



Teste de Programação

- Este desafio consiste de várias partes. Você **não precisa** submeter todas elas para seguir adiante no processo, mas é claro que pedimos para que você dê o seu melhor.
- O seu tempo máximo para submissão é de 72 horas. Se você realmente precisar de mais tempo ou realmente quiser mais tempo para desenvolver algo a mais, não hesite em entrar em contato.
- Você será avaliado pelo escopo do seu código (ele resolve todos os problemas que você falou que resolve?), pelo design (qualidade do código), bugs, e também eficiência
- Se não houverem restrições especificadas, significa que é livre (exemplo, linguagem de programação, formato da interface para o usuário, etc)
- Quaisquer dúvidas entre em contato com recrutamento@voxus.tv
- Este teste contém 3 páginas além desta capa
- Dica: leia tudo antes de começar!

Dashboard de Tasks

Uma das funções importantes dos programadores na Voxus é desenvolver ferramentas internas para os times da empresa serem mais eficientes. A cada dia encontramos novas maneiras de aumentar a nossa produtividade. O time de devs tem um papel fundamental nisso. Recentemente, notamos que a maneira como essas ferramentas eram solicitadas para os devs era extremamente desorganizada e ineficiente. Por conta disso, desenvolvemos um "Dashboard de Tasks" para nos ajudar. Sua tarefa é implementar algumas partes dessa ferramenta.

Parte 1: CRUD

Implemente uma aplicação web [CRUD](#) para essas **tasks**, onde cada **task** consiste apenas de um título. Em outras palavras, sua aplicação deve suportar:

- criação de **task**
- deleção de **task**
- edição de **task**
- leitura de **task**

A interface para o usuário não precisa ser sofisticada, apenas funcional e clara.

Parte 2: Login Autenticado com o Google

Integre o dashboard com o login autenticado do Google. Em outras palavras, exija do usuário que ele logue com uma conta Google para ter acesso ao dashboard desenvolvido na Parte 1.

Parte 3: Tasks Mais Complexas

Faça com que as **tasks** suportem, pelo menos:

- nome da **task** (já existia na Parte 1)
 - serve como título para se referir à **task**
- descrição da **task**
 - importante para sabermos o que deve ser feito, porque é necessária, quais os requisitos, etc.
- quantidade indefinida de anexos
 - permite que o usuário possa anexar documentos que podem servir como suporte/evidência/guia para desenvolvimento da **task**
- prioridade da **task** (1 a 5)
 - importante para saber como alocar o tempo dos devs
- usuário que submeteu a **task**
 - importante para saber com quem se comunicar

Parte 4: Tasks Completas

Permita que os usuários possam marcar as **tasks** como "**done**". Depois, faça com que cada **task** indique quem foi o usuário que marcou-a como "**done**".

Parte 5: Suporte Robusto para Anexos + Busca

Atualize a plataforma implementada para obedecer o seguinte fluxo:

- **task** é submetida utilizando o dashboard
- **task** vai para uma fila F
- consumer de F faz upload dos anexos correspondente à **task** no [Amazon S3](#)
- status da **task** no dashboard é atualizado para "**processada**"
- quando desejado, usuário marca **task** como "**done**" pelo dashboard
- toda **task** marcada "**done**" é indexada no [Elastic Search](#) para buscas futuras

Submissão:

- Link com acesso para código no repositório GIT ou equivalente
- Instruções para testarmos o seu código (ele deve rodar pelo menos localmente!)
- Arquivo no repositório reportando, **para cada parte desenvolvida**:
 - quanto tempo você demorou
 - as assumptions que você fez para o desenvolvimento, se alguma
 - as principais dificuldades e desafios
- Credenciais de leitura da [Amazon S3](#) (se for o caso)