Projeto Final
Curso Agentes Inteligentes I2A2
FISCALIA

Índice

1 Referencial

1.1 Projeto

O delineamento geral e recolha de informação decorreu de maio a outubro/2025.

O planejamento final do desenvolvimento incluindo especificação iniciou-se em 06-10-2025.

O desenvolvimento com assistência Claude iniciou-se em 13-10-2025 e uma versão do tipo MVP ficou pronta em 27-10-2025.

A data de entrega do trabalho totalmente concluído incluindo relatório, apresentação, e vídeo é 30-10-2025.

1.2 Estratégia recomendada

- Thread de planejamento - Discutir arquitetura, decidir abordagem
- Thread de implementação - Criar estrutura de pastas e arquivos, desenvolver código, resolver bugs específicos
- Thread de deploy - Configurar Railway, resolver problemas de produção
- Usar busca de contexto quando precisar retomar

1.3 Ferramenta Framework ou no-code recomendável

Ao longo de alguns meses foi feita a avaliação de diversas frameworks e ferramentas no-code para determinar possibilidades adequadas ao desenvolvimento, em diversas vertentes, nomeadamente baseado em agentes inteligentes, facilidade de desenvolvimento, custo e atualidade.

Foi realizada também pesquisa com Claude, obtendo-se os seguintes resultados:

- https://claude.ai/share/f4951aec-2db6-4855-94b4-353fdc477253

Avançaremos com CrewAI e Banco de Dados SQLite para uso local com Streamlit e para uso remoto com Streamlit e em Railway. Preferencialmente PostgreSQL deverá ser também considerado, para garantia total de persistência de dados mesmo entre novos arranques do sistema.

2 Especificação

O desenvolvimento é realizado com assistência Claude ao longo do projeto, de acordo com as linhas gerais da especificação e requisitos descritos na presente secção.

Desenvolvimento do sistema Fiscalia baseado em Agentes Inteligentes.

2.1 Enquadramento geral

O projeto, resultando num sistema a que se atribuiu a designação "Fiscalia", tem desenvolvimento integral da arquitetura e codificação Python baseado em Agentes Inteligentes para a implementação.

Com know-how em desenvolvimento de programação Python, VSC e GitHub, assume-se como crucial a ajuda detalhada passo a passo, para desenvolvimento de cada módulo, incluindo a respetiva codificação.

É um processo interativo: desenvolvimento => testes =>correções => desenvolvimento.

Utiliza-se **CrewAI** por ser um framework excelente pela sua boa conceção e organização do tipo Crew-Agents-Tasks-Tools.

A solução baseia-se numa arquitetura e codificação Python o mais simples possível, assentando o mais possível em LLM e menos em tools específicas.

Depois de delineada a arquitetura geral Crew-Agents-Tasks-Tools, a implementação Python avança com base em **VSC**, com instruções detalhadas e codificação realizada módulo a módulo, seguindo-se testes para as devidas afinações passo a passo.

Em paralelo a aplicação **Streamlit** vai sendo desenvolvida para criar a interface gráfica para usuário, que vai servir também para testes em modo localhost e posteriormente online em Railway (aplicação Streamlit em modo híbrido).

Finalmente implementa-se o acesso público através de **Railway** com ajuda de Docker e GitHub.

Utiliza-se a LLM **OpenAI** (https://platform.openai.com), mas preparando tudo para poder usar alternativamente a LLM **Groq** (https://groq.com), gratuita, por simples configuração.

Variáveis: OPENAI_API_KEY e OPENAI_MODEL (gpt-4o-mini) + GROQ_API_KEY e GROQ_MODEL (groq/llama-3.3-70b-versatile).

Em resumo a abordagem preconizada é:

1. Definir Arquitetura e estrutura de pastas e arquivos
2. Desenvolvimento com VSC, incluindo módulo Streamlit para testar em modo localhost, e arquivo Docker para permitir usar Railway no final
3. Usar Railway com GitHub para acesso público e teste
4. Relatório Técnico do Projeto

A seguir descreve-se os requisitos gerais do sistema pretendido, mas iremos focar consideravelmente, simplificando para obter um **MVP** com relevância numa fase inicial.

