Entrega Final: Relatório de Implantação dos agentes inteligentes na solução: **Plataforma de Análise de Notas Fiscais SMART** Financial Solutions

Github do Grupo: https://github.com/gabryel-barboza/smart_financial_solutions

Grupo: Grupo Comunicação Integrantes:

Nome	**E-mail**
Rosenildes Melo	rosenildesmelo@gmail.com
Luciano Cyllio	cyllio@gmail.com
Luis Gustavo	eng.gustavo90lima@gmail.com
Thiago Araújo	thiagojerem@gmail.com
Gabryel Barbosa	gabrielbarbosa.alternativa@gmail.com
Thais Petrin	tha.petrin@gmail.com
Flavio Pereira	flavupereira@gmail.com

Relatório Executivo: Smart Financial Solutions v2.0

Plataforma de Análise Fiscal com Inteligência Artificial e RAG

---

1. Sumário Executivo

O **Smart Financial Solutions** é um sistema inteligente de análise financeira projetado para facilitar a compreensão de dados de forma rápida e conversacional. A solução opera por meio de uma equipe de assistentes de Inteligência Artificial (IA) que colaboram para processar informações complexas. A arquitetura é dividida em um serviço de processamento robusto (o "cérebro") que executa as análises e uma interface de conversação amigável (o "rosto") para a interação direta do usuário. O conjunto completo é empacotado para instalação e execução simplificada utilizando contêineres Docker.

Este projeto permite aos usuários realizar tarefas financeiras complexas conversando com o sistema. As funcionalidades principais incluem a **Análise de Dados Simples**, onde o usuário pode enviar arquivos (como planilhas) e solicitar resumos ou tendências à IA. Além disso, o sistema oferece **Busca Inteligente** sobre documentos (incluindo XMLs), encontrando informações relevantes mesmo em textos menos claros. Ao final do processamento, o sistema pode gerar **Relatórios detalhados em PDF** e, se desejado, enviá-los por e-mail, acompanhados de **Visualização Automática** com gráficos exibidos diretamente na tela de chat para facilitar a compreensão dos resultados.

---

2. Descrição da Solução

A **Plataforma de Análise de Notas Fiscais SMART v2.0** é um sistema automatizado e inteligente que revoluciona a maneira como as empresas gerenciam, extraem valor e garantem conformidade de seus dados fiscais. A solução abrange desde o upload e a análise automatizada de documentos fiscais até a geração de relatórios dinâmicos, com um diferencial estratégico: a capacidade de realizar **buscas semânticas** em toda a base de documentos fiscais armazenados.

Frontend Interativo (React / TypeScript)

O frontend é um **single-page application (SPA)** interativo que provê a experiência conversacional e de upload.

Funcionalidade	Detalhe Técnico
Interface Conversacional	Chatbot que responde perguntas, aceita upload de arquivos e gerencia o histórico de mensagens.
Gerenciamento de Estado	Utiliza Context API e ServerContext para gerenciar o estado da aplicação.
Renderização de Gráficos	O frontend recebe o ID para busca de gráficos do backend, requisita o JSON do gráfico resultante e renderiza o gráfico de forma dinâmica com Plotly.js.
Upload Assíncrono	Gerencia o upload de arquivos de dados (CSV/XLSX/ZIP) e imagens (JPEG, PNG, TIFF, BMP), utilizando o canal WebSocket para mostrar o status de processamento em tempo real.
Configuração Dinâmica	A ConfigPage permite o mapeamento e alteração dos modelos LLM (ex: llama3-8b) para tarefas/agentes específicas (SUPERVISOR, DATA ANALYST), enviando a configuração para o backend.

2.1. Principais Funcionalidades

A plataforma oferece um conjunto abrangente de funcionalidades que cobrem todo o ciclo de vida da gestão fiscal:

- **Ingestão Inteligente de Documentos Fiscais**: Processamento automatizado de XMLs de notas fiscais (NFe, NFCe, CTe) com extração e validação de campos críticos.
- **Armazenamento Vetorial com RAG**: Transformação de documentos fiscais em embeddings vetoriais para busca por similaridade semântica.
- **Busca Semântica Avançada**: Consultas em linguagem natural sobre toda a base de documentos fiscais, retornando resultados relevantes mesmo sem correspondência exata de palavras-chave.
- **Análise Estatística de Dados**: Geração de gráficos interativos, detecção de outliers, análise de correlação e clustering com K-Means.
- **Validação e Conformidade Fiscal**: Verificação automática de campos obrigatórios, validação de CNPJs, datas e valores.
- **Geração de Relatórios Profissionais**: Criação de relatórios em PDF com insights fiscais e envio automatizado por e-mail via SMTP.
- **Chat Inteligente com Contexto**: Interação em linguagem natural com acesso ao histórico de conversação e dados fiscais armazenados.
- **Isolamento Multi-Tenancy**: Garantia de segurança e privacidade com isolamento completo dos dados de cada usuário.

2.2. Arquitetura RAG Implementada

A arquitetura de **Retrieval-Augmented Generation** permite que os agentes de IA acessem informações específicas dos documentos fiscais armazenados, combinando a capacidade de recuperação de informações com a geração de respostas contextualizadas.

Componentes principais da arquitetura RAG:

1. **Modelo de Embeddings**: FastEmbed com modelo paraphrase-multilingual-MiniLM-L12-v2 (384 dimensões)
2. **Vector Database**: Qdrant com algoritmo HNSW para busca aproximada de vizinhos mais próximos (ANN)
3. **Isolamento de Dados**: Índice de payload com campo metadata.user_id configurado como tenant
4. **Busca por Similaridade**: Cálculo de distância cosine entre vetores de query e documentos
5. **Contextualização**: LLMs utilizam os documentos recuperados para gerar respostas precisas e contextualizadas.

