

# Desafio Delivery



Considere que você irá projetar a API e o banco de dados de um sistema de delivery, o **EnaFood**. Considere que este delivery seja muito parecido com o iFood:

- Cada usuário poderá colocar na sacola produtos que estejam disponíveis para compra
- Ao final o usuário escolhe a forma de pagamento (cartão de crédito, vale alimentação, ...), confirma o endereço de entrega

Considere as seguintes fases do projeto:

- **Fase 1 - MVP:** Nesta fase o **EnaFood** possui apenas poucos usuários (por volta de 100) e cada usuário pede, em média, 5 vezes por mês. Cada pedido possui de 1 a 5 produtos.
- **Fase 2 – early adopters:** Nesta fase o **EnaFood** possui mais usuários (por volta de 10.000) e cada usuário compra, em média, 10 vezes ao mês. Cada compra possui de 1 a 15 produtos.
- **Fase 3 – early majority:** Nesta fase o **EnaFood** possui ainda mais usuários (por volta de 1.000.000) e cada usuário compra, em média, 25 vezes ao mês. Cada compra possui de 1 a 20 produtos.
- **Fase 4 – late majority:** Nesta fase o **EnaFood** é um sucesso e é a principal rede de delivery Brasil (por volta de 50.000.000 de usuários) e cada usuário publica, em média, 30 vezes ao mês. Cada compra possui de 1 a 20 produtos (sendo que está em teste um novo módulo para atendimento de empresas, no qual poderão ser feitos pedidos ainda maiores, com preços mais em conta).

Considerando este cenário, implemente (em NodeJS +MongoDB) uma API que seja capaz de:

- Obter os produtos que estão disponíveis para compra
- Manipular os produtos na sacola (inserir, escolher quantidades, remover...)

O que deverá ser entregue?

- A prioridade na avaliação será no racional da abordagem;

- Os códigos deverão estar disponíveis para avaliação em algum repositório público (github por exemplo);
- Um relatório curto (1 ou 2 páginas) expondo as justificativas das escolhas de projeto e implementação (pode ser o próprio arquivo de README do projeto);