

# RELATÓRIO TÉCNICO

## Agente Autônomo de Análise Fiscal

*Sistema Inteligente para Processamento de Notas Fiscais*

**Projeto I2A2 - Grupo 4**

Brasil, 03 de Novembro de 2025

# 1. IDENTIFICAÇÃO DO PROJETO

## 1.1. Nome do Grupo

I2A2 Grupo 4

## 1.2. Integrantes do Grupo

- Monica Nascimento - monica.so.nascimento@gmail.com
- Andre Luiz Costa - ccosta.andreluiz@gmail.com
- Renata Capri Raposo - rraposo@yahoo.com.br
- Fábio Moreira Vilas Bôas - villasboas1980@gmail.com
- Carlos Rosa Júnior - krlusrosajr@gmail.com
- Levinia Capasso - assistentevirtualcapasso@gmail.com
- Luan Mateus Dantas de Oliveira - luanmateus.ro@gmail.com
- Vanessa Assis - nessa.pub@gmail.com
- Douglas Santos - douglas2015sf@gmail.com
- Suzane Braga - oisuzanebraga@gmail.com

# 2. DESCRIÇÃO DO TEMA ESCOLHIDO

O projeto desenvolvido consiste em um **Agente Autônomo de Análise Fiscal**, um sistema baseado em Inteligência Artificial que automatiza o processamento, classificação e análise de documentos fiscais (notas fiscais eletrônicas - NF-e) em formato CSV.

O sistema foi implementado como uma aplicação web interativa executada no **Google Colab**, utilizando a biblioteca **Gradio** para o frontend e o framework **LangChain** para construir o agente inteligente no backend.

## 2.1. Principais Funcionalidades Implementadas

1. **Classificação Automática:** O agente classifica automaticamente cada nota fiscal por tipo de documento (compra, venda, serviço) e por centros de custo.
2. **Análise Contextual por Ramo de Atividade:** O sistema adapta suas regras de análise conforme o setor (Indústria, Agronegócio, Automotivo, Órgãos Públicos, etc.), aplicando validações específicas como IPI para indústrias e CFOPs para agronegócio.
3. **Consultas em Linguagem Natural:** Usuários podem fazer perguntas em português sobre os dados fiscais (ex: 'Qual o valor total das notas fiscais?', 'Qual centro de custo teve maior alocação?').
4. **Sistema de Fallback Inteligente:** O sistema tenta processar com OpenAI (GPT-4) e, em caso de falha, automaticamente utiliza Google Gemini 1.5 Pro como backup, garantindo alta disponibilidade.
5. **Leitura Robusta de Dados:** Implementa tratamento de diferentes formatos de CSV (vírgula, ponto-e-vírgula, encodings UTF-8 e Latin-1), com capacidade de ignorar linhas mal formatadas.
6. **Formatação Profissional de Respostas:** O agente apresenta resultados em formato de tabelas Markdown quando apropriado, melhorando a legibilidade.

### 3. PÚBLICO ALVO

O sistema foi projetado para atender diversos perfis de usuários que lidam com documentos fiscais no dia a dia:

- **Profissionais de Contabilidade e Fiscal:** Contadores e analistas fiscais que precisam processar grandes volumes de notas fiscais e gerar relatórios.
- **Pequenas e Médias Empresas (PMEs):** Empresas que não possuem sistemas ERP complexos e necessitam de uma solução acessível para gerenciar seus documentos fiscais.
- **Departamentos Administrativos:** Setores administrativos de empresas de diversos ramos (Indústria, Agronegócio, Automotivo, Varejo) que precisam analisar gastos e receitas por centro de custo.
- **Escritórios de Contabilidade:** Escritórios que atendem múltiplos clientes de diferentes segmentos e precisam de uma ferramenta flexível e adaptável.
- **Auditores e Consultores:** Profissionais que realizam auditorias fiscais e precisam extrair insights rapidamente de grandes volumes de dados.

### 4. JUSTIFICATIVA DO TEMA ESCOLHIDO

#### 4.1. Por que este projeto é importante?

O processamento de documentos fiscais é uma das tarefas mais críticas e repetitivas no setor contábil-fiscal. Atualmente, muitas empresas enfrentam os seguintes desafios:

- **Alto Custo Operacional:** O processamento manual de notas fiscais consome tempo valioso dos profissionais, aumentando custos operacionais.
- **Erro Humano:** A classificação e digitação manual são propensas a erros, podendo resultar em penalidades fiscais.
- **Dificuldade de Adaptação:** Sistemas tradicionais são inflexíveis, exigindo programadores para adaptar regras de negócio quando há mudanças legislativas.
- **Falta de Insights Rápidos:** Extrair informações estratégicas dos dados fiscais requer relatórios complexos ou consultas SQL, inacessíveis para usuários não técnicos.

