Documentação - AIStockAgent

Fluxo Geral do Sistema

1. Inicialização: Configuração da API OpenAI

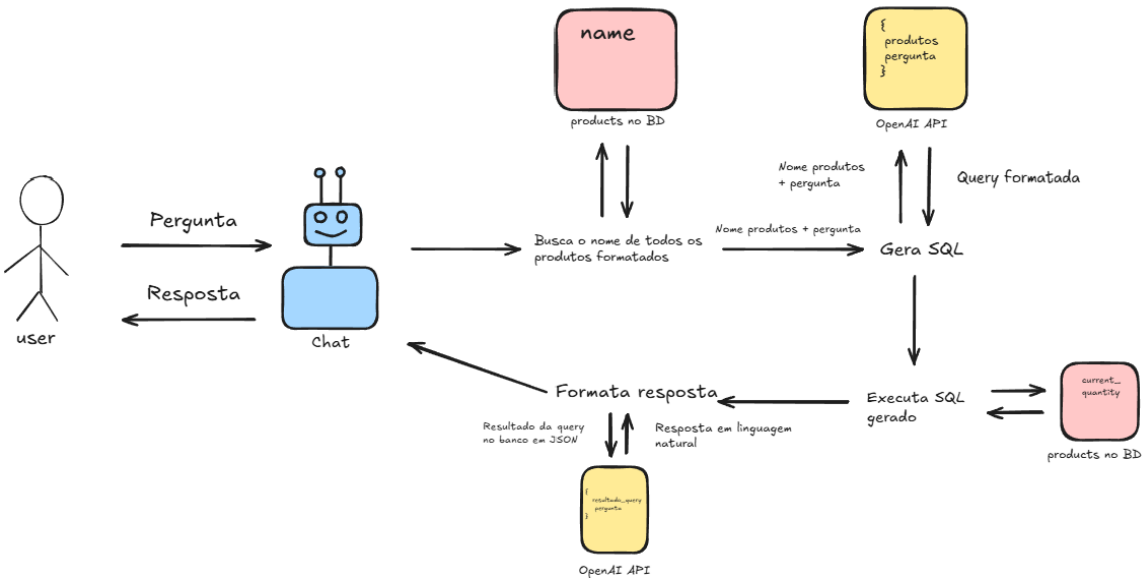
2. Contexto: Busca e formatação dos produtos disponíveis

3. Interpretação: Conversão da pergunta em SQL via IA

4. Execução: Consulta ao banco de dados

5. Formatação: Conversão dos resultados em resposta natural

6. Retorno: Entrega da resposta ao usuário



Dependências:

OpenAI SDK 1.54.3

SQLAlchemy 2.0.23

Streamlit (qualquer versão)

Visão Geral

A classe AIStockAgent implementa um agente de IA para consultas inteligentes de estoque, utilizando a API da OpenAI para interpretar perguntas em linguagem natural e convertê-las em consultas SQL.

Métodos da classe:

__init__(self)

Entrada: Nenhuma

Retorno: Nenhum (construtor)

Descrição: Inicializa o agente configurando a API da OpenAI. Obtém a chave da API através da variável de ambiente OPENAI_API_KEY e cria uma instância do cliente OpenAI.

Fluxo: Verifica se a chave da API está definida → Configura a chave globalmente → Cria cliente OpenAI

get_all_product_names(self) -> List[str]

Objetivo: listar produtos que já foram registrados no estoque. Isto é necessário, pois o usuário não irá digitar o produto com o mesmo nome que foi cadastrado no banco.

Entrada: Nenhuma

Retorno: `List[Dict[str, str]]`

O retorno é estruturado da seguinte forma:

```
[
{"brand": "Marca 1", "product": "Nome"},
```

```
{"brand": "Marca 2", "product": "Produto 1"} ]
```

Descrição: Busca todos os nomes distintos de produtos armazenados no banco de dados para uso como contexto nas consultas.

Fluxo: Abre sessão do banco → Executa query para nomes distintos → Extrai nomes dos resultados → Fecha sessão → Retorna lista

get_product_context(self) -> str

Objetivo: função auxiliar que chama o método `get_all_product_names()` para iterar na lista retornada e extrair somente os nomes

Entrada: Nenhuma

Retorno: str - String formatada com lista numerada dos produtos ou mensagem de estoque vazio

Descrição: Prepara um contexto textual formatado contendo todos os produtos disponíveis no estoque para fornecer à IA.

Fluxo: Obtém lista de nomes → Verifica se há produtos → Formata em lista numerada → Retorna contexto formatado

generate_sql_query(self, user_question: str, product_context: str) -> str

Entrada:

- user_question: str - Pergunta do usuário em linguagem natural

- product_context: str - Contexto dos produtos disponíveis

Retorno: str - Query SQL gerada pela IA

Descrição: Utiliza a API da OpenAI para converter uma pergunta em linguagem natural em uma query SQL válida para SQLite, considerando a estrutura da tabela de produtos.

Fluxo: Monta prompt do sistema com estrutura da tabela → Envia pergunta para GPT-3.5-turbo → Extrai query SQL da resposta → Retorna query

execute_query(self, sql_query: str) -> List[Dict[str, Any]]

Entrada:

- sql_query: str - Query SQL a ser executada

Retorno: List[Dict[str, Any]] - Lista de dicionários com os resultados da consulta

Descrição: Executa a query SQL no banco de dados e converte os resultados em formato de dicionário para facilitar o processamento.

Fluxo: Abre sessão do banco → Executa query SQL → Obtém colunas e linhas → Converte para dicionários → Fecha sessão → Retorna resultados

format_response(self, user_question: str, query_results: List[Dict[str, Any]], product_context: str) -> str

Entrada:

- user_question: str - Pergunta original do usuário

- query_results: List[Dict[str, Any]] - Resultados da consulta SQL

- product_context: str - Contexto dos produtos

Retorno: str - Resposta formatada em linguagem natural.

Descrição: Utiliza a API da OpenAI para converter os resultados brutos da consulta SQL em uma resposta clara e amigável em português brasileiro.

Fluxo: Monta prompt com contexto e resultados → Envia para GPT-3.5-turbo → Extrai resposta formatada → Retorna resposta em linguagem natural

process_question(self, user_question: str) -> str

Entrada:

- user_question: str - Pergunta do usuário sobre estoque

Retorno: str - Resposta final formatada ou mensagem de erro

Descrição: Método principal que orquestra todo o processo de resposta a uma pergunta sobre estoque, integrando todos os outros métodos da classe.

Fluxo: Obtém contexto dos produtos → Gera query SQL → Executa query → Formata resposta → Retorna resposta final