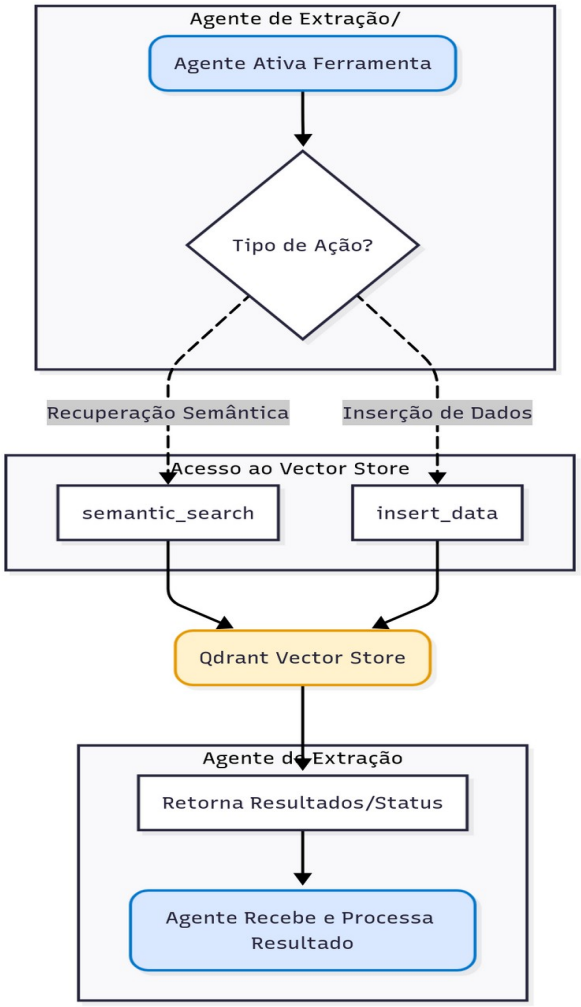


Figura 1: Diagrama do fluxo de interação com o banco de dados vetorial.

---

3. Público-Alvo

A aplicação é ideal para **empresas de todos os portes** que buscam otimizar sua gestão fiscal, de compliance e de auditoria. Ela se destina especialmente a:

- **Departamentos Fiscais e Contábeis**: Que necessitam de automação na validação e análise de notas fiscais.
- **Empresas de Auditoria**: Que realizam conferências cruzadas entre bases de dados fiscais e operacionais.
- **Gestores Financeiros**: Que precisam de insights rápidos sobre despesas, receitas e impostos.
- **Empresas com Alto Volume de Documentos Fiscais**: Que se beneficiam da busca semântica para localizar informações específicas rapidamente.
- **Organizações que Buscam Conformidade Fiscal**: Que necessitam de validação automática e rastreabilidade de documentos.

---

4. Instalação e Inicialização

Toda a aplicação é empacotada e executada através do **Docker Compose**, garantindo um setup rápido e confiável. Porém, o usuário tem a opção de clonar o projeto e inserir os comandos manualmente para colocar o projeto em execução.

4.1. Pré-requisitos

Para executar este projeto, você só precisa ter o **Docker** instalado na sua máquina e ter no mínimo **4 GB de armazenamento livre** para a aplicação.

Considerações Importantes: na primeira execução do projeto, todas as imagens e dependências serão baixadas para o seu funcionamento. Esse processo, a depender da conexão do usuário, pode levar um tempo médio de **10 - 20 min**. Em execuções posteriores, as dependências já foram cacheadas e a execução é mais rápida.

4.2. Configuração do Ambiente

- **Clone o repositório:**

```
git clone https://github.com/gabryel-barboza/smart_financial_solutions.git
```

```
cd smart-financial-solutions
```

Uma alternativa mais simples é clicar em <> **Code** e baixar o ZIP do projeto, com a desvantagem de não sincronizar com o repositório remoto.

- **Configurar variáveis de ambiente:**

Copie o arquivo de exemplo .env.example e renomeie-o para .env. Preencha-o com suas credenciais, adicione uma chave de API do LangSmith para serviço de tracing dos agentes. Os valores padrões são o suficiente para o projeto funcionar.

```
cp .env.example .env
```

O arquivo .env, no mínimo:

```
# Credenciais para servidor de email (SMTP Gmail)
```

```
SENDER_EMAIL="seu_email@gmail.com"
SENDER_PASSWORD="sua_credencial_de_app"

# Configurações do Qdrant
QDRANT_URL="http://qdrant:6333"

# Configurações da conexão com banco de dados
DATABASE_URI="sqlite:///databases/db.sqlite"

# Configurações do Langsmith para rastreamento das LLMs (Opcional)
LANGSMITH_TRACING=true
LANGSMITH_API_KEY="api_key"
```

```
LANGSMITH_PROJECT="smart_financial_solutions"
```

4.2.1. Inicialização da Aplicação Manual

Se optar pela inicialização manual, o projeto será executado em modo de desenvolvimento. A conexão com Qdrant Vector Store deve ser modificada para a sua instância, provavelmente no Qdrant Cloud.

Você precisará ter o **Node.js-20** e o **Python-3.12** instalados. Para começar acesse o diretório raiz do projeto e abra terminais nos diretórios frontend e backend.

- **Windows**: Abra um terminal pesquisando por CMD na barra de endereço (C:\user\) na pasta do projeto e pressionando ENTER ou pesquisando por CMD no menu Windows e navegando até o projeto com cd pasta1\pasta2\pasta3.
- **Linux**: Abra um terminal de preferência e navegue com o comando cd diretorio1/diretorio2/diretorio3.

Insira o seguinte comando no diretório frontend:

```
npm run dev
```

No diretório backend, insira os comandos a seguir no terminal:

```
# crie um ambiente virtual com:
```

```
python -m venv .venv

# ou outro gerenciador de ambientes virtuais e ative-o com:
```

```
.venv/Scripts/activate  # Windows
source .venv/bin/activate   # Linux

# Faça a instalação das dependências com:
pip install -r requirements.txt

# Execute o projeto com
```

```
fastapi dev src/main.py
```

Acesse os serviços nas rotas retornadas pelo terminal.

