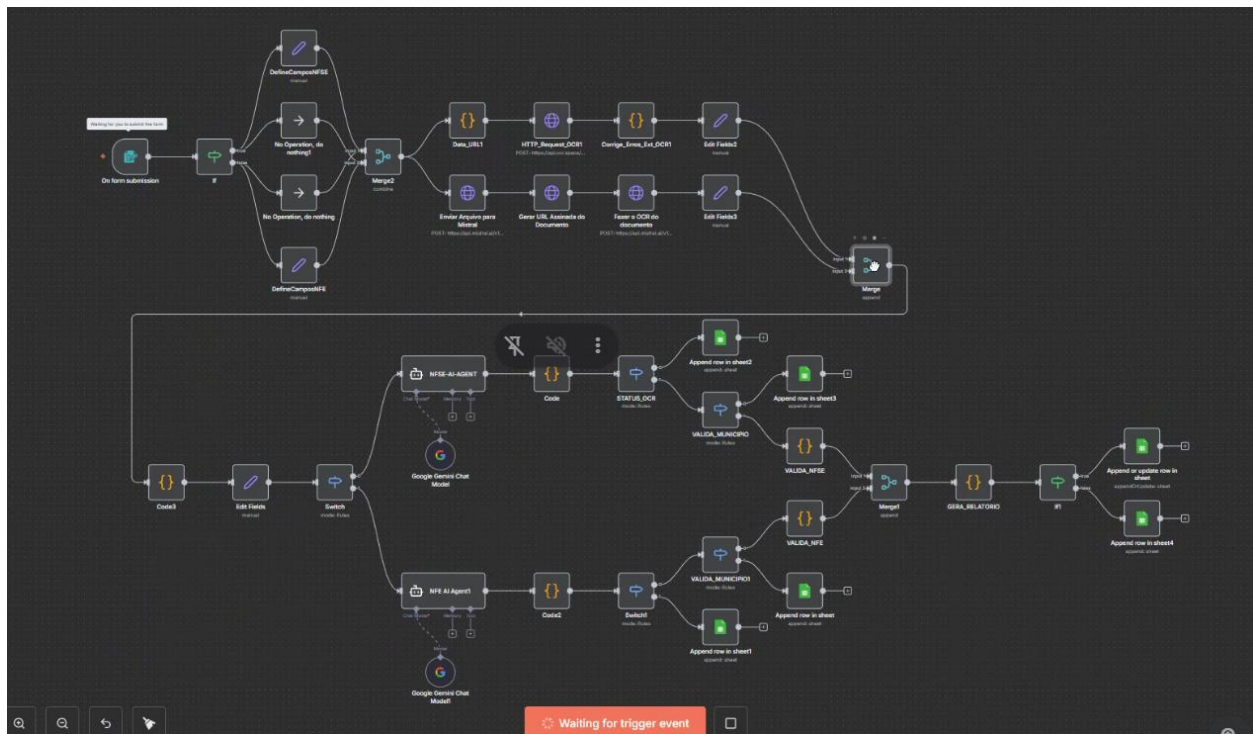
## 9. Considerações Finais

A solução proposta representa um avanço significativo na automação do tratamento de notas fiscais, alinhando inteligência artificial, OCR e automação de processos. Os desafios encontrados fazem parte do processo de refinamento, e as melhorias já em andamento demonstram um compromisso com a entrega de uma solução robusta e confiável.

## 10. Documentação do Fluxo – FiscalTron

### 10.1 Visão Geral do Fluxo:

O workflow começa monitorando duas pastas no Google Drive para novos arquivos PDF ou XML (NFS-e e NF-e). Ele então baixa esses arquivos, executa duas abordagens de OCR (OCR.space para texto bruto e Mistral AI para Markdown estruturado), mescla os resultados, e usa modelos de IA (Google Gemini) para extrair dados estruturados. Finalmente, ele valida os dados extraídos e classifica os arquivos com base no resultado da validação.



### 10.2 Passo a Passo Detalhado:

#### Gatilho Inicial

(Ao clicar em 'Executar workflow' - 6cad2d02-ff7b-487c-8046-0502410886f5):

- Este é um gatilho manual, o que significa que o workflow precisa ser executado manualmente (clcando em "Execute workflow" no n8n) para iniciar o processo.

