

Prosel 2025.2

Desafio Técnico - Trilha Front-End

App de Trivia

1. Problema

Para aumentar o engajamento de usuários em nossas plataformas digitais, a Ecomp Jr. quer trazer experiências interativas e divertidas para o usuário. Como primeiro passo, iremos desenvolver uma aplicação de Trivia (Quiz) que servirá como um produto mínimo viável (MVP) para futuras iniciativas de gamificação.

Como futuro(a) desenvolvedor(a) Front-end da Ecomp Jr., seu papel será construir a interface para esta aplicação de Trivia. O objetivo é criar uma experiência de usuário fluida e agradável, desde a configuração do quiz até a exibição dos resultados, consumindo dados de uma API externa (Open Trivia DB).

2. Requisitos

a. Tecnologias:

- i. Utilizar React, Vite e Tailwind CSS

b. Comunicação com a API:

- i. Utilizar a biblioteca Axios para realizar as requisições à API do Open Trivia DB (<https://opentdb.com/api.php>).

c. Tela de Configuração do Quiz:

A tela inicial deve permitir que o usuário configure as perguntas que deseja responder. Esta tela deve conter os seguintes campos:

- i. Number of Questions: Um campo para definir a quantidade de perguntas.
- ii. Select Category: Um seletor para escolher a categoria das perguntas.
- iii. Select Difficulty: Um seletor para o nível de dificuldade (Easy, Medium, Hard).
- iv. Select Type: Um seletor para o tipo de questão (Multiple Choice, True/False).
- v. Um botão para "Iniciar Quiz" que, ao ser clicado, busca as perguntas na API com os parâmetros selecionados e navega para a tela do quiz.

d. Tela do Quiz:

- i. Após iniciar o quiz, o usuário deve ser apresentado a uma tela onde responderá às perguntas sequencialmente.
- ii. O usuário deve conseguir selecionar uma resposta.
- iii. Deve haver um mecanismo para avançar para a próxima pergunta.

- iv. Exibir um contador de progresso (ex: "Questão 3 de 10").
 - v. Tela de Resultados: Ao final de todas as perguntas, o usuário deve ser levado a uma tela de resultados que exibe a pontuação final (ex: "Você acertou 7 de 10 perguntas").
 - vi. Ao final do quiz, deve ser apresentado um botão de "Jogar Novamente" que retorna o usuário para a Tela de Configuração.
- e. **Documentação (README.md):**
O repositório do projeto deve conter um arquivo README.md com instruções claras sobre como instalar as dependências e executar o projeto localmente.
- f. **Commits Padronizados no GitHub:**
Utilizar uma convenção de commits padronizada no GitHub, como o uso de mensagens descritivas e significativas para cada commit, facilitando a compreensão e o acompanhamento do progresso do projeto. ([Não sabe o que é?](#))

3. Diferenciais

- a. **Componentes com shadcn/ui:** Utilizar a biblioteca shadcn/ui para construir os componentes da interface (inputs, seletores, botões, cards, etc.), conforme recomendado.
- b. **Responsividade:** Garantir que a aplicação seja funcional e visualmente agradável em diferentes tamanhos de tela (desktop e mobile).
- c. **Feedback Visual:** Fornecer feedback instantâneo ao usuário ao responder uma pergunta (ex: destacar a resposta correta em verde e a incorreta em vermelho).
- d. **Prototipação da Interface:** Criar um protótipo visual da aplicação utilizando o Figma. Caso feito, o link de visualização do protótipo deve ser incluído no README.md do projeto.

4. Produto

Espera-se que o produto final seja uma aplicação web completa, bem documentada e eficiente para a finalidade de Trivia. A interface deve ser intuitiva e robusta, proporcionando uma experiência de usuário agradável.

5. Entrega

O prazo para entrega do produto vai até o dia 13/10/2025, via e-mail (ecompjr@uefs.br) contendo o link do repositório do projeto no GitHub.

6. Barema

A nota final será calculada pela soma dos pontos obtidos em cada critério de avaliação, conforme definido no Barema. O resultado será um valor entre 0 e 10, com a pontuação máxima limitada a 10. Caso a somatória de um candidato exceda esse valor, como na pontuação máxima possível de 11, sua nota final será registrada como 10.

Critério	Descrição do critério	Nota
Tela de Configuração	Avalia a correta implementação do formulário para configurar e iniciar o quiz.	De 0 a 3
Tela de Quiz e Lógica	Avalia a exibição das perguntas, a lógica de respostas e a navegação entre as questões.	De 0 a 3
Tela de Resultados	Avalia a exibição da pontuação final e a funcionalidade para jogar novamente.	De 0 a 2
Qualidade do Código	Avalia a organização do projeto (componentização), clareza e aplicação de boas práticas de React.	De 0 a 1
Commits Padronizados	Avalia o uso de uma convenção de commits.	De 0 a 1
Diferenciais (extra)	Avalia a implementação de um ou mais dos diferenciais propostos.	De 0 a 1

7. Dicas e Recomendações

Sabemos que desenvolver uma interface trivial é algo complexo e trabalhoso. Por isso, separamos algumas dicas e recomendações que podem ajudar a desenvolver o sistema:

1. **Documentação Oficial:** A documentação do React, Vite e Tailwind CSS são seus maiores aliados. Consulte-as sempre que tiver dúvidas.
2. **Componentes Prontos:** A recomendação de usar shadcn/ui é para agilizar o desenvolvimento da UI, permitindo que você foque mais na lógica da aplicação.
3. **API do Trivia:** Explore a documentação da [Open Trivia DB](#) para entender como montar a URL de requisição com os parâmetros do formulário.
4. **Ferramentas de Desenvolvimento:** Utilize as ferramentas de desenvolvedor do seu navegador para inspecionar elementos, testar a responsividade e depurar o código.