

I2A2 AGENTES INTELIGENTES PROJETO FINAL



Grupo: AgentesIA

Integrantes do Grupo:

Herbert – **hbrjunior@outlook.com**

Diego – **diegoquinto@gmail.com**

Ricardo – **contato@ricardoluz.com**

1. Descrição do Tema Escolhido:

O objetivo do projeto é desenvolver uma solução capaz de ler e processar automaticamente informações estruturadas de documentos fiscais (como notas fiscais eletrônicas – NFe, Nota Fiscal de Serviço NFS-e, cupons fiscais etc.) a partir de fontes como arquivos CSV. Com os dados fiscais estruturados armazenados em um banco de dados, o agente permitirá a extração de informações gerenciais valiosas, como indicadores de faturamento, impostos pagos, produtos mais vendidos e desempenho por cliente ou fornecedor. Será possível gerar KPIs contábeis e fiscais com agilidade, a solução também facilitará a integração com sistemas de gestão e a automação de obrigações acessórias, promovendo maior controle, precisão e eficiência na gestão fiscal de pequenos negócios.

2. Público alvo:

Pequenas Empresas e Negócios Informais Foco em empreendedores e comerciantes que operam com baixo volume de recursos tecnológicos e que frequentemente enfrentam dificuldades em organizar suas obrigações fiscais e dados de compras/vendas.

3. Justificativa do tema escolhido:

- Pequenas empresas ainda enfrentam dificuldades para acessar soluções simples e acessíveis que automatizem o processamento de dados fiscais, mesmo quando esses dados já estão estruturados.

I2A2 AGENTES INTELIGENTES PROJETO FINAL

- Apesar dos dados estarem estruturados, a diversidade de formatos e campos entre diferentes fontes (como sistemas emissores ou ERPs) ainda exige flexibilidade na leitura e interpretação das informações.
- A solução agregará valor ao eliminar a necessidade de digitação manual, reduzir erros operacionais e facilitar o controle tributário e contábil por meio da leitura automatizada de dados estruturados.
- Atende à crescente demanda por soluções digitais acessíveis que aproveitem dados já disponíveis em sistemas simples, especialmente no setor informal e entre pequenas e médias empresas.
- Pode ser facilmente integrado a sistemas de gestão (ERPs) básicos ou operado de forma autônoma, a partir da leitura de arquivos CSV ou conexões com bancos de dados.

4. Detalhamento do que foi desenvolvido

Agente Fiscal: CSV/XML → SQLite → Perguntas em Português

1) Visão geral (o que o sistema se propõe a fazer)

O sistema permite carregar dados fiscais estruturados a partir de CSVs, ZIPs com CSVs e XMLs, armazená-los automaticamente em um banco SQLite (uma tabela por arquivo) e, em seguida, realizar a consulta, consultá-los em linguagem natural (PT-BR). O agente converte a pergunta para SQL e executa no banco, retornando tabelas e análises úteis (ex.: faturamento, produtos mais vendidos, impostos, desempenho por cliente/fornecedor).

Benefício principal: criar um pipeline simples e universal para ingestão + consulta inteligente de dados fiscais para KPIs contábeis e fiscais, relatórios gerenciais e integração posterior com ERPs/BI.

2) Principais funcionalidades

- Upload multi-arquivo: CSV(s), ZIP com CSVs e XMLs.
- Criação automática do banco: um arquivo ⇒ uma tabela (nome higienizado).
 - Para XML de NF-e: cria duas tabelas (*_cabecalho, *_itens).

I2A2 AGENTES INTELIGENTES PROJETO FINAL

- Para XML no formato “Envio → detList → det → prod”: cria *_cabecalho (mínimo 1 coluna) + *_itens com cProd, xProd, qCom, vUnCom, vProd.
 - Para XML genérico: flatten em uma tabela única.
 - Normalização leve: nomes de colunas sem espaços; números convertidos para float quando possível.
 - Pré-visualização de dados e PRAGMA (schema) direto na UI.
 - Agente NL → SQL: perguntas em PT-BR; gera e executa o SELECT (com limpeza para evitar textos extras).
 - SQL manual (opcional): campo para você rodar SELECT diretamente.
 - Logs do agente: configuráveis (verbose/debug) para auditoria de como o SQL foi gerado.
-

