

Desafio Técnico Serasa

Descrição: “Gestão de Dívidas Pessoais”

Crie uma aplicação frontend/backend para ajudar os usuários a gerenciar suas dívidas, permitindo que eles organizem melhor sua vida financeira.

Funcionalidades Principais

Cadastro de Dívidas

O usuário pode registrar novas dívidas com os seguintes campos

Título: Nome ou descrição da dívida (ex.: “Cartão de Crédito”).

Valor: Valor total da dívida.

Data de vencimento: Quando a dívida deve ser paga.

Status: “Pendente”, “Pago”, ou “Atrasado”.

Observações: Campo opcional para anotações.

Listagem de Dívidas

Mostrar todas as dívidas cadastradas.

Destaque visual para dívidas vencidas.

Edição e Exclusão

O usuário pode atualizar ou remover uma dívida existente.

Resumo Financeiro

Exibir um painel com

Total de dívidas cadastradas.

Valor total de dívidas pendentes.

Quantidade de dívidas pagas.

Autenticação

Registro e login do usuário.

As dívidas devem ser associadas ao usuário logado.

Resumo do desafio

Este desafio foi criado para avaliar suas habilidades e sua abordagem ao resolver problemas, não para limitar sua criatividade. Por isso, não se prenda à escolha de tecnologias específicas! Sinta-se à vontade para usar as ferramentas, linguagens e frameworks com os quais você se sente mais confortável ou que acha mais adequados para o desafio. O que importa para nós é a solução que você propõe, sua lógica de implementação, organização, e como você aplica boas práticas no desenvolvimento.

Lembre-se: não estamos avaliando apenas o resultado final, mas também como você pensa e estrutura o projeto. Portanto, mostre sua personalidade técnica e divirta-se com o processo!"

Diferenciais na execução do teste

Embora não sejam obrigatórios, diferenciais como o uso de **Design Patterns** e a inclusão de **testes** podem destacar sua solução e demonstrar sua atenção às boas práticas. Aplicar padrões para organizar o código para lidar com dados de forma eficiente, mostra sua capacidade de estruturar sistemas robustos e escaláveis. Escolher um padrão adequado ao contexto do desafio é suficiente para evidenciar seu pensamento arquitetural.

Além disso, a implementação de **testes unitários** e **de integração** é um grande diferencial.

Testes unitários garantem a confiabilidade de funcionalidades isoladas, enquanto testes de integração validam o funcionamento conjunto de diferentes partes do sistema. Mesmo que sejam simples, esses elementos mostram sua preocupação com a qualidade e manutenção do código, o que é altamente valorizado na prática profissional.