

ETAPA 7

Aplicações práticas de LLM I



Gerenciamento de chaves de API para segurança de dados

Application Programming Interface (API): credenciais exclusivas utilizadas para acessar de forma segura serviços e recursos. Funcionam como uma senha digital, garantindo que apenas usuários ou sistemas autorizados possam interagir com a API.

- Acessar serviços de modelos de linguagem para geração de texto.
- Implementar integrações em aplicações, como chatbots ou sistemas de análise de dados.
- Monitorar o uso de recursos da API para controlar custos e limitar abusos.

Deputados

GET **/deputados** Listagem e busca de deputados, segundo critérios

GET **/deputados/{id}** Informações detalhadas sobre um deputado específico

GET **/deputados/{id}/despesas** As despesas com exercício parlamentar do deputado

GET **/deputados/{id}/discursos** Os discursos feitos por um deputado em eventos diversos

GET **/deputados/{id}/eventos** Uma lista de eventos com a participação do parlamentar

GET **/deputados/{id}/frentes** As frentes parlamentares das quais um deputado é integrante

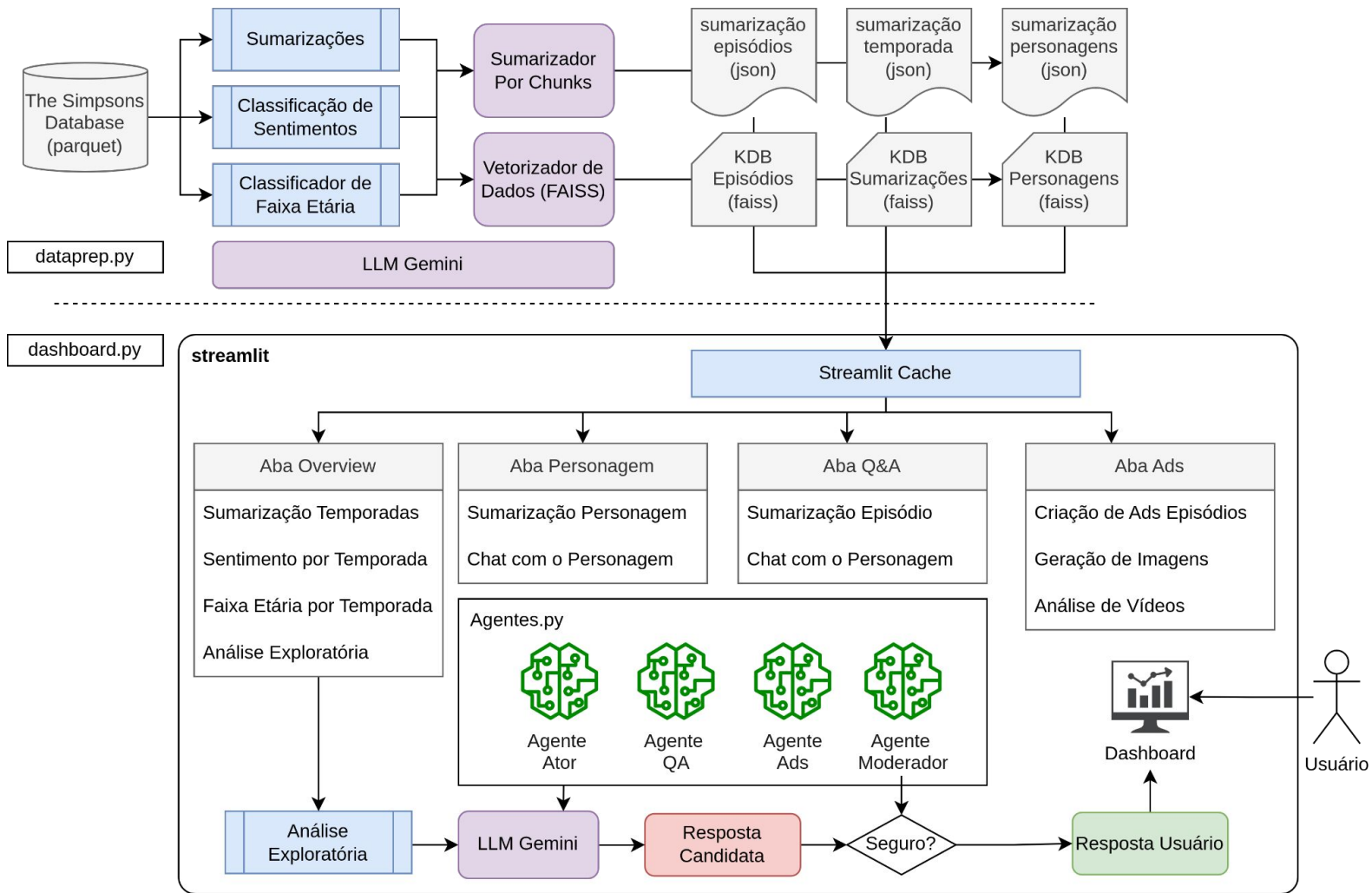


Gerenciamento de chaves de API para segurança de dados

Para segurança, é essencial seguir as práticas recomendadas:

- Criptografar as chaves durante o armazenamento e transmissão.
- Usar ferramentas como arquivos .env para evitar a exposição de chaves no código.
- Limitar os acessos definindo escopos específicos, como permissões de leitura ou escrita.