3) Arquitetura resumida

- Frontend/Orquestração: Streamlit.
- Persistência: SQLite local (dados_fiscais.db, nome ajustável).
- Ingestão:
 - CSV: leitura resiliente (tenta sep=None/engine python + encodings comuns; fallback para ,).
 - ZIP: extração em memória e ingestão dos CSVs.
 - XML:
 - NF-e: mapeia nós ide, emit, dest, total(ICMSTot), det/prod.
 - Envio(det/prod): extrai cabeçalho (quando houver) e explode os itens para linhas.
 - Genérico: flatten profundo.
- IA/Agente: LangChain + LLM (config via .env).
 - Prompt do agente instruído para gerar apenas SQL SELECT.
 - Pós-processamento: extrai apenas o SQL a partir do primeiro SELECT (remove texto/markdown) e executa.

I2A2 AGENTES INTELIGENTES PROJETO FINAL

- Observabilidade: verbose/debug do LangChain habilitáveis para imprimir o raciocínio/ferramentas no terminal.
-

4) Como usar (passo a passo)

4.1. Preparar ambiente

1. (Opcional) Criar venv:
2. `python -m venv .venv`
3. `.\venv\Scripts\activate` # Windows
4. # ou
5. `source .venv/bin/activate` # macOS/Linux
6. Instalar dependências:
7. `pip install -r requirements_any_csv.txt`
8. Configurar .env (na pasta do app):
9. `LLM_API_KEY=SEU_TOKEN`
10. `LLM_MODEL_NAME=gpt-4o-mini` # ajuste conforme seu provedor
11. `LLM_BASE_URL=https://api.openai.com/v1`


4.2. Executar

`streamlit run app.py` # ou `app_any_csv_xml_v2.py`, conforme o seu arquivo

4.3. Carregar dados

- Na UI: **Upload de CSV/XML/ZIP** (multi-arquivo).
- Se “Recriar tabelas” estiver marcado, tabelas homônimas são **substituídas**.
- O app mostra **prévia** e lista as **tabelas** detectadas.

4.4. Consultar em linguagem natural

- Área “ Faça sua pergunta em Português”.
- Exemplos:
 - “Quais os 5 produtos mais vendidos?”
 - “Top 10 por valor vendido”

I2A2 AGENTES INTELIGENTES PROJETO FINAL

- “Total de faturamento por mês”
- O sistema:
 1. gera SQL (apenas SELECT),
 2. **limpa** qualquer texto extra,
 3. **executa** no SQLite,
 4. mostra o **resultado**.

4.5. Consultar em SQL manual (opcional)

- Campo “Executar SQL” para SELECT direto.
 - Exemplos:
 - SELECT xProd, SUM(qCom) AS total_qtd
 - FROM envio_construcao_10k_itens_itens
 - GROUP BY xProd
 - ORDER BY total_qtd DESC
 - LIMIT 5;
-

5) Casos de uso e indicadores gerados

- Vendas/Produtos: top itens por quantidade (SUM(qCom)) ou por valor (SUM(vProd)).
 - Financeiro/Fiscal (NF-e): faturamento (SUM(vNF)), base de produtos (SUM(vProd)), impostos (SUM(vICMS, vPIS, vCOFINS, vIPI)), margens (se houver campos de custo), distribuição por UF/cliente.
 - Clientes/Fornecedores: ranking por dest_CNPJ/emit_CNPJ, volume por período, ticket médio.
-

6) Esquemas de tabelas (principais)

6.1. CSV (genérico)

- Tabela: <nome_do_arquivo>
- Colunas: conforme o header do CSV (espaços viram _).

I2A2 AGENTES INTELIGENTES PROJETO FINAL

6.2. XML NF-e

- <base>_cabecalho:
nNF, serie, dhEmi, tpNF, natOp, cUF, tpAmb, mod, emit_CNPJ, emit_xNome, emit_UF, dest_CNPJ, dest_xNome, dest_UF, vNF, vProd, vICMS, vIPI, vPIS, vCOFINS
- <base>_itens:
nNF, cProd, xProd, NCM, CFOP, uCom, qCom, vUnCom, vProd

6.3. XML “Envio → detList → det → prod”

- <base>_cabecalho: pelo menos arquivo (e quaisquer metadados que o topo do XML tiver).
- <base>_itens: cProd, xProd, qCom, vUnCom, vProd (tipados como numéricos quando possível).