#### Busca de Arquivos no Google Drive

**(NFS-E - ad7fb7c5-e41c-42e1-a40c-43908394abb1 e NF-E - 1d5e528a-45fe-43db-baf8-0a4533e2226b):**

- O workflow busca por arquivos PDF ou XML (ou TXT) em duas pastas específicas do Google Drive, presumivelmente uma para Notas Fiscais de Serviço Eletrônicas (NFS-e) e outra para Notas Fiscais Eletrônicas (NF-e).
  - NFS-E: Busca na pasta com ID `'1NHeGaPlf2Am6TUuoyzfa_B-BKUramwYT'`
  - NF-E: Busca na pasta com ID `'1ulCdg-87K1TnVwxK4wirkRA6QsY2wiSU'`
- Ele filtra por (, , ) e garante que os arquivos não estejam na lixeira ().`mimeTypeapplication/pdfapplication/xmltext/xmltrashed = false`

### 10.3 Definição de Campos

**(DefineCamposNFSE - 3a90f816-b0ed-4754-8ada-b88b97aef5ae e DefineCamposNFE - 26e21bde-6e44-477c-8d89-96d419751a0a):**

- Após encontrar os arquivos, esses nós adicionam metadados aos itens, como o do arquivo, o (nome do arquivo) e o (definido como "nfse" ou "nfe" dependendo da origem da pasta). Isso é crucial para que o workflow saiba que tipo de nota fiscal está processando.`SetidnametipoDocumento`

### 10.4 Download do PDF

**(DownloadPDF - 24b0aee2-2111-414c-9bed-9b3817dc64c5):**

- Baixa o conteúdo binário dos arquivos PDF (ou outros tipos suportados) encontrados nas etapas anteriores, usando o `do arquivo.id`

### 10.5 Preparação para OCR - Data URL

**(Data\_URL1 - 1cfd41d5-1182-434a-8ecd-d4dc2f429fd5):**

- Este é um nó de código JavaScript que converte o conteúdo binário do arquivo baixado (PDF/imagem) em uma string Base64 com um prefixo de Data URL ().`data:mimeType;base64,...`

- Ele lida com a extração do e garante que os dados binários estejam no formato correto para serem enviados aos serviços de OCR. Ele também inclui logs de depuração para erros.mimeType

## 10.6 OCR 1 - OCR.space

(HTTP\_Request\_OCR1 - 15eff216-069d-4887-bcc8-c0b00994327f):

- Envia a string Base64 do documento para a API do OCR.space.
- Usa o "K85073331588957".apikey
- Define o como "por" (português) e como "2".languageOCREngine
- Este OCR é provavelmente usado para obter um texto bruto do documento ().ParsedText

## 10.7 Correção de Erros de OCR e Extração de Chave

(Corrige\_Erros\_Ext\_OCR1 - 5d52856e-10a6-49e2-af9e-f2039469104f):

- Este nó de código JavaScript processa o texto bruto retornado pelo OCR.space.
- **Objetivo:** Extrair a "chave de acesso" da Nota Fiscal (um número de 44 dígitos), corrigindo erros comuns de OCR como 'S' para 'g', 'O' para 'o', e 'B' para '8'.
- Utiliza expressões regulares para encontrar a chave de acesso, mesmo que precedida por "CHAVE DE ACESSO".
- Armazena a chave limpa no campo .chave\_acesso\_extraida\_limpa

## 10.8 Edição de Campos OCR 1

(Edit\_Fields2 - 7fdac0df-c9ee-4ae0-b0d8-71276ba542df):

- Pega o texto processado pelo e o atribui a um novo campo dentro da estrutura JSON, junto com e (nome do arquivo). Este será o .Corrige\_Erros\_Ext\_OCR1textotipoDocumentonameocr\_1

**OCR 2 - Mistral AI**

(Enviar Arquivo para Mistral - e898865d-5266-4305-8f75-69ec74bffd35 e Gerar URL Assinada do Documento - 24c07762-f47f-4c7e-bdaa-6cc795dfb0de e Fazer o OCR do documento - e5bcbe3c-20b0-4db9-8148-954a918a3528):

