



Teste Técnico - Vaga Desenvolvedor Back End Pleno

Problemática

Vamos considerar que uma empresa X necessita ter seu próprio programa de fidelidade para o seu negócio, ele deseja que os seus clientes possam se cadastrar no programa, e qualquer compra efetuada, o cliente vá ganhando **1 ponto a cada R\$5,00 gastos**.

Cada cliente terá seu saldo individual em pontos que posteriormente ele poderá usar para resgatar prêmios do estabelecimento, **sendo os prêmios fixos no valor de 5, 10 e 20 pontos**.

Considere que os prêmios são:

Nome do prêmio	Valor em Pontos
Suco de Laranja	5 pontos
10% de desconto	10 pontos
Almoço especial	20 pontos

Com base nos requisitos acima, você será responsável pela **estruturação de uma API** que irá resolver o problema acima e irá alimentar uma aplicação frontend no futuro.

Pontos técnicos:

- Sua API deverá ser estruturada usando o framework Laravel
- Considerar para o uso do banco de dados, o MySQL (O uso das migrations do Laravel é opcional para a construção das tabelas no banco de dados) (Necessário enviar o SQL caso não seja feito o uso das migrations)
- Para testes relacionados aos envios dos e-mails, pode usar o [Mailtrap](#) ou qualquer outra ferramenta similar (a visualização do email sendo recebido será solicitada na call de análise do teste)

- A aplicação dos seguintes recursos do Laravel para elaboração da API será obrigatória para análise dos conhecimentos do candidato: Eloquent ORM, Jobs, Validations, Artisan Console (Cronjobs), Mail, Middlewares.

Considere que a API deverá contemplar os seguintes requisitos:

- 001 - Cadastrar um cliente no programa de fidelidade
- 002 - Buscar um cliente (seja por ID ou qualquer chave de identificação dele usada no cadastro)
- 003 - Listar todos os clientes cadastrados no programa de fidelidade
- 004 - Buscar o saldo de pontos desses clientes, bem como prêmios que já foram resgatados por ele (Importante mencionar que o decremento de pontos do saldo é necessário de acordo com o prêmio em questão)
- 005 - Resgate de um prêmio em específico pelo cliente, caso ele tenha saldo disponível
- 006 - Pontuar um cliente através de um valor gasto enviado (Não permitir pontuar, caso o valor seja menor que o mínimo de pontos)
- 007 - Listar as métricas relacionadas a somatória dos pontos e o valor gasto em um determinado período.
- Enviar um email ao cliente quando ele ganhar pontos e quando ele resgatar um prêmio) (Fique a vontade para elaborar o template a ser enviado ao cliente, seja criativo!)
- Todos os dias, rodar um script que será responsável por enviar um email aos clientes que tiverem saldo suficiente para resgatar o prêmio máximo do programa.
- Considere que todos os endpoints deverão conter uma autenticação usando Bearer Token nos endpoints. Segue abaixo as permissões que cada token deverá conter.

Token	Permissões
4b5f8f32c96a9aa152e0c6615d4e632f	001,002,003,004,005,006, 007
117ae721e424e7f819893edb2c0c5fd6	002,003,004
3b7d6e2cb06ba79a9c9744f8e256a39e	005,006

Para a análise do teste, será solicitado para que você compartilhe sua tela para apresentação, tanto do código em si, quanto da API rodando localmente ou hospedada em algum local de preferência.

Use o postman, Insomnia ou qualquer outra ferramenta para que possamos visualizar as requisições sendo realizadas e iremos conferir as informações sendo armazenadas no banco de dados.

Organize-se para a demonstração da solução desenvolvida acima. A explicação da sua solução será primordial para nossa análise.