7) Boas práticas de uso

- Padronize nomes de arquivos para refletir o conteúdo (ex.: 202401_nfs_itens.csv).
- Evite colunas sem nome ou misturas de tipos no mesmo campo no CSV.
- Para XML, use o arquivo de fato (não HTML do DANFE, nem PDF ou ZIP renomeado).
- Para perguntas ao agente, use termos que existam nas colunas (ex.: xProd, qCom, vProd).

8) Troubleshooting (erros comuns e correções)

Sintoma	Causa	Correção
The 'low_memory' option is not supported with the 'python' engine	read_csv com engine="python" não aceita low_memory	Removido do código (feito).
ModuleNotFoundError: xmltodict	Pacote não instalado	pip install xmltodict ou pip install -r requirements_any_csv.txt.
XML inválido: not well-formed (invalid token)	Arquivo não é XML (HTML/PDF/ZIP) ou tem BOM/lixo	Versão robusta remove BOM, detecta formatos; usar o XML correto.
Consulta não executa e	O LLM devolveu	Limpador de SQL implementado

I2A2 AGENTES INTELIGENTES PROJETO FINAL

Sintoma	Causa	Correção
aparece texto “A tabela ... contém ...”	explicação + SQL	(pega só o SELECT).
near ")": syntax error ao gravar _cabecalho	DataFrame sem colunas	Garantido arquivo como coluna mínima ou pular a criação (feito).
Não vejo o “pensamento” no terminal	Verbose/desligado	verbose=True no LLM e no agente + set_debug(True)/set_verbose(True).

9) Segurança e configuração

- As chaves do LLM ficam no .env (não commitar).
- O SQLite fica local; proteja a pasta do projeto se houver dados sensíveis.
- Para ambientes corporativos, considere apontar para modelos self-hosted/VPC e armazenar o banco em volumes seguros/criptografados.

10) Limitações atuais

- Inferência de joins depende de nomes usuais (ex.: nNF, cProd, CNPJ); para schemas diferentes, o agente pode errar.
- XML genérico ainda vira uma tabela (flatten) se não casar com NF-e ou Envio(det/prod).
- Não há ainda catálogo de metadados/linhagem ou mapeamento de chaves persistente entre tabelas heterogêneas (opção futura).

11) Roadmap sugerido (próximas melhorias)

1. Dicionário de colunas (“alias”) por domínio fiscal para orientar o agente (ex.: tratar descricao como xProd).
2. Config UI de delimitador/encoding para CSV (override manual).
3. Suporte a NFS-e (padrões mais comuns) com tabelas normalizadas.
4. Validações fiscais: regras de integridade (ex.: vProd \approx qCom * vUnCom).

I2A2 AGENTES INTELIGENTES PROJETO FINAL

5. Camada de métricas (views) com KPIs prontos: faturamento mês, mix de produtos, ticket médio, top clientes.
 6. Exportação (CSV/Excel) de resultados e integração nativa com Power BI.
-

12) Exemplos de perguntas úteis

- “Top 5 produtos por quantidade”
 - “Top 10 por valor vendido”
 - “Total de faturamento (vNF) por mês”
 - “Ranking de clientes por valor total”
 - “Quantidade total por NCM”
 - “Média de preço por produto (vUnCom)”
-

Conclusão

Um pipeline único para ingerir dados fiscais (CSV/XML), estruturar em SQLite e perguntar em PT-BR com geração/execução de SQL. Isso acelera a criação de KPIs e análises fiscais/contábeis e prepara o terreno para integração com BI/ERP e automação de obrigações acessórias.

Prints da aplicação

I2A2 AGENTES INTELIGENTES PROJETO FINAL

Configurações

Nome do arquivo SQLite

dados_fiscais.db

Recriar tabelas se já existirem (replace)

Mostrar prévia de dados após carga

LLM

Modelo: gpt-4

Base URL: https://api.openai.com/v1

Agente Fiscal – CSV/XML → SQLite → Perguntas em Português

Faça upload de CSVs / XMLs (ou ZIP com CSVs). Para NF-e, geramos tabelas de cabeçalho e itens. O parser de XML agora é mais robusto (remove BOM, tenta múltiplos encodings, detecta HTML/ZIP por engano).