2.2 Objetivo (Requisitos)

- Automatizar o processamento e análise de documentos fiscais em uso no Brasil
- Podem ser documentos físicos ou eletrônicos (ex.: XML de NFe/NFCe/CTe/MDF-e)
- Foco em otimização/aprimoramento:
  - Redução de erros manuais na escrituração;
  - Otimização de tempo no fechamento contábil e fiscal;
  - Detecção de inconsistências fiscais (valores, CFOP, CST, NCM etc.);
  - Integração com ERPsde mercado e sistemas contábeis (Domínio, Alterdata, Protheus, etc.).

Em seguida identificam-se os temas que consideramos na implementação inicial.

1-Extração de dados: limitar aos arquivos XML NFe. Deixar tudo preparado para uma futura versão com inclusão de outras tipologias e também de arquivos de notas fiscais em PDF e em JPEG/PNG para o caso de notas fiscais físicas em papel. Os arquivos serão inseridos manualmente em aplicação Streamlit, depositados individualmente ou em grupo ou agregados em ZIP para processamento em batch.

2-Validação e Auditoria: utilizar especificações das notas fiscais das entidades relacionadas do Brasil. Deverá também detectar eventuais arquivos de notas fiscais duplicados.

3-Classificação, Categorização e Customização por ramo de atividade: classificar compra, venda, serviços.

4-Automação de Processos Fiscais/Contábeis: lançamentos contábeis.

5-Ferramentas Gerenciais: relatórios personalizados

A **simplificação inicial** para **MVP** é, resumidamente e em geral, considerar que trataremos de arquivos XML de NFe inseridos individualmente ou em grupo ou agregados em ZIP. Não trataremos de outros formatos nem se avançará com lançamentos contábeis, isto é, com a implementação de interface com ERPs. O sistema é concebido para acomodar na sua arquitetura o tratamento de outros formatos futuramente.

Os temas acima identificados são em seguida descritos textualmente de forma mais detalhada.

1-Extração de Dados:

- Recuperar documentos fiscais em fontes conhecidas
- Utilizar OCR (Reconhecimento Óptico de Caracteres) em conjunto com NLP (Processamento de Linguagem Natural) para extrair dados relevantes dos documentos:
  - Informações do emitente e destinatário
  - Itens da nota (descrição, quantidade, valor)
  - Impostos (ICMS, IPI, PIS, COFINS)
  - CFOP, CST e outros códigos fiscais
- Desafios**:**
  - Como tornar o agente capaz de se adaptar a diferentes layouts e formatos de documentos.
  - Como se adaptar às mudanças legais (ex. IVA)

2-Validação e Auditoria:

- Agentes que verificam a consistência dos dados, comparando-os com regras fiscais e cadastros de clientes/fornecedores.
- Identificar e sugerir correção para erros comuns, como:
  - Cálculo incorreto de impostos
  - Códigos fiscais inconsistentes
  - Divergências entre pedido de compra e nota fiscal
- Produzir relatórios de auditoria, destacando possíveis problemas e áreas de risco e enviando-os aos responsáveis (que podem ser outros agentes)
- Desafios:
  - Identificar maiores agressores e sugerir melhorias
  - Adaptar às mudanças legais ou do ambiente de negócios

3-Classificação e Categorização e Customização por ramo de atividade:

- Classificar automaticamente os documentos fiscais por tipo (compra, venda, serviço) e por centros de custos.
- Organizar e o arquivar corretamente os documentos.
- Realizar ações customização por ramo de atividade. Ex.:
  - Agronegócio: Monitoramento de CFOPs específicos do setor (venda de produtos agrícolas, insumos), cálculo de impostos com particularidades do agronegócio.
  - Setor Automotivo: Validação de notas fiscais de peças e serviços automotivos, conferência de códigos de peças e compatibilidade com as atividades da empresa.
  - Indústria: Apuração de impostos específicos da indústria (IPI, Substituição Tributária, etc.), Geração de insumos para cálculo de custos de produção
- Desafios:
  - Adaptar às mudanças legais ou do ambiente de negócios
  - Como tratar ramos de atividade específicos –órgãos públicos, terceiro setor, etc.