- Monitorar e rotacionar chaves regularmente para evitar usos indevidos.

```
from dotenv import load_dotenv
# Cada um deve criar o seu proprio .env na raiz do repositório
# Esse .env deve ter os pares de chave=valor
# GEMINI_KEY
# OPENAI_KEY
load_dotenv('../.env')
```





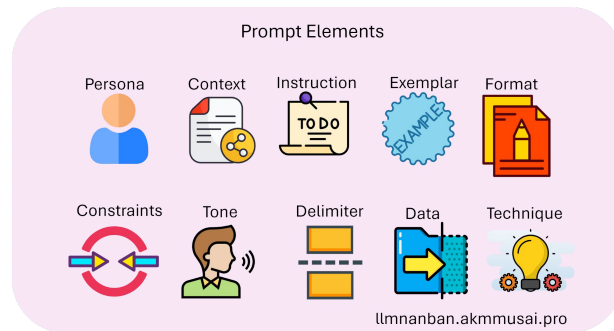
Resumir vídeos com Gemini

dataprep.py:

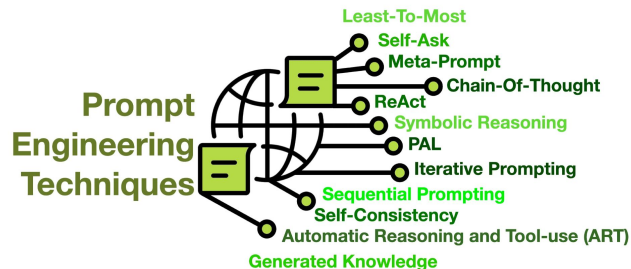
- **Sumarização de episódios:** Utilizar técnicas de prompt engineering para sumarizar um episódio do programa.
- **Sumarização da temporada:** Quebrar a temporada em chunks de episódios para sumarização dividida.

dashboard.py:

- **Análise Exploratória da Estatística dos Episódios:** Utilizar técnica de prompt chaining para implementar uma análise exploratória dos dados de audiência, personagens e reviews dos episódios e temporadas.
- **Generated Knowledge para gerar Insights:** Usar o LLM como um cientista de dados para gerar insights de acordo com a estatística e análises realizadas na base de dados.



12 Prompt Engineering Techniques





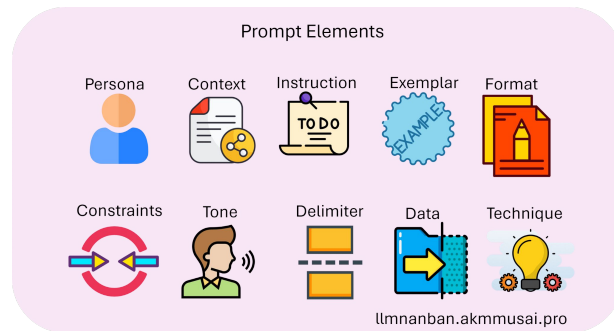
Gerador de Notícias com Gemini

dataprep.py:

- **Sumarizador de personas:** Utilizar técnicas de chunks para consolidar as características do personagem segundo suas falas.

dashboard.py:

- **Meta prompting para agentes personalizados:** Criar Agente Ator de IA que possa interpretar os personagens na interação com o usuário.
- **Assistente de Marketing:** geração de ads para campanha de promoção do show!



12 Prompt Engineering Techniques

