

## Desafio Desenvolvedor Backend

### Cachaças Canuto

Bem vindo ao teste para a vaga de Desenvolvedor Backend na Monomyto Game Studios!

Nosso teste é uma API REST para a empresa Cachaças Canuto (fictícia), que está buscando modernizar sua gestão e vender seus produtos na internet.

O dono da Cachaças Canuto, o Sr. Victor Canuto, durante anos registrou em arquivos .JSON (convenientemente) todos os seus produtos, clientes e vendas. O Sr. Canuto registrou tudo à mão sem usar nenhum sistema ou ferramenta, então pode ser que contenha alguns erros de digitação ou problemas de formatação e padronização. O Sr. Canuto precisa de um backend, em forma de uma API REST, para gerenciar os dados e fornecer informações para o seu negócio, e ele precisa da sua ajuda.

Eis os arquivos que o Sr. Canuto disponibilizou:

[Catalogo.json](#)

[Clientes.json](#)

[Vendas.json](#)

Boa sorte!

### Avaliação

**Nesse teste iremos avaliar:**

- Arquitetura do código e boas práticas
- Simplicidade, sem over-engineering
- Facilidade de leitura, clean code
- Atenção aos detalhes

**Requisitos obrigatórios:**

- API REST: deve usar o protocolo HTTP, comunicar usando o formato JSON, e seguir boas práticas de REST.
- Instruções de como rodar o projeto no CMD (Linux ou Windows)

- Teste unitário de uma das features
- Requests de exemplo (cURL ou uma coleção no Postman)

Outros detalhes importantes:

- **( IMPORTANTE )** Esse teste foi feito para ser difícil para um Backend júnior, se não conseguir fazer ele funcionar 100% ainda sim envie pois iremos avaliar como você lidou com as dificuldades do teste.
- **Não é necessário um frontend** para visualizar as informações.
- **Não é necessário fazer operações de CUD** (create, update, delete) em nenhum dos domínios (venda, produto, cliente). Basta seguir a lista de funcionalidades.
- Entre em contato com os avaliadores caso tiver alguma dúvida! Ter dúvida não é problema, o problema é ficar em dúvida.
- Você pode usar Docker para distribuir o projeto para os avaliadores.
- Você pode usar a linguagem de programação que quiser. Na Monomyto usamos C# e [ASP.NET](#), mas você pode fazer em Java, C#, Rust, Python, Golang, Clojure, Elixir, Node, etc.
- É totalmente opcional usar um banco de dados para realizar o teste. Você pode armazenar os dados em memória na aplicação como quiser, e fazer a geração de relatórios na própria aplicação.
  - Caso usar um banco de dados e gerar os relatórios via SQL, iremos avaliar como usou o banco de dados para gerar os relatórios, e como fez para importar as informações para o seu DB.
  - Caso contrário, iremos avaliar o código que gera os relatórios.

*Não haverá surpresas intencionais no arquivo Vendas.json*

## Funcionalidades

Os filtros marcados como *\*busca textual\** significam que: deve ser case insensitive, e a string de busca pode aparecer em qualquer lugar do nome, como se fosse um ``lower(nome) like lower('%busca%')`` em SQL.

## Clientes

Retornar dados básicos de clientes, sem paginação:

- Adicionar busca textual por nome

- Adicionar filtro por data de nascimento, podendo filtrar em um range de datas (data inicial e final)
- Retornar dados de um cliente específico por ID

## **Produtos**

Retornar dados básicos de produtos, sem paginação:

- Adicionar busca textual por nome
- Adicionar filtro com range de teor alcoólico (mínimo, máximo)
- Retornar dados de um produto específico por ID

## **Vendas**

**Listagem simples de vendas com paginação:**

- Adicionar filtro por range de datas
- Adicionar busca textual por nomes de clientes
- Adicionar busca textual por nomes de produtos
- Adicionar opções de paginação (número da página e número de itens por página)

**Relatório de vendas por cliente, sem paginação:**

- Para cada cliente, retornar o total das vendas para esse cliente
- Para cada cliente, retornar os produtos que esse cliente mais comprou, podendo especificar uma quantidade máxima de itens
  - Ex: se eu especificar 1, deverá retornar apenas o item que o cliente mais comprou. Isso não altera a estatística de total de vendas, que ainda deve ser feito sob todas as vendas do período.
  - Retornar também a quantidade que ele comprou daquele produto, e o valor total das vendas.
- Permitir filtrar por range de datas (para considerar apenas vendas em um determinado período). Isso afeta a estatística de totais de vendas.

**Relatório de vendas por produto:**

- Para cada produto, retornar o total de vendas desse produto e a quantidade total vendida
- Opções de ordenação: Valor total vendido, Quantidade total vendida, crescente ou decrescente

**Bonus Points:**

O principal diferencial de um candidato à vaga de Backend na Monomyto é ter experiência prévia com a Unity, seja em projetos pessoais ou profissionais.

Caso queira usar sua experiência com a Unity como um diferencial no seu teste, precisamos também verificar seus conhecimentos. Como não é o foco da vaga, você pode nos linkar um repositório pessoal de algo feito na Unity ou criar uma pequena UI ou jogo na Unity que utilize o backend de alguma forma. Isso é completamente opcional, e se você não tiver tempo, não tem problema. Não tem problema mesmo. A avaliação na Unity seria totalmente subjetiva. Não é o foco da vaga, mas é um nice-to-have.

Caso for criar algo, você já tem que estar extremamente confortável com a Unity: se você não conhece a Unity, não precisa correr pra aprender em 1 semana para apresentar depois. O foco da vaga é Backend. Não adiantará você fazer o backend mal feito para dar tempo de criar algo na Unity apenas para ganhar os "bonus points": Você só ganha bonus points na avaliação se o backend estiver bem feito!

O único fator que determinará o resultado do teste é a aplicação backend. *\*O autor desse teste, por exemplo, não conseguiria esses bonus points :)\**