

Sistema de Consulta de Erros NFE - SEFAZ

Este sistema foi desenvolvido para auxiliar usuários que trabalham com NFE (Nota Fiscal Eletrônica) do SEFAZ a esclarecer dúvidas sobre os principais erros de processamento.

Arquitetura

O sistema utiliza a seguinte arquitetura:

- **FastAPI:** Interface web para acesso às funcionalidades
- **Python:** Linguagem de programação principal
- **ChromaDB:** Banco de dados vetorial para armazenamento da base de conhecimentos
- **LangChain:** Framework para integração com LLMs e agentes de IA
- **Perplexity AI:** LLM para busca e processamento de informações

Funcionalidades Implementadas

Função 1: Geração/Atualização da Base de Conhecimentos

Esta é a funcionalidade principal implementada, acessível através da **opção 1** do menu da aplicação.

Características:

1. Verificação e Criação do Banco de Dados:

- Verifica se o banco ChromaDB existe
- Cria um novo banco se não existir
- Conecta ao banco existente se disponível

2. Estrutura da Base de Dados:

- **Código do erro:** Código específico do erro NFE
- **Pergunta/Consulta:** Query original utilizada na busca
- **Resposta:** Conteúdo detalhado sobre o erro
- **Data da pesquisa:** Timestamp da coleta de informações
- **Fonte:** Origem da informação
- **Tipo:** Classificação do documento

3. Fontes de Informação:

- Sites oficiais do SEFAZ
- Receita Federal
- Outras fontes governamentais confiáveis
- Documentação oficial sobre NFE

4. Guardrails de Segurança:

- Rate limiting para evitar sobrecarga
- Validação de domínios permitidos
- Controle de acesso apenas a fontes oficiais
- Tratamento de erros robusto

Arquivos do Sistema

Arquivos Principais

1. `nfe_knowledge_base.py` : Módulo principal com a classe `NFEKnowledgeBase`
2. `nfe_updated.py` : Versão atualizada do módulo FastAPI com integração
3. `templates/gerar_base.html` : Interface web para a função de geração da base

Arquivos de Teste

1. `test_simple.py` : Teste básico da funcionalidade
2. `test_nfe_system.py` : Teste completo do sistema

Como Usar

1. Instalação das Dependências

Bash

```
pip install fastapi uvicorn langchain-chroma langchain-perplexity python-dotenv
```

2. Configuração

Certifique-se de que a chave da API Perplexity está configurada no código:

Python

```
self.perplexity_api_key = "sua-chave-aqui"
```

3. Execução

Teste Básico:

Bash

```
python test_simple.py
```

Uso da Função Principal:

Python

```
from nfe_knowledge_base import gerar_base_conhecimento

resultado = gerar_base_conhecimento()
print(resultado)
```

4. Integração com FastAPI

Para integrar com seu sistema FastAPI existente, substitua o arquivo `nfe.py` pelo `nfe_updated.py` e adicione:

Python

```
# No seu arquivo principal FastAPI
from nfe_updated import router as nfe_router

app.include_router(nfe_router)
```

Estrutura da Resposta

A função `gerar_base_conhecimento()` retorna um dicionário com:

JSON

```
{
  "status": "sucesso",
  "message": "Base de conhecimentos atualizada com sucesso",
  "detalhes": {
    "consultas_processadas": 10,
```

```
"buscas_bem_sucedidas": 8,  
"documentos_adicionados": 8,  
"total_documentos_final": 15,  
"erros": []  
}  
}
```

Endpoints da API

GET /gerar_base

Retorna a página HTML para geração da base

POST /gerar_base

Executa a geração/atualização da base de conhecimentos

GET /status_base

Retorna o status atual da base de dados

POST /consultar_base

Permite consultar diretamente a base de conhecimentos

Principais Erros NFE Cobertos

O sistema busca informações sobre os principais erros, incluindo:

- Erro 539: Duplicidade de NFe
- Erro 565: Erro de validação
- Erro 101: Erro de formato
- Erro 204: Erro de certificado
- Erro 215: Erro de schema
- E muitos outros...

Guardrails de Segurança

1. **Rate Limiting:** Máximo de 10 requisições por minuto
2. **Domínios Permitidos:** Apenas sites oficiais (.gov.br, sefaz, receita federal)
3. **Tratamento de Erros:** Logs detalhados e recuperação automática

4. **Validação de Entrada:** Sanitização de consultas e parâmetros

Logs e Monitoramento

O sistema gera logs detalhados para:

- Inicialização da base de dados
- Progresso das consultas
- Erros e exceções
- Estatísticas de uso

Próximos Passos

Para expandir o sistema, considere:

1. Implementar cache para consultas frequentes
2. Adicionar mais fontes de informação
3. Criar interface de administração
4. Implementar backup automático da base
5. Adicionar métricas de qualidade das respostas

Suporte

Para dúvidas ou problemas, verifique os logs do sistema e consulte a documentação oficial do SEFAZ.