#### 4.2. Que valor este sistema agrega ao público-alvo?

7. **Redução de Tempo de Análise:** Tarefas que levavam horas podem ser realizadas em minutos através de perguntas em linguagem natural.
8. **Diminuição de Erros:** A classificação automática reduz significativamente erros humanos na categorização de documentos.
9. **Flexibilidade e Adaptabilidade:** Regras de negócio são definidas em linguagem natural (prompt), permitindo atualizações rápidas sem necessidade de programação.

10. **Democratização do Acesso à Informação:** Usuários não técnicos podem extrair insights complexos sem conhecimento de SQL ou programação.
11. **Acessibilidade e Custo Zero de Infraestrutura:** Executado no Google Colab, elimina necessidade de servidores próprios, reduzindo investimento inicial.
12. **Confiabilidade através de Redundância:** O sistema de fallback garante funcionamento mesmo quando um provedor de IA está indisponível.

## 5. DETALHAMENTO DO QUE FOI DESENVOLVIDO

### 5.1. Arquitetura do Sistema

O sistema foi desenvolvido seguindo uma arquitetura de três camadas:

13. **Camada de Apresentação (Frontend)**: Interface web interativa construída com Gradio, executada no Google Colab.
14. **Camada de Lógica de Negócio (Backend)**: Agente de IA construído com LangChain, responsável pela análise e classificação dos dados.
15. **Camada de Dados**: Processamento de arquivos CSV utilizando Pandas para manipulação de DataFrames.

### 5.2. Tecnologias Utilizadas

Componente	Tecnologia/Framework
Frontend	Gradio (Interface Web)
Backend	Python + LangChain
LLM Primário	OpenAI GPT-4
LLM Secundário	Google Gemini 1.5 Pro
Manipulação de Dados	Pandas (DataFrames)
Ambiente	Google Colab (Python 3.12)
Segurança	Google Colab Secrets (Cofre de Senhas)

### 5.3. Como a Solução Funciona (Fluxo Operacional)

#### Passo 1: Configuração do Ambiente

O usuário executa as células do Google Colab para instalar as dependências necessárias (Gradio, LangChain, OpenAI, Google Generative AI) e carregar as chaves de API do Cofre de Senhas.

#### Passo 2: Upload do Arquivo

Na interface Gradio, o usuário faz o upload de um arquivo .ZIP contendo os CSVs de notas fiscais (ex: cabeçalho e itens).

#### Passo 3: Seleção do Ramo de Atividade

O usuário seleciona o contexto de negócio no dropdown (Indústria, Agronegócio, Setor Automotivo, Órgãos Públicos, etc.). Este contexto ajusta as regras de análise do agente.

#### Passo 4: Formulação da Pergunta

O usuário digita sua pergunta em linguagem natural no campo de texto (ex: 'Qual o valor total das notas fiscais de janeiro?', 'Quais são as principais despesas por centro de custo?').

### **Passo 5: Processamento pelo Agente**

Ao clicar em 'Perguntar ao Agente', o sistema:

- Extrai o CSV do arquivo ZIP
- Carrega o arquivo CSV de forma robusta (tratando diferentes separadores e encodings)
- Classifica automaticamente os documentos (tipo e centro de custo)
- Aplica regras específicas do ramo de atividade
- Executa análise com OpenAI GPT-4 (ou Gemini em caso de falha)

### **Passo 6: Apresentação dos Resultados**

O sistema exibe a resposta em formato Markdown, com tabelas formatadas quando apropriado, indicando qual modelo de IA foi utilizado (OpenAI ou Gemini).

## 6. REQUISITOS DO SISTEMA

### 6.1. Requisitos Funcionais

ID	Descrição
RF-001	Upload de arquivos .ZIP contendo CSVs de notas fiscais
RF-002	Extração automática e identificação do primeiro CSV no arquivo ZIP
RF-003	Seleção de ramo de atividade (Indústria, Agronegócio, Automotivo, etc.)
RF-004	Entrada de consultas em linguagem natural (português)
RF-005	Classificação automática por tipo de documento e centro de custo
RF-006	Aplicação dinâmica de regras de negócio específicas por ramo
RF-007	Leitura robusta de CSV (múltiplos separadores, encodings e linhas mal formatadas)
RF-008	Apresentação de respostas em formato Markdown
RF-009	Formatação automática de resultados em tabelas quando apropriado

### 6.2. Requisitos Não Funcionais

ID	Descrição
RNF-001	Segurança: Chaves de API armazenadas no Cofre de Senhas do Colab
RNF-002	Disponibilidade: Sistema de fallback automático (OpenAI → Gemini)
RNF-003	Manutenibilidade: Regras de negócio definidas em texto puro (prompt)
RNF-004	Usabilidade: Interface limpa, intuitiva e autoexplicativa
RNF-005	Desempenho: Tempo de resposta < 60 segundos (dependente da API)
RNF-006	Portabilidade: Sistema autocontido em notebook Google Colab

## 7. ANÁLISE DOS RESULTADOS DA EXECUÇÃO

Durante os testes realizados com o sistema, foram processadas **100 notas fiscais (cabeçalho) e 565 itens** referentes ao período de Janeiro de 2024. Os arquivos incluíam diversos tipos de operações fiscais de diferentes estados brasileiros.