4-Automação de Processos Fiscais/Contábeis:

- Lançamentos Contábeis:
  - Os agentes geram automaticamente os lançamentos contábeis a partir dos dados obtidos nos documentos fiscais.
- Apuração de Impostos:
  - Os agentes calculam os impostos a pagar e a recuperar, gerando guias de recolhimento.
  - Automação a entrega de obrigações acessórias (SPED Fiscal, EFD Contribuições).
- Conciliação Bancária:
  - Cruzamento dos dados das notas fiscais com os extratos bancários, facilitando a conciliação.
  - Identificação pagamentos e recebimentos pendentes.
- Desafios:
  - Manter os agentes atualizados em relação a critérios contábeis e obrigações acessórias?
  - Como se beneficiar do Open Banking?
  - Como garantir a segurança dos processos?

5-Ferramentas Gerenciais:

- Relatórios Personalizados:
  - Geração de relatórios personalizados, com informações relevantes para o seu setor.
    - Utilizar informações internas
    - Agregar informações externas relevantes para a empresa
- Análises preditivas e simulações de cenários.
- Assistente Consultor Especializado:
  - Suporte para dúvidas e decisões estratégicas.
  - Informações sobre contabilidade e tributação.
- Desafios:
  - Como garantir a qualidade das informações apresentadas?
  - Como maximizar a experiência do usuário

2.3 Descrição do ponto de vista do usuário

O usuário terá disponível as seguintes funcionalidades oferecidas pelo sistema:

1)-Selecionar uma opção, e só uma de cada vez, entre as seguintes possibilidades:

a) Adicionar, por upload ou arrastando, um ou mais **arquivos de Nota Fiscal individual** (XML, PDF, PDF imagem/OCR, imagem PNG/JPEG) ou arquivos individuais agregados em ZIP. O formato XML para Notas Fiscais é padronizado no Brasil pela respetiva legislação.

OU (futuramente, fora do âmbito do MVP)

b) **Carregamento, por upload ou arrastando, de dois arquivos CSV inter-relacionados**, designadamente: Lista enumerando diversas Notas Fiscais e Lista de Items das respetivas Notas Fiscais enumeradas. Os dois arquivos estão inter-relacionados pela coluna "NÚMERO" por default. O usuário poderá eventualmente alterar na aplicação o nome da coluna de inter-relacionamento.

2)-Comando para iniciar pré-processamento da opção selecionada, que consiste em registar em Banco de Dados as Notas Fiscais individuais. No caso dos dois arquivos CSV, o tratamento será unificar num único CSV para depois registar em BD.

3)-Visualizar resultado do comando efetuado (sucesso ou insucesso).

4)-Comando para visualizar em BD (banco de dados) registos de processamento. Estes registos de processamento servem para obtenção de estatísticas diversas e como base para posterior processamento em ERP.

5)-Comando para visualizar em BD (banco de dados)

6)-Comando para obter estatísticas pré-definidas (quantidades, valores, impostos processados de Notas Fiscais, de Recibos, etc).

7)-Comando para fazer perguntas livres sobre o processamento e registos em BD, com alguns exemplos. Os registos em BD podem ser faturas de despesas ou recibos. Deverão conter a indicação se foram processados já em ERP ou ainda não. Juntar também conjunto de perguntas tipo que o usuário pode selecionar.

2.4 Descrição do Sistema

O sistema terá uma Banco de Dados "bd_fiscalia" descrita em secção própria mais abaixo.

O sistema processa cada arquivo de Nota Fiscal individual da pasta /entrados distribuindo ao agente especializado (agentes especializados: XML, PDF, PDF imagem/OCR, imagem PNG/JPEG). Nesta implementação de uma versão inicial apenas serão processados os arquivos XML. Ou seja, os restantes agentes ou tasks especializados em formatos não XML retornarão erro do tipo formato ainda não suportado.

