

EASY ECOMMERCE!

Front end, Micro serviço, CI/CD!

ARQUITETURA

- Front End – NextJS
- Micro serviços
 - Serviço de produto – NodeJS, Express.
 - Serviço de cotação de frete – NodeJS, Express.
 - Serviço de Ordem de compra – NodeJS, Express.

FRONT END

Usamos NextJS para contemplar todo o seu *tooling* e a capacidade de usar React com Server Side Rendering.

Junto, para auxiliar no estilo do projeto, usamos styled-components e Antd.

Repositório:

<https://github.com/vNNi/Easy-ecom/tree/master/application/easy-ecom>

MICRO SERVIÇOS

Separamos os micro serviços por domínios. Um para Produto, Cotação de frete e Ordem de produto.

Por conta de nenhuma necessidade grande de computação, resolvemos utilizar **NodeJS**, que vemos como um grande potencial para operações de I/O.

Todos estão seguindo o padrão da OpenAPI, junto do **Swagger**.

Repositório:

<https://github.com/vNNi/Easy-ecom/tree/master/application/microservices>

BANCO DE DADOS

Utilizamos **MySQL** para salvar os dados, pensando em escalabilidade, manutenção e padronização entre os projetos.

CI/CD

Como utilizamos o Github como plataforma para salvar nosso código fonte, fizemos todo o fluxo de pipeline utilizando o Github Actions Workflow.

Steps:

- Lint
- Test (Front).
- Test (Micro serviço).
- Deploy imagem Docker no Docker HUB.

COMUNICAÇÃO SÍNCRONA E ASSÍNCRONA

Entre os micro serviços, utilizamos comunicação assíncrona, com o Kafka intermediando as mensagens.

Para o front end, usamos HTTP síncrono para se comunicar com os serviços.

VÍDEO DE APRESENTAÇÃO



OBRIGADO!

André Luiz Soares - 339880

Juan Carielo - 339589

Rafael Marinho - 339763

Vinícius Komninakis - 339762