4.2.2 Inicialização da Aplicação com Docker

Para subir todos os serviços (**Frontend**, **Backend FastAPI** e **Qdrant**), execute o comando adiante no diretório raiz. Tenha certeza de estar no diretório que contém o arquivo compose.yml:

```
docker compose up --build
```

O argumento opcional --build garante que quaisquer atualizações no código sejam incorporadas nos containers, necessário quando houver mudanças no projeto.

Serviço	URL
Frontend (React)	http://localhost:8080
API Docs (FastAPI - Swagger UI)	http://localhost:8000/api/docs
Qdrant Dashboard	http://localhost:6333/dashboard

---

5. Justificativa e Geração de Valor

O cenário atual de gestão fiscal é marcado por ineficiências, complexidade da legislação tributária brasileira e fragmentação de dados. A dependência de processos manuais resulta em erros, inconsistências e um alto custo operacional. A plataforma SMART v2.0 aborda diretamente essas dores, oferecendo uma solução que vai além da automação básica.

5.1. Benefícios Estratégicos

Os principais benefícios e fontes de valor incluem:

- **Eliminação do Trabalho Manual**: Redução drástica de erros e do tempo gasto com tarefas repetitivas de digitação, validação e busca de documentos.
- **Extração Precisa de Informações**: Agentes de IA garantem acurácia e consistência na extração de dados de documentos fiscais estruturados e não estruturados.
- **Busca Inteligente e Rápida**: Localização instantânea de informações fiscais através de consultas em linguagem natural, sem necessidade de conhecer a estrutura exata dos dados.
- **Validação Instantânea e Padronizada**: A validação das informações fiscais é realizada de forma rápida e uniforme, minimizando riscos de não conformidade.
- **Auditoria Cruzada Automatizada**: Capacidade de correlacionar dados fiscais com outras bases internas da empresa (estoque, financeiro, RH).
- **Redução de Tempo e Custos**: Liberação de equipes para focar em atividades de maior valor agregado, impulsionando a produtividade.
- **Aceleração da Auditoria e Conformidade**: Visibilidade e controle aprimorados, garantindo a aderência às regulamentações fiscais e mitigando riscos de multas.
- **Escalabilidade e Segurança**: Arquitetura multi-tenancy permite atender múltiplos clientes com isolamento completo de dados.

5.2. Retorno sobre Investimento (ROI)

A implementação da plataforma SMART v2.0 gera retorno mensurável em múltiplas dimensões:

Métrica	Antes da Plataforma	Com a Plataforma	Ganho
Tempo de Validação de NF	5-10 min/documento	30 seg/documento	**90% de redução**
Busca de Informações Fiscais	15-30 min/consulta	1-2 min/consulta	**95% de redução**
Erros de Digitação	5-10% dos documentos	< 0.1%	**99% de redução**
Tempo de Auditoria	40-60 horas/mês	10-15 horas/mês	**75% de redução**
Custo Operacional	Alto (processos manuais)	Baixo (automatizado)	**60-70% de redução**

---

6. Detalhamento Técnico do Desenvolvimento

A seguir, apresentamos um detalhamento técnico da solução desenvolvida, incluindo a arquitetura do software, a implementação dos agentes de IA, as ferramentas utilizadas, as validações criadas e os relatórios gerenciais gerados.

6.1. Stack Tecnológica Completa

Backend

- **Framework**: FastAPI 0.119.0 (assíncrono)
- **Linguagem**: Python 3.12
- **Orquestração de IA**: LangChain 0.3.19 + LangGraph
- **Vector Database**: Qdrant (cliente assíncrono)
- **Modelo de Embeddings**: FastEmbed com paraphrase-multilingual-MiniLM-L12-v2
- **Provedores de LLM**:
    - Google Gemini 2.5 Flash e Pro (langchain-google-genai 2.1.12)
    - Groq com Llama 4 Maverick/Scout e Qwen 3 (langchain-groq 0.3.8)
    - Suporte para OpenAI via LangChain
- **Análise de Dados**: Pandas 2.3.2, NumPy 2.3.3
- **Visualização**: Plotly 6.3.0
- **Machine Learning**: Scikit-learn 1.7.2 (K-Means clustering)
- **OCR**: Pytesseract 0.3.13
- **Banco de Dados Relacional**: SQLite com SQLAlchemy 2.0.43
- **Geração de PDF**: markdown-pdf 1.10
- **Envio de E-mail**: SMTP (smtplib) com suporte a Gmail
- **Servidor ASGI**: Uvicorn 0.35.0

Frontend

- **Framework**: React 18
- **Linguagem**: TypeScript
- **Build Tool**: Vite
- **Visualização**: Plotly.js
- **Servidor Web**: Nginx (em produção)
- **Novos Componentes**: UserInput para identificação de usuário

Infraestrutura

- **Orquestração**: Docker Compose (3 serviços)
- **Tracing de LLM**: LangSmith
- **Persistência**: Volumes Docker para Qdrant e banco de dados

6.2. Arquitetura de Deployment Atualizada

A aplicação é orquestrada com **Docker Compose**, incluindo o serviço Qdrant para armazenamento vetorial, a API em FastAPI e a interface gráfica em React.

Serviços Docker:

6 **Frontend (Nginx:80)**: Interface React com novos componentes para identificação de usuário
7 **Backend (FastAPI:8000)**: API assíncrona com integração ao Qdrant
8 **Qdrant (6333)**: Vector database com HNSW para busca ANN

Volumes Persistentes:

- qdrant_storage: Armazenamento dos vetores e índices
- fastembed_cache: Cache dos modelos de embeddings
- databases/: Banco de dados SQLite

6.3. Envio de Relatórios via SMTP

A plataforma utiliza o protocolo **SMTP** para envio automatizado de relatórios PDF por e-mail. A implementação utiliza a biblioteca nativa smtplib do Python, com suporte específico para Gmail.