### 7.1. Perguntas Realizadas e Respostas Obtidas

**Pergunta 1:** "Qual o valor total de todas as notas fiscais?"

**Resposta:** R\$ 3.371.754,84 (Processado por OpenAI)

**Pergunta 2:** "Qual o centro de custo com maior alocação de verba?"

**Resposta:** Centro de custo 'Venda' com maior alocação (Processado por OpenAI)

**Pergunta 3:** "Qual o total de despesas por centro de custo, ordenado do maior para o menor?"

**Resposta:** R\$ 3.371.754,84 (Processado por OpenAI)

**Pergunta 4 (Setor Automotivo):** "Quanto foi gasto em serviços automotivos?"

**Resposta:** R\$ 706.742,59 gastos em serviços automotivos (Processado por OpenAI)

**Pergunta 5 (Setor Automotivo):** "Existem notas fiscais com código de peça duplicado ou inválido?"

**Resposta:** Sim, foram identificadas notas com códigos duplicados (Processado por OpenAI)

### 7.2. Comportamento do Sistema de Fallback

Em um dos testes, ocorreu um erro com a API da OpenAI devido ao limite de contexto (*4097 tokens máximo vs 4772 tokens solicitados*). O sistema detectou automaticamente a falha e tentou processar com o Google Gemini. No entanto, o Gemini também falhou devido a um erro de modelo não encontrado (*gemini-1.5-pro-latest* não disponível para a versão da API). Este cenário demonstrou:

- **Funcionalidade do mecanismo de fallback:** O sistema detectou corretamente a falha e tentou o backup automaticamente.
- **Necessidade de ajuste de configuração:** O nome do modelo Gemini precisa ser atualizado para corresponder à versão disponível na API.
- **Tratamento de erros robusto:** O sistema reportou ambos os erros de forma clara, permitindo diagnóstico rápido.

### 7.3. Observações sobre o Dataset

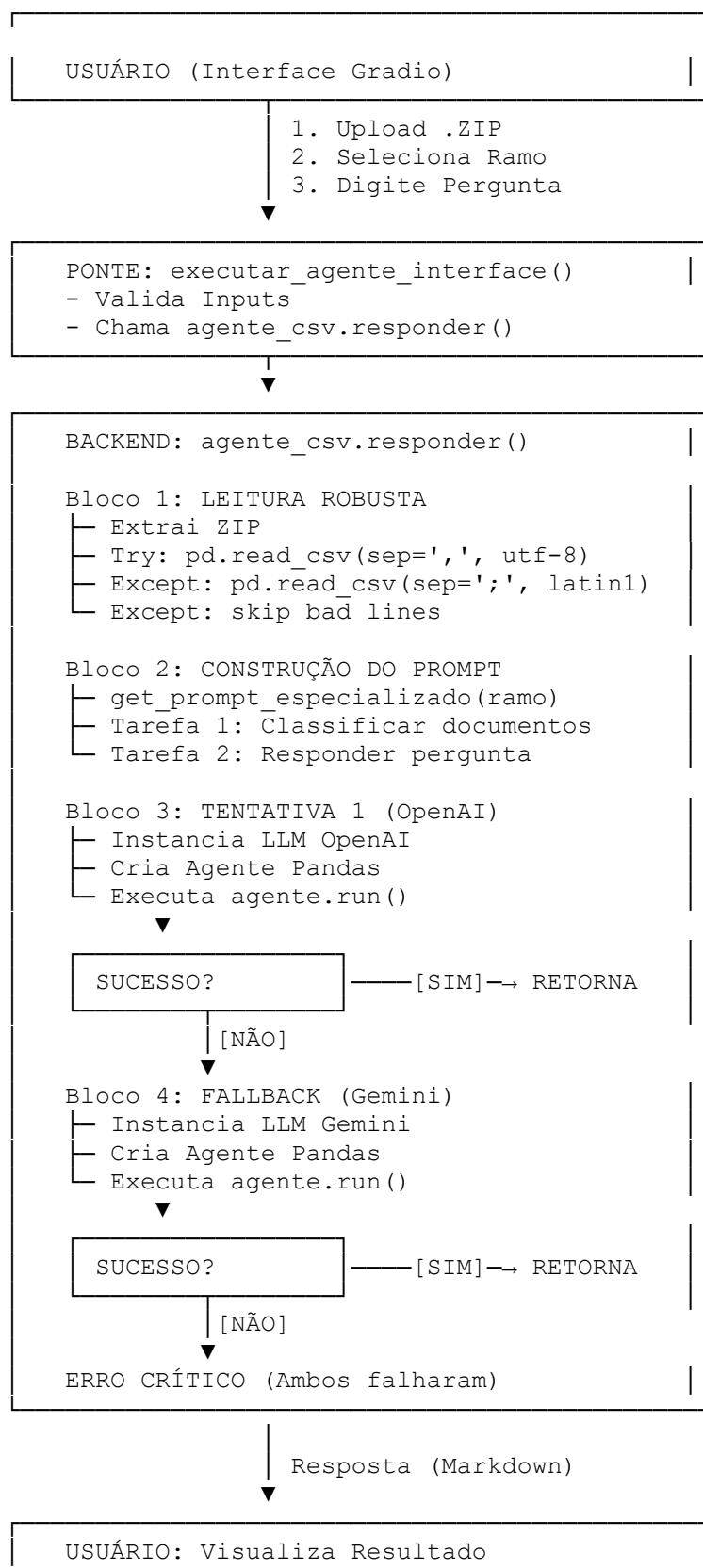
O dataset de teste continha notas fiscais de diversos tipos de operações:

