**Desenvolvimento de um Assistente Virtual com Arquitetura RAG para Orientação em Desastres Naturais**

**Descrição:**

Nesta GS, vocês deverão projetar e implementar um **assistente virtual inteligente** capaz de fornecer respostas **personalizadas, atualizadas e contextualizadas** sobre **desastres naturais** e as **melhores práticas de segurança e resposta**.

Para isso, o assistente deverá ser desenvolvido utilizando a **arquitetura RAG (Retrieval-Augmented Generation)**, combinando:

* **Modelos de linguagem da OpenAI (via API)**, responsáveis pela geração das respostas em linguagem natural.
* Um mecanismo de **busca e recuperação de informações atualizadas** de bases externas (ex.: notícias, bases de dados de abrigos, centros de ajuda, manuais de proteção civil, etc.).

O assistente deve ser capaz de atender aos seguintes perfis de usuários:

1. **Vítimas** diretamente afetadas pelo desastre.
2. **Moradores** da região em risco ou já afetada.
3. **Familiares das vítimas**, em busca de informações e orientações.

**Objetivos:**

* Entender e aplicar a **arquitetura RAG** para enriquecer a capacidade de resposta de assistentes virtuais com **informações externas e atualizadas**.
* Integrar a **API da OpenAI** com mecanismos de **busca e recuperação de informações**.
* Desenvolver habilidades de **engenharia de prompts** (Prompt Engineering) para garantir que o assistente gere respostas **precisas, empáticas e adequadas** aos diferentes perfis de usuário.
* Refletir sobre os **impactos sociais e éticos** do uso de assistentes virtuais em contextos críticos e de risco.

**Etapas Recomendadas:**

**1. Definição de Perfis de Usuário e Casos de Uso:**

* Definam claramente como o assistente deve se comportar diante de cada perfil:
  + Vítimas: fornecer orientações imediatas de segurança, rotas de evacuação, primeiros socorros.
  + Moradores: alertas preventivos, como se preparar, onde buscar ajuda.
  + Familiares: canais de contato, localização de abrigos, procedimentos de busca.

**2. Estruturação da Arquitetura RAG:**

**Retrieval (Recuperação):**

* Escolham fontes de informação confiáveis: bancos de dados locais, APIs de alertas de desastres, notícias atualizadas.
* Implementem um mecanismo para buscar e **recuperar os documentos** ou dados mais relevantes com base na consulta do usuário.  
  Exemplos: ElasticSearch, Weaviate, FAISS ou mesmo um sistema simples com busca semântica.

**Augmented Generation (Geração Aumentada):**

* Estruturem prompts que combinem:  
  a) A consulta do usuário.  
  b) As informações recuperadas.
* Usem a **API da OpenAI** para gerar a resposta final, baseada não apenas nos dados internos do modelo, mas também nas **informações atualizadas**.

**3. Engenharia de Prompt (Prompt Engineering):**

* Criem templates de prompt que garantam que o modelo gere respostas:
  + **Claras** e objetivas.
  + **Empáticas**, reconhecendo a situação emocional do usuário.
  + **Responsáveis**, indicando procedimentos corretos.

Exemplo de template:

"Como assistente especializado em resposta a desastres, utilizando as seguintes informações recuperadas: [contexto recuperado], oriente o usuário que relatou: '[consulta do usuário]'."

**4. Desenvolvimento da Interface:**

* Criem uma interface simples dentro do próprio colab usando a função input, ou web do chatbot para que usuários possam:
  + Selecionar o tipo de perfil.
  + Fazer suas perguntas.
  + Receber a resposta gerada.

Se quiserem, podem utilizar ferramentas como **Streamlit**, **Gradio**.

**5. Testes e Avaliação:**

* Realizem testes simulando diferentes perfis e situações: enchentes, incêndios, terremotos etc.
* Avaliem a qualidade das respostas geradas:
  + Aderência à informação recuperada.
  + Clareza e adequação emocional.
  + Correção e utilidade prática.

**6. Reflexão Crítica:**

* Discutam os desafios e limites desse tipo de assistente:
  + O risco de **informações incorretas**.
  + A necessidade de fontes sempre **atualizadas e verificadas**.
  + A importância de uma linguagem **empática** e não alarmista.

**Exemplo de Fluxo de Uso do Assistente:**

**Usuário:** “Estou preso no segundo andar de casa durante uma enchente, o que devo fazer?”  
**Recuperação:** Informações locais sobre níveis de água e rotas de evacuação.  
**Resposta gerada:** “Entendo sua situação. De acordo com os dados mais recentes, o nível da água continua subindo. Se for seguro, tente mover-se para um ponto mais alto e sinalizar sua posição com objetos visíveis. Não tente atravessar águas profundas. As equipes de resgate estão atuando na sua região, mantenha-se calmo.”

**Entrega Esperada:**

**Relatório Técnico** contendo:

* Descrição da arquitetura implementada.
* Fontes de informação utilizadas.
* Exemplos de prompts e respostas.
* Reflexão crítica sobre o sistema.

**Código-fonte comentado**: Preferencialmente em **notebook Jupyter** ou **Google Colab**.

(Opcional) **Demonstração** do assistente via link ou vídeo.

**Critérios de Avaliação:**

* Correção e clareza na aplicação da arquitetura RAG.
* Efetividade da integração entre recuperação e geração.
* Qualidade das respostas (clareza, adequação, empatia).
* Reflexão crítica sobre o uso ético e responsável.