- Este é um processo de múltiplos passos para usar a API de OCR da Mistral AI, que é capaz de retornar uma saída mais estruturada (Markdown).
- Enviar Arquivo para Mistral: Envia o arquivo binário para a API de arquivos da Mistral com o propósito "ocr".
- Gerar URL Assinada do Documento: Solicita uma URL assinada para o arquivo recém-enviado, permitindo que a API de OCR da Mistral acesse o documento.
- Fazer o OCR do documento: Envia uma solicitação de OCR para a API da Mistral AI, especificando o modelo "mistral-ocr-latest" e a URL do documento. Pede para incluir a imagem em base64. O resultado esperado é um markdown estruturado nas `.pages[0].markdown`

## 10.9 Edição de Campos OCR 2

(Edit Fields3 - baf1c007-315f-4648-9174-cfa73d2f8cb1):

- Pega o resultado do OCR da Mistral (espera-se que seja um markdown) e o atribui a um novo campo dentro da estrutura JSON, junto com e . Este será o `.textotipoDocumentonameocr_2`

## 10.10 Fusão dos Resultados de OCR

(Merge - 51bc454a-645b-4480-8b41-d4168cc552eb):

- Combina os resultados dos dois caminhos de OCR ( e ).Edit Fields2Edit Fields3

## 10.11 Combinação de OCRs

(Code3 - cdadd7b4-0605-48a5-8006-2f5d518d754e):

- Este nó de código JavaScript pega os dois resultados de OCR (OCR.space e Mistral AI) e os combina em um único item, onde o `o` é o texto bruto do OCR.space e o `o` é o markdown estruturado da Mistral AI.  
`Al.ocr_1ocr_2`

## 10.12 Atribuição Final de Campos

(Edit Fields - e1e7c4a1-0e92-4af8-8ced-860768559aa9):

- Este nó organiza os dados para a próxima etapa, atribuindo os valores , , e corretamente ao item para serem usados pelos agentes de IA.ocr\_1ocr\_2tipoDocumentoname

## 10.13 Seleção do Agente de IA

(Switch - ff3d820d-54d8-4a77-88b6-adf0e7c65269):

- Este nó direciona o fluxo com base no (se é NFS-e ou NF-e), para o agente de IA apropriado.tipoDocumento
  - Se for "nfse", ele vai para .tipoDocumentoNFSE-AI-AGENT
  - Se for "nfe", ele vai para .tipoDocumentoNFE AI Agent1

## 10.14 Agentes de Extração de IA (Google Gemini):

- **NFS-e (NFSE-AI-AGENT - 24960253-f276-4277-b7b5-83a75520017c):**
  - Usa o modelo Google Gemini (especificamente ).models/gemini-2.5-flash-preview-05-20
  - Recebe um prompt detalhado que instrui o modelo a atuar como um "especialista em extração de dados JSON de notas fiscais eletrônicas (NFS-e)".
  - As instruções incluem:
    - **Estratégia de Análise Dual:** Usar (pré-processado) e (bruto original).ocr\_1ocr\_2
    - **Protocolo de Validação Cruzada para CNPJ/CPF:** Regras rigorosas para extrair, comparar e validar CNPJ/CPF de ambos os OCRs, priorizando consistência e formato válido. Inclui regra para nunca usar o mesmo CNPJ/CPF para emitente e destinatário.
    - **Regras Detalhadas para Campos Gerais:** Usar como principal para número da nota, data/hora de emissão, código de verificação, nome/razão social, inscrição municipal, endereço e valores financeiros.ocr\_1
    - **Código do Serviço:** Procurar linha com código numérico, usar a versão mais completa.

- **Protocolo de CEP:** Regras específicas para CEP com/sem hífen.
- **Validação de ISS:** Calcular e comparar com (com tolerância de 0.01). Se divergente, recalcular a alíquota.  $(base\_calculo * aliquota\_iss) / 100$  valor\_iss
- **Estrutura JSON de Saída:** O modelo é instruído a gerar um JSON com uma estrutura predefinida (tipo de documento, identificação, emitente, destinatário, serviço, valores e status de processamento).
- O é muito robusto, com exemplos de raciocínio interno.systemMessage

## 10.15 NF-e

(NFE AI Agent1 - caafa8a2-7f9c-49bc-a61e-ef1e9583446a):

