# Case Pipo

Beatriz C. da Costa

# Banco de dados

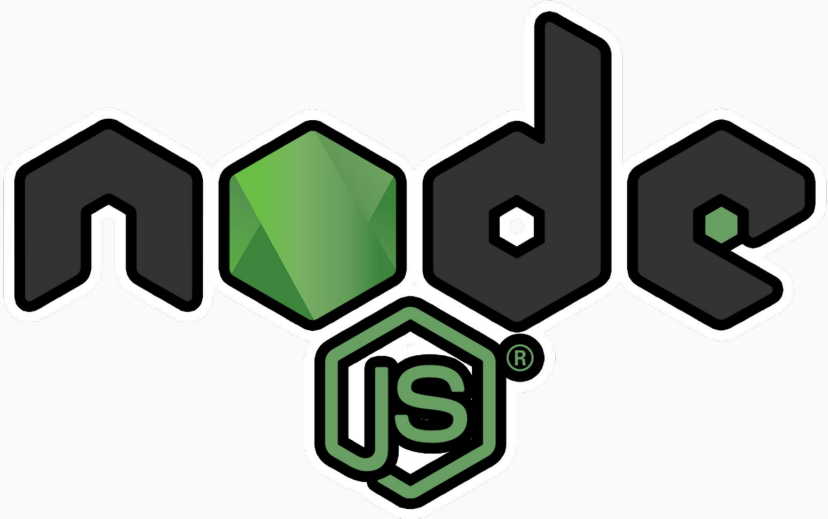
# Relacionamentos

Atualmente, podemos escolher entre bancos de dados relacionais ou não relacionais, eu irei desenvolver o desafio com banco de dados relacional, mas também seria possível usar o não relacional levando em conta que da forma mapeada na imagem abaixo verificamos que alguns campos ficarão em branco para alguns tipos de planos. Mas optei pelo relacional pois existem outras relações a levar em conta, como com a tabela cliente,planos ou ainda tabelas não descritas no desafio que possam fazer parte do sistema no futuro.


Table Pessoa
cpf
id
Table pessoa_beneficio
nome
id
id_pessoa
id_cliente
data
email
endereço
peso
altura
hora_meditacao
Table cliente
nome
id
Table plano_oferecido
id
id_cliente
id_plano
Table plano
nome
id

# Tecnologia de desenvolvimento

Optei pelo NodeJS, pois tenho mais experiência nessa linguagem, usando o framework Nestjs. Assim como o banco de dados PostgresSQL.

# Problemas enfrentados

Durante o desenvolvimento, percebi que deixar a lógica sobre o plano de saúde dentro do código, pode vir a ser um grande problema. Claro que para o escopo pequeno do desafio imagino que serviu bem. O que pode acontecer em outros cenários:

## 1. Não ser possível verificar qual plano é apenas pelos campos.

Nesse desafio, não é preciso digitar qual plano está sendo salvo, então fiz a decisão de acordo com os campos, mas caso houvesse planos diferentes com campos iguais, não seria possível.
Uma solução, ainda usando o mapeamento descrito no slide 4, é criar um endpoint que disponibilize a relação cliente-plano, assim o front consumiria esse endpoint para montar um select por exemplo, e então a api sempre receberia a info correta, mas é pouco seguro deixar sem verificação seja do banco com foreign key ou de lógica no código.

## 2. Muitos planos, muitas verificações

Caso haja muitos planos diferentes, existiriam muitas verificações para enfim poder salvar a informação correta, ou seja, muito código e seria lento.

Acredito que o melhor seria então, ao cadastrar um novo plano, criar também uma tabela específica para o mesmo, com as informações necessárias.

Observação: Quando me deparei com esse problema, não possuía tempo para implementar essa nova solução, então o código implementado foi a versão com as verificações.

# Obrigada pela atenção e oportunidade!

Contato:

53 984178299

beatrizdacosta1@gmail.com