Fluxo do Relatório PDF:

9 O **Report Generation Agent** usa a ferramenta create_and_send_report para criar um arquivo PDF a partir de conteúdo Markdown.
10 Após a criação do PDF, o agente utiliza a função _send_report que estabelece uma conexão SMTP com o servidor Gmail (smtp.gmail.com:587).
11 O e-mail é enviado com o PDF anexado para o destinatário especificado pelo usuário.
12 A função retorna confirmação de sucesso ou mensagem de erro detalhada.

Configuração Necessária:

Para utilizar o Gmail como servidor SMTP, é necessário:

13 Habilitar a autenticação de dois fatores na conta Google
14 Gerar uma "Senha de App" específica para a aplicação
15 Configurar as variáveis SENDER_EMAIL e SENDER_PASSWORD no arquivo .env

---

7. Agentes de Inteligência Artificial

A inteligência da aplicação é distribuída em um **sistema multiagente**, onde cada agente possui uma especialização e um conjunto de ferramentas para cumprir suas tarefas. A orquestração é liderada por um agente Supervisor, que delega o trabalho para os agentes apropriados.

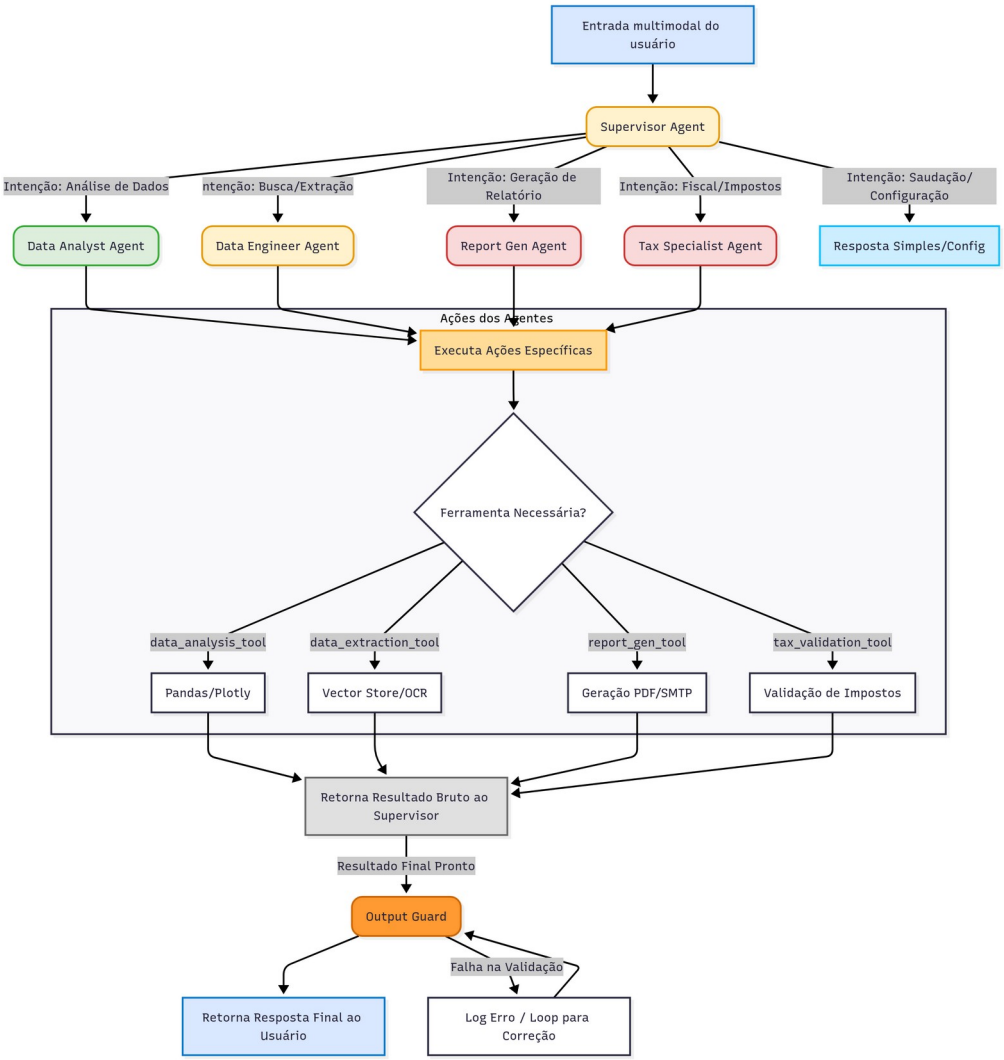


Figura 2: Diagrama com o fluxo de orquestração dos agentes.

7.1. Supervisor Agent (Smartie)

Responsabilidade: Atua como o orquestrador central. Ele recebe todas as solicitações do usuário, interpreta a intenção e delega a tarefa para o agente especialista mais adequado.

Modelo LLM: Qwen 3 32B (Groq) ou Gemini 2.5 Flash (Google)

Ferramentas:

- use_agent_tool: Invoca outros agentes especializados
- get_current_datetime: Fornece data e hora atual

Workflow:

16 Recebe a solicitação do usuário
17 Analisa a intenção e o contexto
18 Identifica o agente especialista mais adequado
19 Delega a tarefa via use_agent_tool
20 Recebe o resultado e formata a resposta final

7.2. Data Engineer Agent - Especialista em Extração Fiscal

Responsabilidade: Especializado em extrair, validar e armazenar dados de documentos fiscais brasileiros (XMLs de NFe, NFCe, CTe) no banco de dados vetorial Qdrant.