- Também usa o modelo Google Gemini ().models/gemini-2.5-flash-preview-05-20
- O prompt para NF-e é igualmente detalhado e específico para as características de uma NF-e.
- Instruções incluem:
  - **Regra Fundamental:** deve ser copiado diretamente dos metadados.nome\_arquivo
  - **Regra Geral:** Usar o Markdown (Fonte 2 - ) como base, com o Texto Bruto (Fonte 1 - ) para verificação e preenchimento de lacunas.ocr\_2ocr\_1
  - **Exceção Crítica:** Lógica especial para corrigir a tabela que pode estar desalinhada.CÁLCULO DO IMPOSTO
  - **Impostos PIS e COFINS:** Ignorar estrutura da tabela e localizar rótulos específicos (, ). Preencher com "0.00" se ausente.VALOR DO PISVALOR DA COFINS
  - **Chave de Acesso:** Extrair sequência numérica, remover espaços, 44 dígitos.
  - **Data e Hora de Emissão:** Extrair e formatar (DD/MM/AAAA HH:MM:SS).

- **Endereços:** Preferir Markdown, mas usar Texto Bruto se incorreto/abreviado, e normalizar abreviações.
- **Descrição dos Produtos:** Obrigatória, usar Texto Bruto se ausente/ilegível no Markdown.
- **Código NCM:** Extrair para cada produto, corrigindo desalinhamentos com o Texto Bruto. Incluir campos como `.produto_0_ncm`
- **Valores Numéricos:** Converter vírgulas para pontos, remover espaços, usar formato string, preencher com "0.00" se ausente.
- **Demais Campos:** Priorizar Markdown, usar Texto Bruto para verificação.
- **Formatação:** Saída apenas do JSON puro e válido.

## 10.16 Processamento da Saída do Agente

(Code - `3c165762-512c-4322-b6ff-2c2c995de66b` para NFS-e e Code2 - `58d5aff1-2c71-47c3-a14c-143dec18bce6` para NF-e):

- Estes nós de código Python recebem a saída JSON (como string formatada com markdown, ex: `"`) dos agentes de `IA.json\n[...]\n`
- Eles limpam a string, removendo o `,` e então parseiam a string JSON para um objeto `JSON.json`` e
- Cada objeto JSON de documento fiscal é então preparado para ser um item separado no `n8n`.
- O (para NF-e) é ligeiramente mais robusto no tratamento de erros, adicionando `e` ao item de `saída.Code2status_processamentomensagem_erro`

## 10.17 Verificação do Status de OCR

(STATUS\_OCR - `08e3b385-fa1a-4e47-b167-c228f74e7db9` para NFS-e e STATUS\_OCR1 - `95e354fb-18e2-4516-89ef-0a2c55c5e763` para NF-e):

- Verifica o que foi adicionado pelo nó ou `.status_processamentoCodeCode2`

- Se o status for "sucesso" (ou "SUCESSO" para NF-e), continua para a validação do município.
- Se não for sucesso (houve um erro no OCR ou parsing JSON), o arquivo é encaminhado para o nó .PEGA\_ID\_ARQUIVO1

## 10.18 Validação do Município

(VALIDA\_MUNICIPIO - 62c64b9c-67af-4c3d-9c9a-bcf63a6cde8e para NFS-e e

VALIDA\_MUNICIPIO1 - a13d464c-f840-4426-b3a2-49eea94a9b6a para NF-e):

- Verifica se o do (extraído pelo agente de IA) é "São Paulo".municipioemittente
- Se for "São Paulo", o fluxo prossegue para a validação mais detalhada da nota.
- Se não for "São Paulo", o arquivo é encaminhado para o nó (para movimentação de "Outro Município").PEGA\_ID\_ARQUIVO

## 10.19 Validação Detalhada da Nota Fiscal

(VALIDA\_NFSE - 0160521d-5e88-4da1-8bf0-56c9c8e7d64f para NFS-e e VALIDA\_NFE - 9be42d29-39c7-490e-b328-c7e99accebe5 para NF-e):

- Estes são nós de código Python com funções robustas de validação.
- **NFS-e (VALIDA\_NFSE):**
  - Valida CNPJ/CPF do emitente.
  - Valida o formato da data de emissão.
  - Valida o código de verificação.
  - Valida o número da nota (somente dígitos, máx 15).
  - Realiza a validação do cálculo do ISS ( versus , com tolerância de 0.05).base\_calculo \* aliquota\_iss / 100valor\_iss
  - Retorna um objeto com (booleano), (um dicionário de erros encontrados) e .validoerrosdados\_originais

