Guia pratico: uso da OpenAI em um projeto Python

1) Chaves e seguranca
- Guarde a OPENAI_API_KEY em variavel de ambiente (.env) e nunca comite no git.
- Em runtime, valide se a chave existe e, se nao existir, solicite o input no terminal.

2) Modelos e custo
- Para embeddings, use um modelo de embeddings e mantenha o mesmo na ingestao e na busca.
- Para respostas (LLM), escolha um modelo adequado ao tamanho do contexto e latencia desejada.
- Controle custos limitando o tamanho do contexto e o numero de documentos recuperados.

3) Embeddings e RAG
- Extraia texto do PDF, divida em chunks (por exemplo 1000 caracteres com overlap 150).
- Gere embeddings de cada chunk e armazene em um banco vetorial (por exemplo Postgres com pgvector).
- Na pergunta, gere embedding da query e recupere os top-k mais similares (k=10).
- Monte o prompt com CONTEXTO e REGRAS para responder somente com base no contexto recuperado.

4) Qualidade e confiabilidade
- Use temperatura baixa para reduzir alucinacao.
- Logue k, scores e paginas para depurar recuperacao.
- Se o contexto nao contiver a informacao, responda com uma mensagem padrao de falta de dados.

5) Operacao
- Defina timeouts de rede e trate erros de API e de conexao com o banco.
- Rode a ingestao antes do chat e mantenha uma rotina para resetar a colecao em desenvolvimento.

Boas praticas adicionais: mantenha prompts versionados, use validacao de entrada, e monitore latencia e taxa de erros para evoluir o sistema com seguranca.

Boas praticas adicionais: mantenha prompts versionados, use validacao de entrada, e monitore latencia e taxa de erros para evoluir o sistema com seguranca.

Boas praticas adicionais: mantenha prompts versionados, use validacao de entrada, e monitore latencia e taxa de erros para evoluir o sistema com seguranca.

Boas praticas adicionais: mantenha prompts versionados, use validacao de entrada, e monitore latencia e taxa de erros para evoluir o sistema com seguranca.

Boas praticas adicionais: mantenha prompts versionados, use validacao de entrada, e monitore latencia e taxa de erros para evoluir o sistema com seguranca.

Boas praticas adicionais: mantenha prompts versionados, use validacao de entrada, e monitore latencia e taxa de erros para evoluir o sistema com seguranca.

Boas praticas adicionais: mantenha prompts versionados, use validacao de entrada, e monitore latencia e taxa de erros para evoluir o sistema com seguranca.

Boas praticas adicionais: mantenha prompts versionados, use validacao de entrada, e monitore latencia e taxa de erros para evoluir o sistema com seguranca.

Boas praticas adicionais: mantenha prompts versionados, use validacao de entrada, e