

Especificação de projeto

Este documento descreve quais serão as funcionalidades mínimas do sistema. O projeto possui um tema geral a partir do qual os alunos devem definir um tema específico.

O tema geral para o projeto é: integração com o ChatGPT.

A aplicação desenvolvida se integrará com o ChatGPT por meio de sua API e proverá alguma funcionalidade de interesse para as pessoas comuns. Veja alguns exemplos:

<https://platform.openai.com/examples>

Esses são apenas alguns exemplos. Seja criativo! Seu tema não pode ser igual a nenhum daqueles existentes na página de exemplos do ChatGPT.

Instruções

1. O projeto pode ser desenvolvido por grupos de até nove alunos.
2. O Front End deverá ser uma aplicação React.
3. O Front End deverá dar acesso gráfico a uma ou mais funcionalidade(s) obtida(s) por meio do acesso ao ChatGPT.
4. A comunicação entre Front End e ChatGPT deve ser feita utilizando-se requisições HTTP com o pacote axios. Haverá um Back End simples feito com NodeJS que operará como “intermediário” entre Front End e ChatGPT.
5. Cada requisição pelo Front deverá ser armazenada, junto com a resposta produzida pelo ChatGPT, em uma tabela de logs gerenciada por um SGBD relacional: MySQL ou PostgreSQL. Sugestão: Use o serviço aiven.io para hospedar a sua base de dados em nuvem.
6. Os alunos deverão montar os seus grupos, escolher o seu tema específico e preencher o formulário a seguir até o dia **08/09/2024**. O não preenchimento implica na perda da nota do projeto. Basta que somente um aluno faça o preenchimento em nome de seu grupo.

https://docs.google.com/forms/d/e/1FAIpQLSfQFQj_X_P2lhTdpA7u8B0xjRa6FnEbizSAsdgRSe2Ko7meeQ/viewform

7. A entrega do projeto deverá ser feita por meio do Github. Para cada entrega, os grupos devem gerar uma release (criar uma tag e fazer upload com git push) do Github.

OBS: Crie um arquivo no repositório chamado README.md. Ele deve conter os RAs e os nomes completos sem abreviações e em ordem alfabética de todos os integrantes.

OBS: Ainda que uma entrega parcial não esteja pronta para uma determinada data, gere uma release da mesma forma até a respectiva data prevista. Na descrição de cada release, explique o que foi implementado e o que deixou de ser. O registro de entregas respeitando as datas será fundamental na hora da correção feita pelos professores.

OBS: As releases ficarão registradas no seu repositório. Por isso, não há a necessidade de realizar uma nova entrega a cada data a seguir. Basta apenas fazer um novo git push com a respectiva tag. Pelo fato de ter entregue o link do repositório, os professores terão acesso às releases.

Cronograma de entregas

Data	Conteúdo	Rótulo da tag	Mensagem da tag
22/09	Front End. Esqueleto inicial da aplicação React criado com npx, como na aula.	v0.0.1	entrega01(front): cria a aplicacao React
29/09	Front End. Início da tela principal da aplicação utilizando, pelo menos, dois componentes React.	v0.0.2	entrega02(front): atualiza interface principal
13/10	Front End. Interface gráfica principal completa. Eventual implementação de outras telas, caso existam.	v1.0.0	entrega03(front): finaliza interface gráfica
20/10	Back End. Um endpoint GET /hello-world. Serviço em execução na porta 3000.	v1.0.1	entrega04(back): cria a aplicação NodeJS
27/10	Back End. Script para criação de base de dados relacional (tabela de logs).	v1.0.2	entrega05(back): cria a tabela de logs
03/11	Back End. Um endpoint GET /consultar que se conecta ao SGBD e registra a sua execução na tabela de logs.	v1.1.0	entrega06(back): registra logs na base
10/11	Back End. Endpoint GET /consultar conversa com o ChatGPT.	v1.2.0	entrega07(back): faz back se integrar com API do ChatGPT
17/11	Front End. Aplicação React integrada ao Back End. Eventuais ajustes das entregas passadas.	v2.0.0	entrega08(front): integra front e back

Semana de pré-banca: 18/11 a 22/11