Modelo LLM: Gemini 2.5 Flash (Google)

Ferramentas:

- qdrant_data_insert: Insere dados validados no Qdrant com embeddings
- extract_data: Busca documentos por similaridade semântica

Campos Fiscais Extraídos:

Campo XML	Descrição	Validação
chNFe	Chave de Acesso da NFe (44 dígitos)	Formato numérico, 44 caracteres
CNPJ (Emitente/Destinatário)	CNPJ das partes envolvidas	14 dígitos numéricos
dhEmi	Data e Hora da Emissão	Formato ISO 8601
vNF	Valor Total da Nota Fiscal	Numérico, maior que zero
natOp	Natureza da Operação	Texto descritivo
xProd	Descrição do Produto/Serviço	Texto
NCM	Nomenclatura Comum do Mercosul	8 dígitos
CFOP	Código Fiscal de Operações	4 dígitos
CST/CSOSN	Código de Situação Tributária	2-3 dígitos
vICMS, vIPI, vPIS, vCOFINS	Valores dos Impostos	Numérico

Campo XML	Descrição	Validação
vBCICMS, pICMS	Base de Cálculo e Alíquota ICMS	Numérico

Workflow:

21 **Identificação e Validação**: Identifica se o input é um documento fiscal e valida campos críticos
22 **Extração de Dados**: Extrai campos fiscais estruturados do XML
23 **Criação de Chunks**: Cria chunks descritivos para embeddings
24 **Geração de Embeddings**: Vetoriza os chunks com FastEmbed
25 **Persistência**: Insere no Qdrant com metadata incluindo user_id para isolamento

7.3. Data Analyst Agent - Análise Estatística

Responsabilidade: Especializado em realizar análises exploratórias de dados (EDA), gerar insights e criar visualizações (gráficos).

Modelo LLM: Llama 4 Maverick 17B (Groq) ou Gemini 2.5 Pro (Google)

Ferramentas (15+ ferramentas de análise):

- get_data_summary: Resumo estatístico do dataset
- create_histogram: Histograma para colunas numéricas
- create_bar_chart: Gráfico de barras
- create_scatter_plot: Gráfico de dispersão
- create_line_chart: Gráfico de linhas
- create_box_plot: Box plot para detecção de outliers
- create_correlation_heatmap: Matriz de correlação
- detect_outliers_iqr: Detecção de outliers com IQR
- create_cluster_plot: Análise de clusters com K-Means
- python_ast_repl: Execução segura de código Python customizado

Workflow:

26 **Exploração**: Usa get_data_summary para entender a estrutura dos dados
27 **Planejamento**: Formula um plano de análise baseado na solicitação do usuário
28 **Seleção de Ferramenta**: Escolhe a ferramenta mais apropriada
29 **Execução**: Realiza a análise e gera visualizações
30 **Otimização**: Retorna apenas graph_id e metadata (não o JSON completo do gráfico)

7.4. Tax Specialist Agent - Cálculos Fiscais

Responsabilidade: Especializado em recuperar dados do Qdrant e realizar cálculos e validações de impostos.

Status: Em desenvolvimento (estrutura criada)

Funcionalidades Planejadas:

- Recuperação de dados fiscais via busca semântica
- Cálculo de impostos (ICMS, IPI, PIS, COFINS)
- Validação de alíquotas aplicadas
- Detecção de inconsistências tributárias
- Simulação de cenários fiscais

7.5. Report Gen Agent - Geração de Relatórios

Responsabilidade: Criar documentos e relatórios profissionais em formato PDF e enviá-los por e-mail ao destinatário solicitado via SMTP.

Modelo LLM: Llama 3.3 70B Versatile (Groq) ou Gemini 2.5 Flash (Google)

Ferramentas:

- create_and_send_report: Cria PDF a partir de Markdown e envia por e-mail via SMTP
- Validação de e-mail com regex
- Geração de PDF com markdown-pdf

Workflow:

1. **Estruturação**: Organiza o conteúdo do relatório em Markdown
2. **Geração de PDF**: Converte Markdown para PDF com formatação profissional
3. **Validação de E-mail**: Valida o endereço de e-mail do destinatário
4. **Envio via SMTP**: Estabelece conexão SMTP e envia o relatório anexado
5. **Confirmação**: Retorna status de sucesso ou erro detalhado

---

8. Fluxos de Dados e Processos

8.1. Fluxo de Ingestão de Dados Fiscais

O processo de ingestão de documentos fiscais envolve múltiplas etapas de validação, extração e armazenamento vetorial.

Etapas do Processo:

36. **Recebimento do Documento**: Usuário envia XML de nota fiscal via frontend
37. **Roteamento pelo Supervisor**: Identifica como tarefa de extração e aciona Data Engineer
38. **Análise e Validação**: Data Engineer valida campos críticos (CNPJ, datas, valores)
39. **Extração de Dados**: Extrai campos fiscais e cria chunks descritivos
40. **Geração de Embeddings**: FastEmbed vetoriza os chunks com dimensionalidade de 384 posições.
41. **Armazenamento no Qdrant**: Insere vetores com metadata incluindo user_id para isolamento
42. **Confirmação**: Retorna IDs dos pontos inseridos e confirmação de sucesso

8.2. Fluxo de Busca Semântica (RAG)

A busca semântica permite que os usuários façam perguntas em linguagem natural e recebam respostas contextualizadas baseadas nos documentos fiscais armazenados.

