



Agente Fiscal IA - Sistema de Análise de NF-e



Descrição

Sistema inteligente de processamento e análise automatizada de Notas Fiscais Eletrônicas (NF-e) desenvolvido com Python, utilizando IA generativa (Google Gemini) para validação inteligente e geração de insights.

Projeto Educacional - Curso de IA e Automação Fiscal



Funcionalidades



Extração Automática de Dados

- Parsing de arquivos XML de NF-e (padrão SEFAZ)
- Extração de dados de emitente, destinatário e produtos
- Identificação automática de impostos (ICMS, IPI, PIS, COFINS)
- Suporte a múltiplos arquivos simultâneos



Validação Inteligente

- Validação matemática de dígitos verificadores (CNPJ/CPF)
- Verificação de chave de acesso da NF-e
- Conferência automática de cálculos fiscais
- Análise de consistência de valores e impostos
- Validação com IA Generativa (Gemini)** - análise semântica e identificação de irregularidades



Geração de Relatórios

- Relatórios em Excel:** resumo consolidado, produtos e impostos
- Relatórios em PDF:** análise completa com insights de IA
- Dashboard interativo com métricas e gráficos
- Visualizações de dados fiscais



Tecnologias Utilizadas

Categoria	Tecnologia
Linguagem	Python 3.10+
Interface	Streamlit
IA Generativa	Google Gemini (via API)
Validação	Pydantic
Parsing XML	lxml
Relatórios	ReportLab, Pandas, OpenPyXL
Visualização	Plotly

Estrutura do Projeto



```
agente-fiscal-ia/
├── app.py           # Interface Streamlit (aplicação principal)
├── models.py        # Schemas Pydantic (modelos de dados)
├── extractor.py     # Extração de dados XML
├── validator.py     # Validação inteligente com IA
├── reporter.py      # Geração de relatórios PDF/Excel
├── requirements.txt # Dependências do projeto
└── README.md        # Este arquivo
```

Instalação e Execução

1. Pré-requisitos

- Python 3.10 ou superior
- Chave de API do Google Gemini ([obter aqui](#))

2. Clone ou baixe o projeto



```
bash

# Se estiver usando git
git clone <url-do-repositorio>
cd agente-fiscal-ia
```

3. Crie um ambiente virtual (recomendado)



```
bash
```

Windows

```
python -m venv venv  
venv\Scripts\activate
```

Linux/Mac

```
python3 -m venv venv  
source venv/bin/activate
```

4. Instale as dependências



bash

```
pip install -r requirements.txt
```

5. Execute a aplicação



bash

```
streamlit run app.py
```

A aplicação abrirá automaticamente no navegador em <http://localhost:8501>



Como Usar

Passo 1: Configurar API Key

1. Acesse a sidebar (menu lateral)
2. Insira sua API Key do Google Gemini
3. A chave é necessária para a validação inteligente


Passo 2: Upload de Arquivos

1. Navegue até "📁 Upload e Processamento"
2. Faça upload de um ou mais arquivos XML de NF-e
3. Clique em "🚀 Processar Notas Fiscais"
4. Aguarde o processamento


Passo 3: Visualizar Resultados

- **Dados:** Informações extraídas da nota
- **Validação:** Inconsistências e alertas encontrados
- **Análise IA:** Insights inteligentes gerados pelo Gemini

Passo 4: Acessar o Dashboard

1. Navegue até " Dashboard"
2. Visualize métricas consolidadas
3. Analise gráficos interativos

Passo 5: Gerar Relatórios

1. Navegue até " Relatórios"
2. Escolha entre:
 - **Excel**: Relatório consolidado de todas as notas
 - **PDF**: Análise detalhada de nota específica
3. Faça o download

Exemplo de Uso

Dados que o sistema extrai automaticamente:



python

- ✓ Chave de Acesso: 1234567890123456789012345678901234
- ✓ Emitente: Empresa Exemplo LTDA (CNPJ: 12.345.678/0001-90)
- ✓ Destinatário: Cliente Final (CPF: 123.456.789-10)
- ✓ Produtos: 5 itens identificados
- ✓ Impostos: ICMS, IPI, PIS, COFINS calculados
- ✓ Valor Total: R\$ 12.450,00

