

Tech Challenge CEFIS 🚀

Bem-vindo(a) ao Tech Challenge da CEFIS.

Neste desafio você irá construir uma versão simplificada de uma parte extremamente importante de uma LXP (Learning Experience Platform).

Caso tenha alguma dúvida, nós estamos disponíveis no e-mail ti@cefis.com.br.

Bom desafio!

1. Contexto

O professor é uma figura central em plataformas inteligentes de aprendizagem. Alguns dos fluxos que ele está envolvido são a produção de conteúdo e o atendimento a dúvidas dos alunos.

Um exemplo de conteúdo é um curso, que sempre tem um professor vinculado. Um aluno que assiste o curso de um instrutor, pode ficar com dúvida em algum ponto apresentado. Ele pode então enviar sua dúvida ao professor, o qual lhe dará a resposta.

Nota: um professor só pode responder a uma pergunta vinculada a um curso que ele é o professor. Um aluno pode enviar no máximo duas dúvidas por curso.

2. Requisitos

Você deve criar uma API REST em Node com os seguintes requisitos:

1. Operações CRUD para o usuário e para o curso.
 - a. O usuário deve conter as informações de id, nome e tipo (aluno ou professor)
 - b. O curso deve conter as informações de id, título, duração e professor
2. O aluno deverá poder cadastrar uma pergunta em um curso.
 - a. A pergunta deve conter as informações de id, curso, aluno e texto da pergunta
3. O professor deverá poder cadastrar uma resposta a uma pergunta.
 - a. A resposta deve conter as informações id, pergunta e texto da resposta

Também deve ser feito um front-end utilizando ReactJS, preferencialmente com NextJS onde será exibido essas informações, possibilitando fazer todos os

processos diretamente na aplicação front-end. Lembrando que a listagem principal dos cursos deverá ter paginação.

Achou muito fácil? Então deixe sua estrela brilhar 🌟👤👤👤

Se os requisitos acima forem muito simples para você, aqui vai uma parte um pouco mais complexa para você se aventurar. Ah, esta parte é **opcional**.

Falando um pouco mais sobre contexto...

Depois de fazer um excelente trabalho de criação de conteúdo e atendimento de alunos, o professor recebe além do reconhecimento pessoal, sua remuneração em dinheiro. Ele recebe R\$300,00 por hora de curso disponibilizado e R\$15,00 por pergunta respondida. (valores fictícios)

4. O sistema deverá conter mais um tipo de usuário, o “admin”. Esse usuário poderá registrar pagamentos para o professor
5. Sempre que um professor receber um pagamento, ele deve receber um e-mail informativo
6. O sistema deverá permitir que o professor consulte seu saldo a receber, já descontados os pagamentos recebidos

3. Restrições

1. O serviço deve ser escrito em Node.js e React
2. O serviço deve armazenar informações em um banco de dados. Você pode escolher o banco que achar melhor. Aqui na CEFIS usamos amplamente MySQL
3. O projeto deve ter um README.md com todas as instruções sobre como executar e testar o projeto e os serviços disponibilizados. O README.md deve conter um link para utilizar o serviço hospedado.
4. O código deve ser versionado com git e armazenado em algum repositório remoto.

4. Tenha em mente

As pessoas que vão usar, precisam saber como elas funcionam e você precisa saber facilmente se sua API está com problemas, certo? (hint: docs)

5. Host it!

O mundo não roda somente em localhost e sua API também não deveria. Use alguma das muitas opções de hosts gratuitos de aplicações (Heroku, Vercel, etc. Hint: <https://free-for.dev>)

6. Entrega

O desafio deve ser entregue enviando o link do repositório através do email ti@cefis.com.br com o assunto [Tech Challenge]-{Seu Nome}. Lembre-se de deixar o repositório público.

Achamos que **1 semana** é um tempo ok para fazer o desafio, mas sabemos que nem todo mundo tem o mesmo nível de disponibilidade. Portanto, nos avise se precisar de mais tempo, ok?

7. Considerações Finais

Não se deixe intimidar, faça até onde conseguir! Te chamaremos para conversar depois de analisar.

Boa sorte :)