Se o arquivo não for de um dos tipos indicados então faz registo da anomalia em BD.

Em cada agente especializado, se o arquivo for de um dos tipos indicados, mas o conteúdo não for decifrável, então faz registo em tabela "registo_resultados" da BD "bd_fiscalia".

Os registos de sucesso ou insucesso de tratamento dos arquivos são em tabela.

Se o arquivo for aceitável então tenta registar em tabela "docs_para_ERP" da Banco de Dados. Se o conteúdo já existe em Banco de Dados, então faz registo da anomalia em BD.

Os registos válidos em BD servem para posterior processamento em interface com ERP (fora do âmbito da versão inicial).

A tabela "docs_para_ERP" da BD "bd_fiscalia" deve ter os campos para registo de acordo com a pesquisa efetuada anteriormente, nomeadamente:

- https://claude.ai/chat/615ced09-baa4-4839-ab85-1b18b9fc6c23

- https://claude.ai/chat/0959af70-984d-4f7e-8089-1bad43b51b2a

ou em:

- https://www.perplexity.ai/search/em-anexo-envio-um-arquivo-com-qMIQVPrcRZKsB.3jFmwq9w

e em especial **Tipos de Notas Fiscais no Brasil**:

https://www.nfe.fazenda.gov.br/portal/principal.aspx

https://www.contabilizei.com.br/contabilidade-online/tipos-de-notas-fiscais/

Para uma fase posterior (Parte 2) ainda sem decisão para avançar:

Um agente coordenador ERP vai verificando se existe nova transação registada em BD e agentes especializados executam a transação ERP.

2.5 Ambiente de Desenvolvimento e Componentes de Software Base

O **ambiente de desenvolvimento** do projeto é:

Windows 11 | VSC (Virtual Studio Code) | Python 3.13.5 | GitHub

Utilizaremos venv (virtual environment). O folder venv deve ser configurado para inicializar automaticamente o ambiente virtual sempre que o terminal VSC é iniciado.

Serão também utilizados os seguintes **softwares base**:

CrewAI 0.140.0 | Streamlit 1.46.1 | Docker | Railway

O **Streamlit deverá ter utilização híbrida** para permitir teste local e deploy para teste online público em Railway.

O repositório **GiHub** é: https://github.com/josefeneto/fiscalia

Acesso à aplicação **Streamlit**: https://fiscalia.up.railway.app

A **estrutura de pastas e arquivos** do projeto é similar à do projeto EDA (Exploratory Data Analysis) desenvolvido em setembro/outubro 2025, naturalmente com as adequadas adaptações.

Pasta em Windows criada para o projeto para uso em VSC:

C:\Users\josef\My_Documents_Projects\fiscalia

Pasta de arquivos de notas fiscais para testes:

C:\Users\josef\My_Documents_Projects\fiscalia\arquivos_teste_NF

2.6 Banco de Dados

Será usada **Banco de Dados SQLite** disponível diretamente em Python (sqlite3, versão SQLite version 3.50). Será usada por CrewAI e Streamlit em modo localhost para testes e em modo online com Streamlit em Railway. Deverá garantir persistência de dados, tabelas e conteúdos, entre sessões localmente e online em Railway.

Deverá ser criada e configurada essa Banco de Dados SQLite com nome "bd_fiscalia" com arquivo respetivo "bd_fiscalia.db".

Tabela "registo_resultados":

O resultado do processamento dos arquivos será registado em tabela registo_de_resultados, com colunas numero_sequencial, time_stamp, path_nome_arquivo, resultado (exemplo: Sucesso, Insucesso) e Causa (exemplo: arquivo não suportado, arquivo duplicado, etc).