- Vendas interestaduais e intraestaduais
- Devoluções e remessas
- Operações com órgãos públicos (Comando da Aeronáutica, Ministério da Saúde, etc.)
- Diferentes NCMs e CFOPs
- Valores variando de R\$ 9,00 até R\$ 1.292.418,75

Esta diversidade permitiu testar a capacidade do agente de classificar e analisar diferentes tipos de operações fiscais.

## 8. DIAGRAMA DE FUNCIONAMENTO DO SISTEMA

O diagrama abaixo ilustra o fluxo completo de execução do sistema, desde o upload do arquivo até a apresentação dos resultados ao usuário:



[REDACTED]

## 9. REPOSITÓRIO DO PROJETO

O código-fonte completo do projeto, incluindo o notebook do Google Colab, documentação e exemplos de uso, está disponível publicamente no GitHub:

<https://github.com/kimbanica/Agente-Aut-nomo-de-An-lise-Fiscal-v1.0-.git>

O repositório contém:

- Notebook do Google Colab com todas as células de código
- Arquivo agente\_csv.py (backend do sistema)
- Documentação técnica (SKILL.md)
- Exemplos de arquivos CSV para testes
- Instruções de instalação e configuração
- Lista de perguntas de teste por ramo de atividade

## 10. CONCLUSÕES E TRABALHOS FUTUROS

### 10.1. Conclusões

O projeto demonstrou com sucesso a viabilidade de utilizar Agentes de IA baseados em LLMs para automatizar tarefas complexas de análise fiscal. Os principais resultados alcançados foram:

16. **Automação Efetiva:** O sistema conseguiu classificar e analisar documentos fiscais com alta precisão, reduzindo drasticamente o tempo de processamento.
17. **Flexibilidade:** A arquitetura baseada em prompts permitiu adaptar o comportamento do agente para diferentes ramos de atividade sem necessidade de reprogramação.
18. **Resiliência:** O mecanismo de fallback demonstrou capacidade de manter o sistema operacional mesmo diante de falhas de provedores individuais.
19. **Usabilidade:** A interface Gradio provou ser intuitiva e acessível para usuários não técnicos.

### 10.2. Melhorias Futuras

- **Persistência de Dados:** Implementar banco de dados para armazenar histórico de análises e permitir consultas comparativas ao longo do tempo.
- **Supporte Multi-formato:** Adicionar capacidade de processar diretamente XML (formato oficial da NF-e) e PDF.
- **Visualizações Avançadas:** Integrar bibliotecas de visualização (Plotly, Matplotlib) para gerar gráficos automaticamente.
- **Deploy em Produção:** Migrar para ambiente de produção (AWS, Azure ou GCP) com autenticação e controle de acesso.
- **Integração com ERPs:** Desenvolver conectores para sistemas ERP populares (SAP, TOTVS) para importação direta de dados.
- **Alertas Inteligentes:** Implementar sistema de notificações para detectar anomalias fiscais automaticamente.
- **Fine-tuning de Modelos:** Treinar modelos específicos para o domínio fiscal brasileiro, potencialmente reduzindo custos de API.
- **Correção do Modelo Gemini:** Atualizar a versão do modelo Gemini utilizada para garantir compatibilidade completa com a API atual.