Etapas do Processo:

43. **Pergunta do Usuário**: "Quais foram as despesas com material de escritório em fevereiro?"
44. **Análise pelo Supervisor**: Identifica necessidade de busca em dados fiscais
45. **Acionamento do Tax Specialist**: Delega para o agente de cálculos fiscais
46. **Extração de Dados**: Tax Specialist solicita busca ao Data Engineer
47. **Vetorização da Query**: FastEmbed gera embedding da pergunta
48. **Busca no Qdrant**: Busca ANN (HNSW) com filtro de user_id para isolamento
49. **Cálculo de Similaridade**: Qdrant calcula distância cosine entre query e documentos
50. **Retorno de Resultados**: Top 10 documentos mais similares com scores
51. **Agregação e Análise**: Tax Specialist agrega valores e gera insights
52. **Resposta Contextualizada**: LLM gera resposta formatada com os dados encontrados

---

9. Isolamento Multi-Tenancy e Segurança

A arquitetura multi-tenancy garante que os dados de cada usuário fiquem completamente isolados, permitindo que o ambiente seja seguro para múltiplos clientes sem risco de acesso indevido.

9.1. Implementação de Multi-Tenancy

Mecanismo de Isolamento:

- **Injeção de user_id**: Cada chunk de dados recebe automaticamente o ID da sessão no campo metadata.user_id.
- **Índice de Payload**: Campo metadata.user_id é indexado como tenant para otimização de performance de busca.
- **Filtros de Busca**: Todas as buscas incluem obrigatoriamente um filtro por user_id.
- **Validação de Acesso**: O sistema valida que o ID da sessão corresponde ao user_id dos dados.

9.2. Garantias de Segurança

Aspecto de Segurança	Implementação	Nível de Proteção
Isolamento de Dados	Filtro obrigatório por user_id em todas as buscas	✅ Alto
Validação de Entrada	Pydantic schemas em todos os endpoints	✅ Alto
Proteção contra Prompt Injection	Instruções de sistema com regras estritas	✅ Médio
Execução Segura de Código	AST REPL para código Python	✅ Alto
Validação de E-mail	Regex pattern matching	✅ Alto
SMTP Seguro	TLS/STARTTLS com autenticação	✅ Alto
Autenticação de Usuários	Não implementada	❌ Baixo
Rate Limiting	Não implementado	❌ Baixo

---

10. Resultados

Esta seção apresenta capturas de tela da plataforma em funcionamento, demonstrando as principais funcionalidades implementadas.

10.1. Interface Principal do Frontend

A interface principal da plataforma apresenta um design moderno e intuitivo, com navegação lateral e área de conversação com o agente.

Figura 3: Tela Principal da Plataforma SMART Financial Solutions]

Características da Interface:

- Menu lateral com acesso a Dashboard, Novo Upload, Histórico e Configurações
- Área de status do processamento em tempo real
- Chat conversacional com o agente Smartie

10.2. Interação com o Agente e Extração de Dados Fiscais

O agente analista processa documentos fiscais e extrai informações estruturadas automaticamente.

Figura 4: Chat com Agente e Extração de Dados de Nota Fiscal]

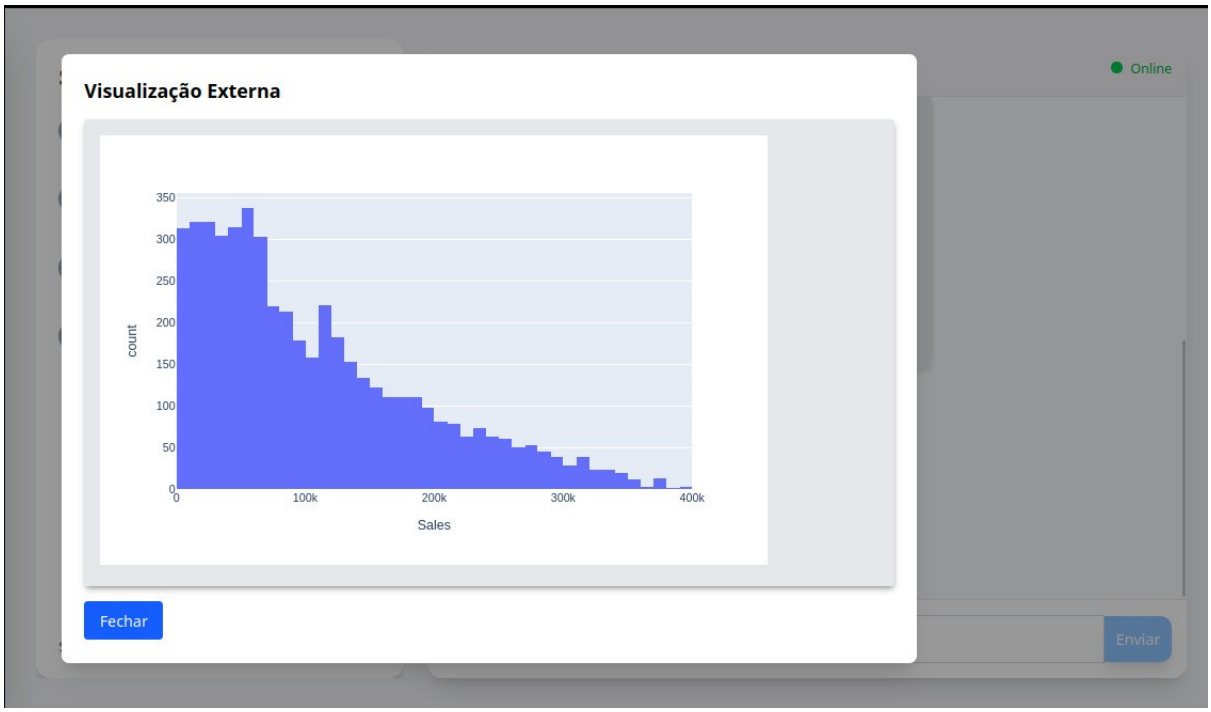
Funcionalidades Demonstradas:

- Status do processamento com etapas visuais
- Extração automática de campos fiscais (Emitente, Item, Valor total, Impostos)
- Resposta estruturada do agente com dados extraídos
- Interface de entrada para novas perguntas

10.3. Visualização de Gráficos Gerados

O Data Analyst Agent gera visualizações interativas utilizando Plotly.