Upload de CSV(s)/XML(s) ou ZIP (CSV)

Selecione um ou múltiplos arquivos

Drag and drop files here

Limit 200MB per file • CSV, ZIP, XML

Browse files

Envio_construcao_10k_itens.xml

3.3MB

Prévia: envio_construcao_10k_itens_cabecalho (de Envio_construcao_10k_itens.xml)

	arquivo
0	Envio_construcao_10k_itens.xml

Prévia: envio_construcao_10k_itens_itens (de Envio_construcao_10k_itens.xml)

	cProd	xProd	qCom	vUnCom	vProd
0	100001	Lâmpada LED 9W	173	318.47	55095.31
1	100002	Brita 1 1m³	44	367.21	16157.24
2	100003	Telha cerâmica romana	35	551.03	19286.05

Faça sua pergunta em Português

Exemplos: 'Qual o total de faturamento por mês?', 'Quais os 10 produtos mais vendidos?'

Quais os 5 itens mais vendidos?

Consulta SQL gerada (pós-limpeza):

```
SELECT
  xProd,
  SUM(qCom) as total_sold
FROM
  envio_construcao_10k_itens_itens
GROUP BY
  xProd
ORDER BY
  total_sold DESC
LIMIT 5;
```

Resultado da consulta:

	xProd	total_sold
0	Argamassa AC-II 20kg	73341
1	Tinta látex 18L	73222
2	Cimento CP-II 50kg	71348
3	Rejunte 1kg	69645
4	Telha cerâmica romana	68567

I2A2 AGENTES INTELIGENTES PROJETO FINAL

Faça sua pergunta em Português

Exemplos: 'Qual o total de faturamento por mês?'; 'Quais os 10 produtos mais vendidos?'

Qual a soma total dessa nota fiscal ?

Consulta SQL gerada (pós-limpeza):

```
SELECT SUM(vProd) AS Total FROM envio_construcao_10k_itens_itens;
```

Resultado da consulta:

	Total
0	511108906.69

Faça sua pergunta em Português

Exemplos: 'Qual o total de faturamento por mês?'; 'Quais os 10 produtos mais vendidos?'

Liste os produtos mais vendidos em ordem decrescente pelo valor total

Consulta SQL gerada (pós-limpeza):

```
SELECT xProd, SUM(vProd) as total_vendido
FROM envio_construcao_10k_itens_itens
GROUP BY xProd
ORDER BY total_vendido DESC;
```

Resultado da consulta:


	xProd	total_vendido
0	Argamassa AC-II 20kg	44122427.01
1	Tinta látex 18L	43574681.59
2	Cimento CP-II 50kg	43260261.2
3	Rejunte 1kg	41463048.23
4	Telha cerâmica romana	41298278.39
5	Disjuntor 20A	40019139.5
6	Bloco de concreto 14x19x39	39125798.82
7	Tomada 2P+T 10A	39011869.59
8	Cano PVC 25mm 6m	38156108.32

I2A2 AGENTES INTELIGENTES PROJETO FINAL

Faça sua pergunta em Português

Exemplos: 'Qual o total de faturamento por mês?', 'Quais os 10 produtos mais vendidos?'

Quais produtos que foram menos vendidos?

 Consulta SQL gerada (pós-limpeza):

```
SELECT xProd, SUM(qCom) as total_vendido
FROM envio_construcao_10k_itens_itens
GROUP BY xProd
ORDER BY total_vendido ASC;
```

 Resultado da consulta:

	xProd	total_vendido
0	Porcelanato 60x60cm	16972
1	Areia Média 1m³	17089
2	Brita 1 1m³	17581
3	Lâmpada LED 9W	61739
4	Tomada 2P+T 10A	62645
5	Tijolo 8 furos	62769
6	Furadeira 500W	63301
7	Disjuntor 20A	63437
8	Bloco de concreto 14x19x39	65993

🔗 **LINK PARA O GITHUB:**
[AgentesIA_2025_ProjetoFinal](#)