



Teste vaga de Python com foco em LLM | Alura

Olá! Ficamos bastantes felizes por você ter se interessado em trabalhar conosco e ter chegado nessa parte da etapa.

Nessa etapa, sua tarefa será realizar o protótipo de uma aplicação web para um cenário que pensamos. É esperado - e parte do processo - que algum requisito possa gerar dúvidas. Recomendamos que você anote o que não ficou claro e explique o porquê de ter seguido pela decisão que você seguiu tendo apenas o contexto desse desafio em mente.

Tecnologias que devem ser utilizadas

- Python (recomendamos 3.10+)
- Flask
- SQL (recomendamos MySQL ou PostgreSQL, mas fique à vontade para usar o que for mais confortável)
- LLMs
- Git & Github

Tecnologias opcionais

- Langchain

Avaliação

O que iremos avaliar durante o processo:

- Sua capacidade técnica
 - Linguagem
 - Framework
 - Uso das LLMs
- Sua linha de pensamento ao desenvolver as funcionalidades pedidas
- Sua capacidade de derivar e entender requisitos
- Sua capacidade de organização
 - Organização de código
 - Divisão de arquivos
 - Nomenclatura
 - Commits coesos
 - Organização do repositório (README)

Para a entrega

Para a entrega, seu desafio deverá estar em um repositório hospedado no Github. Recomendamos um repositório público mas, caso prefira criar privado, pedimos que adicione as seguintes contas como colaboradores - MurilloGodoi, lucasboot e ebsiqueira.

Seu repositório deverá conter um README explicando como executar o seu projeto. Fique à vontade para incluir informações que você achar pertinente.

Sobre o desafio

Cenário

A AluMind é uma startup que oferece um aplicativo focado em bem-estar e saúde mental, proporcionando aos usuários acesso a meditações guiadas, sessões de terapia, e conteúdos educativos sobre saúde mental. Com o alto crescimento da base de usuários, a AluMind está começando a ter desafios para analisar feedbacks vindo

...entando com o feedback e as sugestões para analisar feedbacks vindos dos usuários em diferentes plataformas (canais de atendimento ao cliente; comunidades no Discord; redes sociais). Portanto, nessa tarefa, você criará uma aplicação cuja responsabilidade seja de analisar os feedbacks vindos dos usuários, classificá-los a partir do seu sentimento e elencar as possíveis melhorias contidas neles.

1. Classificação de Feedbacks

Nessa tarefa, você pode considerar que outro time de desenvolvimento ficará responsável por agrupar e enviar os feedbacks para sua aplicação.

Cada Feedback é composto por um identificador e por um texto, que é o feedback em si próprio.

Você deve criar um endpoint em sua aplicação que receba os feedbacks e classifique-os a partir do seu sentimento.

Exemplo de requisição:

```
POST /feedbacks Content-Type: application/json { "id":  
  "4042f20a-45f4-4647-8050-139ac16f610b", "feedback":  
  "Gosto muito de usar o Alumind! Está me ajudando  
  bastante em relação a alguns problemas que tenho. Só  
  quer ia que houvesse uma forma mais fácil de eu mesmo  
  realizar a edição do meu perfil dentro da minha conta" }
```

Exemplo de resposta:

```
{ "id": "4042f20a-45f4-4647-8050-139ac16f610b",  
  "sentiment": "POSITIVO", "requested_features": [ {  
    "code": "EDITAR_PERFIL", "reason": "O usuário gostaria  
    de realizar a edição do próprio perfil" } ] }
```

Você não precisa se preocupar com autenticação/autorização nesse momento.

Cada feedback deve ser marcado como “POSITIVO”, “NEGATIVO” e “INCONCLUSIVO”. Além disso, cada feedback contém possíveis funcionalidades sugeridas. Cada funcionalidade sugerida tem um

código que a identifica unicamente e uma descrição do porquê a funcionalidade é importante.

2. Relatório

Você deve disponibilizar uma página web que forneça um relatório simples do andamento de todos os feedbacks dados até o momento. Não precisa se preocupar com autenticação/ autorização nesse momento. O relatório pode ser uma tabela simples.

Algumas coisas que gostaríamos de ter acesso fácil ao olhar o relatório:

- Qual a porcentagem de feedbacks positivos em relação aos demais?
- Quais são as features mais pedidas que conseguimos perceber através de feedbacks?

Além disso, gostaríamos de alguma forma conseguir acesso mais detalhado à feedbacks específicos, caso a gente julgue necessário.

3. Resumo Semanal

Ao final de cada semana, um email deve ser enviado para alguns stakeholders do AluMind com um breve resumo dos principais feedbacks da semana. O email deve ter as seguintes informações sobre os feedbacks da dada semana:

- % de feedbacks positivos
- % de feedbacks negativos
- Principais funcionalidades pedidas e o porquê cada uma seria importante de ter.

Ganha pontos se o texto do email for gerado com a ajuda de alguma LLM

4. Bônus (não obrigatório)

Implementação de um sistema de filtragem no endpoint de envio de feedbacks que assegure que apenas feedbacks legítimos e não

classificados como spam sejam processados e armazenados.