## 10.20 NF-e (VALIDA\_NFE):

- Similarmente, valida CNPJ do emitente, CPF/CNPJ do destinatário.
- Valida número da nota, série, data/hora de emissão.
- Valida a chave de acesso (44 dígitos e dígito verificador).
- Valida o .indicador\_presenca
- Valida a coerência do em relação à soma dos produtos menos o desconto.valor\_total\_nota
- Verifica a base de cálculo de PIS/COFINS se houver valor de PIS/COFINS.
- Valida os (valor e alíquota).tributos\_aproximados
- Valida a presença de produtos, descrição, valor unitário, quantidade e NCM para cada produto.
- Retorna um objeto com , e .validoerrosdados\_originais

## 10.21 Fusão dos Resultados de Validação

(Merge1 - a05c3d47-217e-442e-ba7f-a24c9d1ae672):

- Combina os resultados das validações de NFS-e e NF-e.

## 10.22 Geração de Relatório

(GERA\_RELATORIO - 2679c0c3-0269-41a2-8cbf-7b4d1c231339):

- Este nó de código JavaScript pega o resultado da validação (se válido ou com erros) e formata um objeto de relatório plano.
- Adiciona e .data\_validacaohora\_validacao
- Compacta os em uma única string .erroserro
- Extrai e organiza campos importantes como , , , nomes de emitente/destinatário, valores financeiros, etc.tipo\_documento nome\_arquivo numero\_nota
- Substitui valores vazios ou "[empty]" por "-".
- A saída é um registro de relatório para cada nota fiscal processada.

## 10.23 Decisão Final de Arquivamento

(Switch2 - 5518cc75-9f32-4330-88e4-cb23abf16418):

- Verifica o campo do relatório gerado.valido
- Se for , o arquivo é encaminhado para .validotruePROCESSADO\_OK
- Se for (houve erros de validação), o arquivo é encaminhado para .validofalsePROCESSADO\_ERRO

**Movimentação e Arquivamento no Google Drive (Nós desabilitados):**

- **PEGA\_ID\_ARQUIVO - bfa0030c-a32e-4f58-b1d5-1e3a7d1c6b22 & PEGA\_ID\_ARQUIVO1 - 861de5e4-c23a-483b-8954-b22ff6e98167:** Estes nós parecem ser usados para buscar o ID do arquivo (que pode ter sido perdido no fluxo ou se o workflow não passou o ID em todos os lugares) antes de movê-lo. Eles buscam o arquivo pelo .nome\_arquivo
- **MOVE\_OUTRO\_MUNICIPIO - 48762ae3-f89a-432d-b3ba-1c3fce32c97f (Desabilitado):** Moveria arquivos para a pasta se o município não fosse São Paulo.'1XX-ZFGLGA\_noXyGnmSswtqJVTE2-KQ5g'
- **PROBLEMA\_OCR - 8a8551f0-fbe8-4214-a793-dca5bbef570b (Desabilitado):** Moveria arquivos para a pasta se houvesse problemas no OCR inicial ou no parsing JSON.'17QgExkli-5dslnX-Ev5qFNtybgilDZqL'
- **PROCESSADO\_OK - 482f914d-2786-4510-9cac-7f3bfa9be4b7 & MOVE\_PROCESSADOS - f30e130f-177e-4577-b5b2-ed83181d7a46 (Desabilitado):** Moveria arquivos processados e validados com sucesso para a pasta .'1LpIlbzkvLjw6owcZqCNlnuWNmMOWvFoo'
- **PROCESSADO\_ERRO - aac635a4-e761-489a-be14-2f69e447f7be & MOVE\_ERROS - obd813df-8be1-407d-bc47-6c5df74f2a6b (Desabilitado):** Moveria arquivos com erros de validação para a pasta .'1l6LXTI6ltXw6PS-uSjkAI5NyHKWIk1Qt'

## 10.24 Em Resumo:

O workflow automatiza a extração e validação de dados de notas fiscais (NFS-e e NF-e) em PDF. Ele emprega uma abordagem de OCR duplo (OCR.space para texto bruto e Mistral AI para markdown estruturado) para maximizar a precisão. Em seguida, utiliza agentes de IA do Google Gemini com prompts complexos e regras de

validação cruzada para extrair os dados. Por fim, valida a integridade dos dados extraídos e, idealmente, moveria os arquivos para pastas específicas no Google Drive com base no sucesso ou falha do processamento (as etapas de movimentação estão atualmente desabilitadas).