Validações realizadas:

- ✓ CNPJ do emitente válido
- ✓ CPF do destinatário válido
- ✓ Chave de acesso correta
- ✓ Cálculos de produtos conferidos
- ✓ Valor total consistente
- ⚠ Alerta: Valor elevado da nota

Análise de IA (Gemini):

"A nota fiscal apresenta conformidade fiscal adequada. O CFOP 5.102 indica venda de mercadoria adquirida de terceiros, apropriado para a operação. Os valores de ICMS estão calculados corretamente sobre a base de cálculo. Recomenda-se verificar o prazo de validade dos produtos perecíveis listados."

Validações Implementadas

1. Validação de Documentos

- **CNPJ:** Algoritmo de dígitos verificadores
- **CPF:** Algoritmo de dígitos verificadores
- **Chave de Acesso:** Validação do DV (44 dígitos)

2. Validação Matemática

- Soma de produtos vs total declarado
- Cálculo de impostos (ICMS, IPI, PIS, COFINS)
- Valor final da nota (produtos + frete + seguro + IPI - desconto)

3. Validação Inteligente (IA)

- Análise de conformidade fiscal (CFOP, NCM)
- Identificação de padrões suspeitos
- Recomendações contextualizadas
- Avaliação de risco fiscal

Relatórios Gerados

Excel (.xlsx)

- **Aba "Resumo":** Visão geral de todas as notas
- **Aba "Produtos":** Detalhamento de itens
- **Aba "Impostos":** Análise tributária

PDF (.pdf)

- Dados completos da nota fiscal
- Partes envolvidas (emitente/destinatário)
- Valores e impostos detalhados
- Resultado da validação com score de confiança
- Inconsistências, alertas e recomendações
- Análise inteligente completa da IA

Configurações Avançadas

Personalizar Validações

Edite `validator.py` para adicionar regras customizadas:



python

Exemplo: Validar alíquota de ICMS

```
def validar_aliquota_icms(self, produto):  
    if produto.impostos.icms_valor == 0:  
        return "ICMS zerado - verificar benefício fiscal"
```

Adicionar Novos Campos

Edite `models.py` para incluir campos adicionais:



python

```
class Produto(BaseModel):  
    # ... campos existentes  
    codigo_barras: Optional[str] = None # Novo campo
```

Troubleshooting

Erro: "API Key inválida"

- Verifique se a chave do Gemini está correta
- Certifique-se de que a API está ativada no Google Cloud

Erro: "Arquivo XML inválido"

- Confirme que o arquivo é um XML válido de NF-e
- Verifique se o namespace está correto

Erro: "Module not found"

- Execute: `pip install -r requirements.txt`
- Confirme que o ambiente virtual está ativado

Recursos Adicionais

- [Documentação Google Gemini](#)
- [Streamlit Docs](#)
- [Layout padrão NF-e \(SEFAZ\)](#)
- [Pydantic Docs](#)

Objetivos Educacionais

Este projeto foi desenvolvido para demonstrar:

1. **Integração de IA Generativa** em aplicações fiscais
2. **Automação de processos** contábeis e tributários

3. **Validação inteligente** de documentos complexos
 4. **Desenvolvimento Full-Stack** com Python
 5. **Boas práticas** de engenharia de software
-

Avisos Importantes

- **Este é um projeto educacional** - não substitui validações oficiais
 - Sempre consulte um contador para questões fiscais reais
 - Não utilize em ambiente de produção sem validações adicionais
 - Os resultados da IA são orientativos e podem conter erros
-

Licença

Este projeto é de código aberto para fins educacionais.

Desenvolvido para

Curso de IA e Automação Fiscal

Data de conclusão: 30 de outubro de 2025

Contribuições

Sugestões e melhorias são bem-vindas! Este é um projeto educacional em evolução.

Suporte

Para dúvidas sobre o projeto:

- Consulte a documentação inline no código
 - Verifique os comentários nos arquivos
 - Revise os exemplos de uso
-

Bons estudos e boa sorte no projeto!  