

Desafio Técnico

API de Gestão de Clínica

Objetivo

Na Capim, trabalhamos com clínicas odontológicas e seus respectivos dentistas. A proposta deste desafio é desenvolver uma aplicação em Go que permita o gerenciamento de registros relacionados a esses conceitos.

Nesse contexto, seu objetivo é criar uma API para a gestão de clínicas (que chamaremos de Sellers), possibilitando a criação de uma ou mais clínicas, bem como a administração de seus dentistas e contas bancárias associadas.

💡 *Cumpra os requisitos, mas não tenha medo de ir além. Este desafio é o momento de mostrar seus conhecimentos e criatividade.*

Sobre o Uso de IA

O uso de ferramentas de IA (ChatGPT, Copilot, Claude, Cursor, etc.) é **permitido e encorajado**. Ninguém programa hoje sem essas ferramentas, e nós também não.

Vamos avaliar como o código está **organizado e estruturado**. Portanto, mesmo usando IA, pense criticamente sobre a arquitetura: separação de responsabilidades, naming, e clareza geral.

Requisitos

1. Clínicas

- Deve ser possível criar, alterar, visualizar e deletar uma clínica
- Informações da clínica: documento (CPF/CNPJ), razão social, nome fantasia
- Para além das informações básicas, uma clínica tem informações bancárias (banco, conta, agência). Essas informações também devem ser passíveis de alteração

2. Dentistas

- Dentistas são atrelados, necessariamente, a uma clínica
- Informações do dentista: nome, telefone, email
- Uma clínica pode ter um ou mais dentistas como administrador e responsável legal da clínica

3. Requisitos Técnicos

- O desafio deve ser executado usando Go, usando quaisquer pacotes que você julgar úteis/necessários.
- O código deve ser bem documentado e seguir boas práticas de programação.
- Incluir testes unitários para as principais funções/métodos.

Justificativa Técnica (Obrigatório)

Inclua no README (ou em documento separado) uma **breve justificativa (máximo 1 página)** respondendo:

1. Quais foram as 3 decisões técnicas mais importantes que você tomou e por quê?
2. O que você faria diferente se tivesse mais tempo?
3. Se usou IA, como ela ajudou e onde você optou por fazer diferente do que ela sugeriu?

Critérios de Avaliação

Critério	O que avaliamos
Funcionalidade	O sistema funciona? A lógica está correta? É possível criar e alterar informações de um seller? E dos membros?
Organização do Código	Separação entre lógica e interface da API. Naming claro. Estrutura coerente de arquivos e módulos.
Testes	Testes cobrindo cenários de sucesso e erro.
Interface	A interface da API é intuitiva? Seria fácil de integrar essa API em um front-end?
Justificativa Técnica	Capacidade de explicar e defender as decisões tomadas, tanto na escrita quanto na apresentação.

Entrega

Prazo: 02/03 às 23:59

Como entregar:

1. Enviar link do código-fonte em um repositório Git (GitHub, GitLab, etc.) para os emails: samilla.macedo@capim.com.br e rafael.barone@capim.com.br
2. README com instruções de como executar o projeto e rodar os testes.
3. Justificativa técnica (no README ou documento separado).

Apresentação

Você irá apresentar sua implementação (**20 minutos + 10 minutos de perguntas**) para dois desenvolvedores da Capim na semana seguinte à entrega. Agendaremos o horário.

Na apresentação, esperamos que você demonstre a API funcionando, explique as decisões técnicas que tomou, e responda perguntas sobre trechos específicos do código.

| 💡 *Boa sorte! Estamos ansiosos para ver o que você vai construir.*