



Teste técnico

Backend Node + Typescript (NestJS)


Objetivo do teste

1. Avaliar sua capacidade de entender problemas e modelar soluções que façam sentido para a situação proposta, pensando em como ela pode evoluir ao longo do tempo
2. Avaliar também a sua habilidade técnica de transformar a solução e modelo propostos em código, considerando que ele será lido e mantido por outras pessoas de um time

Requisitos obrigatórios

1. Deverá ser entregue, para discussão ao vivo, uma imagem estilo “quadro branco” com a modelagem do problema e arquitetura proposta, para ilustrar a sua linha de raciocínio
2. Deverá ser entregue, também para discussão ao vivo, uma pasta contendo o código desenvolvido. Juntamente com o código, deve ser fornecido um arquivo [README.md](#) com as instruções para executar o código, que deve funcionar conforme o esperado
 - a. O código deve ser escrito em Node + Typescript com o framework do NestJS

Orientações gerais

1. Caso você tenha qualquer dúvida, fique à vontade para perguntar ou assumir as premissas que achar que fazem sentido. No caso de assumir algumas premissas, deixe claro a linha de raciocínio que justifica o que foi assumido
 2. Se planeje para apresentar o “quadro branco” com o seu raciocínio e o código funcionando em até 45 minutos, contando com as perguntas que faremos ao vivo.
- 

Desafio do teste

1. Usuários desejam consultar seu saldo total e disponível para resgate via chatbot, no contexto de previdência privada. Elabore uma API que retorna para o chatbot o saldo calculado do usuário.

- O saldo é composto de um valor total e um valor disponível para resgate
- O valor não disponível para resgate possui um prazo de carência (uma data no futuro para liberação do valor)
- O saldo é o somatório de diversas contribuições feitas ao longo do tempo no plano de previdência privada do usuário

2. Além de consultar o saldo, os usuários precisam solicitar resgates dos valores disponíveis para resgate. Elabore também uma API que viabilize os pedidos de resgate do usuário

- O usuário pode solicitar o resgate total sem especificar o valor, ou um resgate parcial especificando o valor
- O usuário não pode solicitar um resgate maior do que o saldo disponível

Observações:

- Você não precisa implementar o chatbot, assumo que ele irá consumir a sua API e lidar corretamente com a resposta fornecida
- Você também não precisa implementar a persistência dos dados em banco de dados - eles podem ser “mockados”, mas é importante explicar como você modelaria os dados e qual tipo de banco de dados usaria, e os porquês.
- Leve em consideração aspectos reais de como esse chatbot pode funcionar, pensando em preocupações relevantes de um sistema em produção

