

Universidade de São Paulo - USP
Instituto de Ciências Matemáticas e de Computação - ICMC

SCC0219 - Introdução ao Desenvolvimento Web

Projeto: Pet Shop
Entrega 03: Entrega Final

Alunos

Ingrid Otani, NUSP 10818971

Gabriel Muniz Morão, NUSP 7236785

Gabriel Angelo Sgarbi Cocenza, NUSP 6448118

Stella Granatto Justo, NUSP 9558882

São Carlos - SP
Dezembro - 2020

Sumário

Sumário	2
O Projeto	3
Como rodar o projeto da parte FrontEnd	3
Dependências:	3
Passo a passo:	3
Funcionalidades / Sitemap	3
Usuário comum:	3
Administrador:	4
Como rodar o projeto da parte Backend	4
Dependências:	4
Passo a passo:	4
Tecnologias	5
Testes	5
Problemas encontrados ao longo de projeto	5

O Projeto

O projeto também pode ser visto através do link [criado no vercel](#).

O objetivo deste projeto consiste na criação de um sistema voltado para o funcionamento de um petshop online. Neste projeto serão implementadas as funcionalidades necessárias tanto para o funcionamento do ecommerce do lado do usuário-cliente como do lado do usuário-administrador do sistema.

Para melhor organização o backend e o frontend foram desenvolvidos separadamente, neste repositório está os arquivos do frontend, realizando a integração com o backend através de API's.

Como rodar o projeto da parte FrontEnd

Dependências:

- [NodeJS](#)
- [Yarn](#) ou [NPM](#)

Passo a passo:

- Instalar todas as dependências
- Clonar o repositório `git clone`
- Na pasta do projeto instalar os pacotes `yarn install` ou `npm install`
- Após instalado os pacotes, rodar o projeto `yarn start` ou `npm start`

Funcionalidades / Sitemap

Usuário comum:

- **Página inicial** carrossel com banner, banner de oferta e lista de todos os produtos.
- **Página de ração** com a lista dos produtos.
- **Página de resultado de busca** com a lista dos produtos a partir de uma busca do usuário.
- **Página do produto** com informações do produto e a opção de adicionar no carrinho.
- **Página do carrinho** com os produtos adicionados e a possibilidade de removê-los. Caso o usuário esteja logado é enviado para a página de endereço de pagamento, caso contrário, para a página de login.

- **Página de pagamento** para o usuário informar as informações do cartão e finalizar a transação. Caso tudo esteja correto, o usuário é enviado para a página da sua conta que possui também histórico de pedido.
- **Página da conta** mostra as informações do usuário logado e histórico de seus pedidos.
- **Página de login** para o usuário e o administrador realizar o acesso.
- **Página de cadastro** para a criação de um novo usuário.

Administrador:

- **Estoque** permitindo a listagem e remoção do produção. com links para edição e adição de produtos .
- **Adição de produto** permitindo a adição de novos produtos.
- **Edição de produto** permitindo a edição de produtos já existentes.

Como rodar o projeto da parte Backend

Dependências:

- [NodeJS](#)
- [Yarn](#) ou [NPM](#)
- criar um arquivo .env.local com as seguintes variáveis:
 - MONGODB_URI=mongodb+srv://zume:bUWo8qllkOWsRBGv@cluster0.h4fzu.mongodb.net/pet_shop?retryWrites=true&w=majority
 - MONGODB_DB=pet_shop
 - JWT_SECRET=bcc476d57d24b7581c63bfbe315d768e78af88dad77cd90fc8f81516477cc4f0

MONGODB_URI é para a conexão com o MongoDB Atlas enquanto que JWT_SECRET é a chave para a geração e decodificação dos JWT.

Passo a passo:

- Instalar todas as dependências
- Clonar o repositório `git clone`
- Na pasta do projeto instalar os pacotes `yarn install` ou `npm install`
- Após instalado os pacotes, rodar o projeto `yarn start` ou `npm start`
- Fazer as requests em `http://localhost:3000`

Tecnologias

No frontend, foi utilizado para desenvolver a aplicação o **ReactJS** e três packages específicas: **Font Size Changer** para auxiliar na acessibilidade e o usuário conseguir alterar o tamanho das fontes, **React Router** para a criação das páginas e **File Base 64** para conversão de imagem em código na base 64.

No backend, foi utilizado para desenvolver a aplicação o **NextJS** no Vercel sendo que na pasta API cada arquivo .js se torna um endpoint da API através de função serverless fornecida pelo próprio serviço da Vercel. Utilizou-se o CORS (Cross-origin resource sharing) para que o repositório do frontend fosse capaz de realizar as requests através de um outro repositório. Entretanto este procedimento na produção de um site verdadeiro não é recomendável por questões de segurança.

Para banco de dados utilizou-se uma instância do Mongo Atlas e o Mongoose para funcionar como um ORM facilitando os comandos de inserção, esquema de documentos, consultas e etc.

As informações do carrinho e do usuário são armazenadas em **LocalStorage**.

Todo o projeto foi utilizado o GitHub, o repositório do frontend pode ser acessado neste [link](#). Já o backend pode ser verificado neste [repositório](#).

Testes

Para a testar as telas e funcionalidades, realizamos rodadas de QA entre os integrantes para verificação visual e de códigos a cada componente e página criada. A documentação dos testes pode ser verificada neste [documento de QA](#).

Problemas encontrados ao longo de projeto

Encontramos problemas para juntar os códigos e resolver os conflitos gerados. Sendo assim, resolvemos centralizar o envio dos códigos para apenas um dos membros do grupo mais experiente no uso de React, que fez a junção dos elementos de forma padronizada e subiu a aplicação no Github.

Além disso, tivemos dificuldades para alinhamento do backend e frontend, já que ambos foram desenvolvidos em paralelo, resolvemos se comunicando constantemente e documentação dos campos para as request.