Figura 5: Gráfico de Histograma Gerado pelo Data Analyst Agent



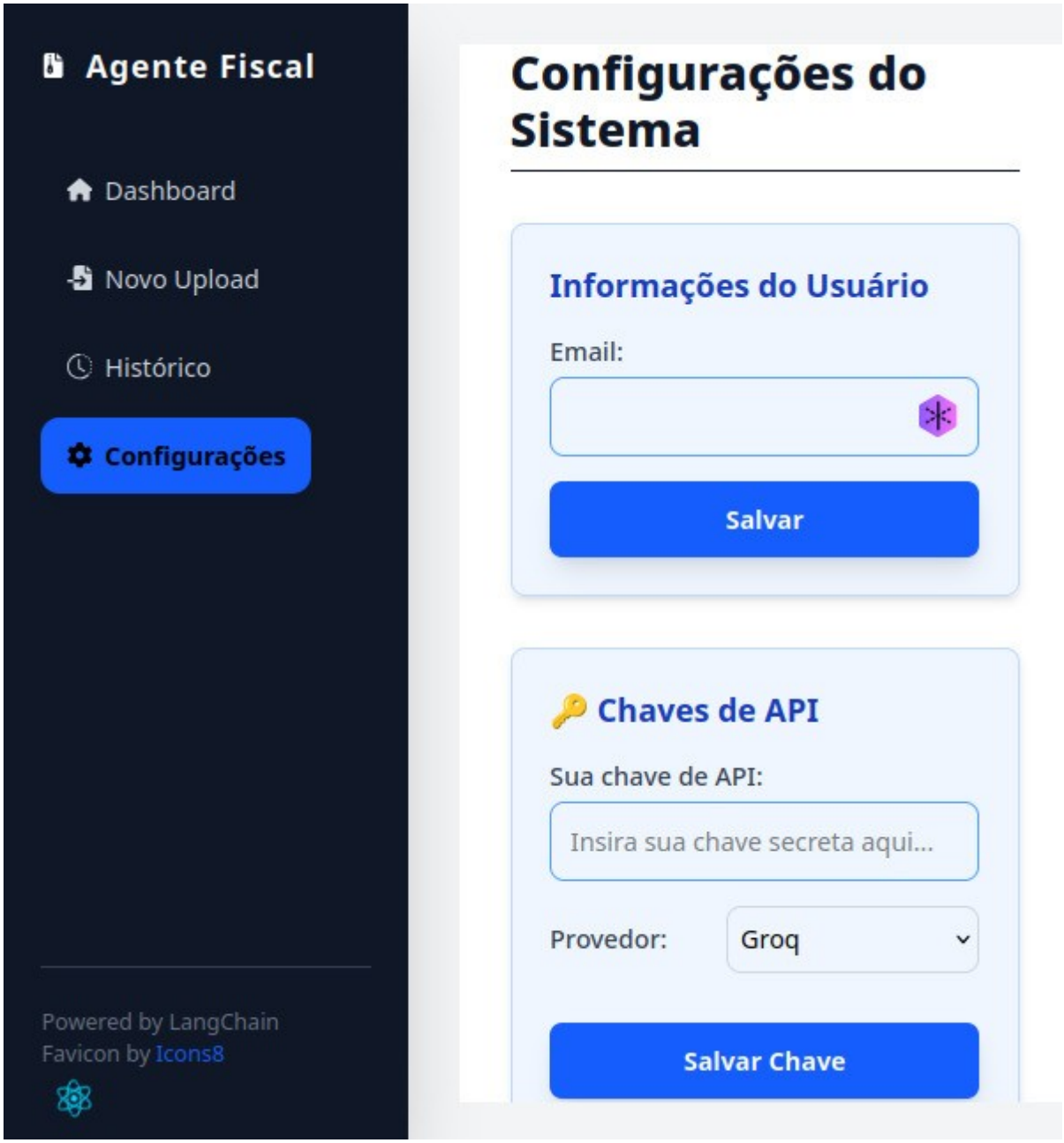
Características das Visualizações:

- Gráficos interativos renderizados com Plotly.js na interface
- Modal de visualização externa para análise detalhada
- Botão para fechar e retornar ao chat

10.4. Configurações do Sistema

A página de configurações permite personalizar o comportamento da plataforma.

Figura 6: Página de Configurações do Sistema



Opções de Configuração:

- **Informações do Usuário**: Cadastro de e-mail para recebimento de relatórios
- **Chaves de API**: Configuração de chaves de acesso aos provedores de LLM (Groq, Gemini, OpenAI)
- **Seleção de Provedor**: Escolha do provedor de LLM preferido
- **Interface Intuitiva**: Ícones visuais e botões de ação claros

---

11. Referências Bibliográficas

11.1. Frameworks e Bibliotecas Principais

Backend:

53 **FastAPI** (v0.119.0)
Tiangolo, S. (2018). *FastAPI: High performance, easy to learn, fast to code, ready for production*.
Disponível em: https://fastapi.tiangolo.com/

54 **LangChain** (v0.3.19)
Chase, H. (2022). *LangChain: Building applications with LLMs through composability*.
Disponível em: https://python.langchain.com/

55 **Qdrant** (Vector Database)
Qdrant Team. (2021). *Qdrant: Vector Database for the next generation of AI applications*.
Disponível em: https://qdrant.tech/

56 **FastEmbed**
Qdrant Team. (2023). *FastEmbed: Lightweight, fast, Python library for embeddings*.
Disponível em: https://github.com/qdrant/fastembed

57 **Pandas** (v2.3.2)
McKinney, W. (2010). *Data Structures for Statistical Computing in Python*.
Proceedings of the 9th Python in Science Conference, 56-61.