Tabela "docs_para_ERP":

No caso de processamento bem-sucedido, será feito o registo completo dos dados de cada Nota Fiscal em tabela "docs_para_ERP". Este registo deverá ser de acordo com os atributos conforme requisitos de dados de Notas Fiscais conforme legislação brasileira, servindo posteriormente para registo transacional num ERP. Para além do conteúdo padronizado de cada Nota Fiscal, deverá ter em cada linha de registo também numero_sequencial, time_stamp, path_nome_arquivo, erp_processado (Yes/No).

Persistência no Railway:

Para garantir persistência no Railway, deveremos adicionar, se necessário, em "Variables" o atributo "RAILWAY_VOLUME_MOUNT_PATH=/data". O SQLite vai salvar em /data/bd_fiscalia.db

SQLiteStudio:

Para gerir a Banco de Dados, nomeadamente para visualizar registos, a aplicação SQLiteStudio (https://www.sqlitestudio.pl) poderá ser utilizada, se for considerado conveniente. Embora não parece viável a utilização direta em ambiente Railway,

Preferencialmente, deverá ser possível também utilizar PostgreSQL alternativamente ao SQLite com adaptação ligeira. O PostgreSQL é suportado automaticamente no Railway.

Relativamente à **persistência SQLite em Railway** verifica-se que:

a) Em cada **Deploy** completo ou **Redeploy** o banco de dados *bd_fiscalia.bd* é recriado e perde-se consequentemente a informação contida anteriormente.
b) Em cada **Restart** o banco de dados *bd_fiscalia.bd* mantem-se e a informação registada anteriormente é retida e persiste.

A utilização de PostreSQL é crucial para ambientes de produção pois garante persistência dos dados e viabiliza gestão própria de forma mais direta.

3 Informação complementar

Tipos de Notas Fiscais no Brasil

https://www.contabilizei.com.br/contabilidade-online/tipos-de-notas-fiscais/

https://www.nfe.fazenda.gov.br/portal/principal.aspx

Soluções UiPath

https://www.uipath.com/solutions/department/finance-and-accounting-automation

https://www.mirante.net.br/entenda-de-forma-rapida-e-simples-o-que-e-uipath/

ERPs Brasil

https://www.totvs.com/

https://www.sap.com/brazil/index.html

https://www.senior.com.br/

https://www.bling.com.br/

4 Análise prévia dos arquivos de notas fiscais

Com base em pesquisas e assistência Claude, analisamos detalhadamente o conteúdo das Notas Fiscais de dois arquivos CSV (**202401_NFs_Cabecalho.csv e 202401_NFs_Itens.csv**), que estão interrelacionados pela coluna "NÚMERO".

Preparamos uma lista com um conjunto de perguntas que podemos fazer sobre essas notas fiscais (exemplo: quantidade de notas fiscais, quantidade de itens, valor, por destinatário, por estado, por cidade, etc). Médias e Totais.

Finalmente prepara-se um resumo geral classificando e sumarizando as notas fiscais nos diversos aspetos relevantes, incluindo respetivas estatísticas relevantes.

O referencial da especificação de formatos é a legislação brasileira.

Anteriormente já recolhemos alguma informação sobre essa legislação, nomeadamente:

- https://claude.ai/share/a3514645-9b14-4809-a8b0-8f563d489255
- https://claude.ai/share/2d6482c5-c20c-4d06-b98c-9d251fc02d6f

Resultado obtido: https://claude.ai/share/961f475e-9f4d-4bd4-bf05-3bd23af8856b

Seguidamente, tentámos fazer análise idêntica para o outro par de arquivos (**202505_NFs_Cabecalho.csv e 202505_NFs_Itens.csv**).

Mas: no Claude não é possível carregar arquivos maiores que 31 MB e no Perplexity também não é possível carregar arquivos com mais de 50 MB.

No ChatGPT foi possível obter uma resposta parcial, pois embora carregasse ambos os arquivos, não conseguiu processar o arquivo dos items com 81 MB.

Resultado obtido: https://chatgpt.com/share/68eb81e5-1ba4-8013-8c00-4d0505ecc117

No projeto Fiscalia, em evolução posterior, utilizaremos apenas os arquivos 202505, eventualmente de forma parcial para evitar os constrangimentos relativos ao tamanho.