58 **Plotly** (v6.3.0)
Plotly Technologies Inc. (2015). *Collaborative data science*.
Disponível em: https://plot.ly

59 **SQLAlchemy** (v2.0.43)
Bayer, M. (2006). *SQLAlchemy: The Database Toolkit for Python*.
Disponível em: https://www.sqlalchemy.org/

60 **Pytesseract** (v0.3.13)
Smith, R. (2007). *An Overview of the Tesseract OCR Engine*.
Ninth International Conference on Document Analysis and Recognition (ICDAR 2007).

61 **Uvicorn** (v0.35.0)
Encode. (2017). *Uvicorn: The lightning-fast ASGI server*.
Disponível em: https://www.uvicorn.org/

Frontend:

62 **React** (v18)
Meta Platforms, Inc. (2013). *React: A JavaScript library for building user interfaces*.
Disponível em: https://react.dev/

63 **TypeScript**
Microsoft Corporation. (2012). *TypeScript: JavaScript with syntax for types*.
Disponível em: https://www.typescriptlang.org/

64 **Vite**
Evan You. (2020). *Vite: Next Generation Frontend Tooling*.
Disponível em: https://vitejs.dev/

65 **Plotly.js**
Plotly Technologies Inc. (2015). *Plotly.js: Open-source JavaScript charting library*.
Disponível em: https://plotly.com/javascript/

Provedores de LLM:

66 **Google Gemini**
Google DeepMind. (2023). *Gemini: A family of highly capable multimodal models*.
Disponível em: https://deepmind.google/technologies/gemini/

67 **Groq**
Groq, Inc. (2023). *Groq: The fastest AI inference in the world*.
Disponível em: https://groq.com/

68 *OpenAI OpenAI. (2023). GPT-4 Technical Report. Disponível em:*
https://openai.com/research/gpt-4

69 **LangSmith**
LangChain, Inc. (2023). *LangSmith: The all-in-one developer platform for every step of the LLM-powered application lifecycle*.
Disponível em: https://www.langchain.com/langsmith

11.2. Conceitos e Técnicas

70 **Bancos de Dados Vetoriais e Busca Semântica**Qdrant. (s.d.). **What is Qdrant?** *In:* Qdrant Documentation. Disponível em: https://qdrant.tech/documentation/overview/

71 **Retrieval-Augmented Generation (RAG)**Qdrant. (s.d.). **Retrieval Augmented Generation (RAG).** *In:* Qdrant Documentation. Disponível em: https://qdrant.tech/documentation/tutorials/rag/

72 **HNSW Algorithm (Implementação de ANN)**Qdrant. (s.d.). **Indexing and Search.** *In:* Qdrant Documentation. Disponível em: https://qdrant.tech/documentation/concepts/indexing/#hnsw

73 **Multi-Tenancy Architecture**Qdrant. (s.d.). **Multitenancy.** *In:* Qdrant Documentation. Disponível em: https://qdrant.tech/documentation/guides/multitenancy/

74 **Embeddings and Sentence Transformers**Qdrant. (s.d.). **What are Vector Embeddings?** *In:* Qdrant Documentation. Disponível em: https://qdrant.tech/documentation/concepts/vector-embeddings/

11.3. Documentação Fiscal Brasileira

75 **Nota Fiscal Eletrônica (NF-e)**
Portal da NF-e. *Nota Fiscal Eletrônica*.
Disponível em: https://www.nfe.fazenda.gov.br/

12. Conclusão

A aplicação **Smart Financial Solutions v2.0** representa um avanço significativo na gestão fiscal automatizada, combinando o poder da Inteligência Artificial com técnicas avançadas de **Retrieval-Augmented Generation (RAG)**. A integração do Qdrant Vector Database e a implementação do Data Engineer Agent especializado em documentos fiscais brasileiros posicionam a solução como uma ferramenta estratégica e de alto valor para empresas que buscam excelência em compliance e eficiência operacional.

12.1. Principais Conquistas

- **Arquitetura RAG Implementada**: Busca semântica em documentos fiscais com isolamento multi-tenancy
- **Data Engineer Agent**: Especialista em extração e validação de dados fiscais brasileiros
- **Validações Robustas**: Múltiplas camadas de validação garantem qualidade dos dados
- **Escalabilidade**: Arquitetura preparada para crescimento com Qdrant e FastAPI assíncrono
- **Segurança**: Isolamento completo de dados por usuário
- **Envio Automatizado de Relatórios**: Implementação SMTP nativa para envio de relatórios PDF

12.2. Diferenciais Competitivos

76 **Busca Semântica Inteligente**: Localização de informações fiscais por significado, não apenas por palavras-chave
77 **Especialização em Documentos Brasileiros**: Validação específica para NFe, CTe e outros documentos fiscais nacionais
78 **Multi-Tenancy Nativo**: Arquitetura preparada para SaaS desde o início
79 **Orquestração de Agentes**: Sistema multiagente com especialização clara de responsabilidades
80 **Extensibilidade**: Fácil adição de novos agentes e ferramentas

12.3. Impacto Esperado

A implementação completa da plataforma SMART v2.0, incluindo o Tax Specialist Agent planejado, tem potencial para:

- **Reduzir em 90% o tempo de validação de documentos fiscais**
- **Eliminar 99% dos erros de digitação e extração de dados**
- **Acelerar em 75% os processos de auditoria interna**

- **Reduzir em 60-70% os custos operacionais de gestão fiscal**
- **Aumentar a conformidade fiscal e reduzir riscos de multas**

A solução está **pronta para uso em ambiente de desenvolvimento e demonstração**, com uma base técnica sólida para evolução para produção em larga escala.

Data de Elaboração: 30 de Outubro de 2025
Versão do Relatório: 0.1
Repositório: https://github.com/gabryel-barboza/smart_financial_solutions