5 Sugestões de Perguntas

As perguntas seguintes cobrem verificação, agregação, consistência e análises fiscais/comerciais. Nas perguntas pré-definidas o usuário deve selecionar previamente o período a que se refere (data de início e data de fim) e se não preencher este campo o default é o ano corrente.

Contagem / dimensão

1. Quantas notas fiscais (registos na folha de cabeçalho)?
2. Quantos registos de itens no ficheiro de itens?
3. Quantas notas aparecem no cabeçalho mas não têm itens associados e vice-versa?
4. Quantos itens médios por nota (média/mediana/desvio)?

Valores e totais

5. Total do valor bruto (campo valor/vNF) de todas as notas.
6. Total dos descontos (campo vDesc) e percentagem média de desconto por nota.
7. Total de impostos destacados nas NF (ICMS, IPI, PIS, COFINS, ISS se houver) — por imposto e total.
8. Média e mediana do valor por nota; máximo, mínimo, Q1/Q3.

Por destinatário (cliente)

9. Total de notas por destinatário (CNPJ/CPF), total faturado por destinatário, média por nota.
10. Top 10 destinatários por valor faturado e por número de notas.

Geografia (Origem/Destino)

11. Totais por UF do destinatário (estado), número de notas, valor total.
12. Totais por município do destinatário (top 20 cidades).
13. Distribuição por CFOP e por tipo de operação (saída/entrada).

Itens

14. Quantidade/valor agregado por código de item (cProd), descrição do produto.
15. Top itens por quantidade e por valor.
16. Verificar se soma dos valores dos itens por nota = valor total da nota (vProd sum(items) vs vNF). Lista de notas com mismatch acima de X% (ex.: 0.5%).

Tributação / Contabilização

17. Distribuição por CST/CSOSN (ICMS), alíquotas mais comuns e média de alíquotas por grupo.
18. Valores e percentuais de base de cálculo vs imposto efetivo por nota.
19. Verificar presença e preenchimento do campo "valor aproximado dos tributos" (vTotTrib) quando exigido.

Integridade e validação

20. Há números de nota duplicados para o mesmo série/emissor?
21. Validar formato e consistência da **Chave de Acesso** (44 dígitos, composição cUF + AAMM + CNPJ + mod + serie + nNF + tpEmis + cNF + cDV).
22. Validação de CNPJ / CPF (tam, dígitos, check digit básico).

23. Notas com CFOP incomum ou CFOP que não respeita operação (ex.: CFOP de Exportação sem dados de exportação).

Tempo / série

24. Distribuição por data de emissão (diária, semanal, mensal) — tendência ao longo do mês.
25. Séries (série da nota) mais utilizadas e totais por série.

CQ (Controle de qualidade)

26. Notas com campos vazios obrigatórios segundo leiaute NF-e (emitente, destinatário, CNPJ/CPF, modalidade, itens sem descrição, valores/vazios).
27. Notas com diferença entre valor total e soma dos itens > X (flagged).
28. Notas emitidas com CFOP de remessa/retorno/bonificação e respectivos valores.

Estrutura Resumida
Pastas e Arquivos do Projeto

```
fiscalia/
├─── streamlit_app/          # Interface Streamlit
│    ├─── app.py             # Aplicação principal
│    ├─── pages/             # Páginas multi-page
│    └─── components/        # Componentes reutilizáveis
├─── src/                    # Código fonte
│    ├─── agents/            # Agentes CrewAI
│    ├─── crews/             # Crews e tarefas
│    ├─── models/            # Modelos de dados
│    ├─── services/          # Serviços de negócio
│    └─── utils/             # Utilidades
├─── data/                   # Base de dados e temp
├─── tests/                  # Testes automatizados
├─── requirements.txt        # Dependências Python
├─── Dockerfile              # Container Docker
├─── railway.toml            # Config Railway
├─── .env.example            # Template variáveis
└─── README.md               # Guia de utilizador
```