

REACT JS - SERRATEC

GRUPO 5

INTEGRANTES:

Marina Portugal

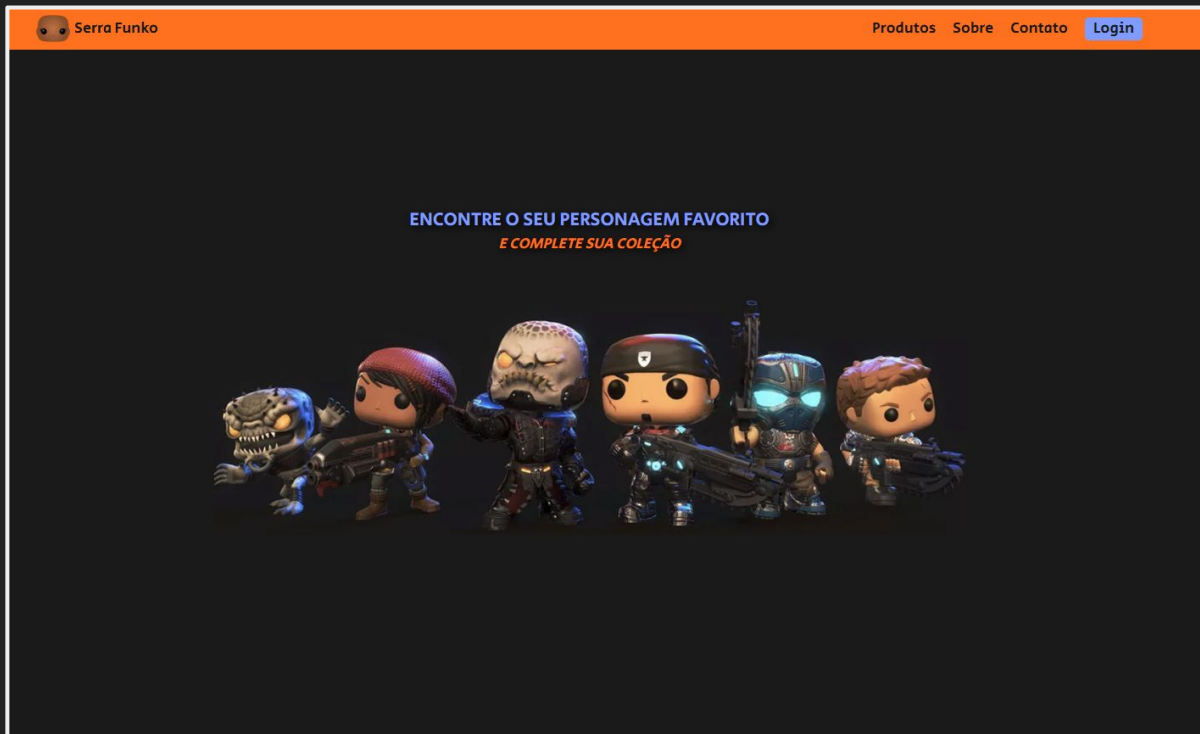
Breno de Medeiros

Ester Baltazar

Kauã Cassiano

Sophia Araujo

Pedro Henrique



SOBRE O PROJETO

- Nós utilizamos nosso projeto final de front-end essencial junto com a api do projeto final também resultado da matéria api restful para a construção desse novo projeto.
- Nosso primeiro passo foi migrar todo o projeto de front utilizando o react js, onde o maiores desafios foram:
 - Componentizar a maioria das tags utilizadas
 - Descobrir como setar os media queries no styled components
 - Importar o bootstrap no react
 - Utilizar a api do google maps

SOBRE O PROJETO

- No que se referia a api, tivemos também que fazer alguns ajustes como:
 - Ajustar para que categoria também salve uma foto.
 - Setar fotos para receber links do imgur (na substituição de um possível servidor de FTP). “Problemas de permissão com pasta local”
 - Criar um endpoint para listar todos os pedidos de um determinado cliente.
 - Setar tabela cliente para armazenar e criptografar senha no post do cliente.
 - Criar endpoint de validação do login do cliente.

FUNCIONALIDADES

- No que se refere aos nossos clientes, caso o mesmo não possua uma conta, deverá criar através do botão **registre-se** na tela de login. (POST CLIENTE)
- Logins e senhas de cliente não estão “mocados” e sim **salvos em banco de dados com senhas criptografadas**. (Autenticação Cliente)
- É possível deslogar também clicando no **botão sair**, para verificar o **status de login**, basta observar as informações do cliente na **Navbar (Desktop)** e no **NavMenu (Mobile)**.
- Também é possível acessar a página de **perfil do cliente para troca de dados, endereço e senha**. (PUT CLIENTE, PUT ENDEREÇO)

FUNCIONALIDADES

- A **página de catálogo** está sendo automaticamente gerada com a **consulta da api**, levando até a página do produto final, onde é possível adicionar o mesmo ao **carrinho**. (GET CATEGORIA, GET PRODUTO, GET ITEM_PEDIDO)
- Quando tentamos adicionar o produto ao carrinho é **verificado se o cliente está logado** através das informações que são salvas no **LocalStorage** no momento do login.
- É possível selecionar a **quantidade de itens** a serem inseridos no **carrinho** assim como a inserção de cupons para desconto, sendo configurados até o momento apenas o **CUPOM10** e **CUPOM15**.

FUNCIONALIDADES

- Depois disso é **verificado se o cliente já possui um pedido em aberto**, se não é realizada automaticamente uma **abertura do novo pedido** e adicionado este item ao carrinho do pedido. (POST PEDIDO, POST ITEM_PEDIDO)
- É possível **remover um item do carrinho** e também **finalizar o pedido** na tela do carrinho, procedimento o qual **gera um e-mail automaticamente através da nossa api** com os dados do pedido.(PUT FINALIZAR_PEDIDO, DELETE ITEM_PEDIDO, DELETE PEDIDO)
- Não fez sentido pra gente lidar com os PUT's de PEDIDO E ITEM PEDIDO uma vez que os mesmos ou são **gerados automaticamente** na inserção de um item ou na **alteração de quantidade de itens** em um pedido, que por **limitação da nossa api**, devemos **deletar o item do carrinho e inserir novamente** com a quantidade correta, **para não gerar erros no estoque**.

FUNCIONALIDADES ADM

- Já no que se refere aos POST's e PUT's de Categoria e Produto, os mesmos estão inseridos dentro de um painel administrativo.
- As rotas para esse painel administrativo estão protegidas por login com informações “mockadas” e também guardadas em LocalStorage. Portanto, não é possível acessar esse painel sem estar logado como admin.
- Nesse painel administrativo também é possível listar todos os Pedidos e excluir algum deles, desde que não contenham itens no pedido, assim como referente a Clientes que possuem ou não pedidos (por limitações de banco).
- As informações de PUT's para Cliente, Endereço e Pedidos já foram justificadas em um slide anterior.

INFORMAÇÕES ADICIONAIS

- Tanto o nosso front quanto o nosso back estão online e é possível acessar nosso sistema nesse exato momento, sendo front upado no **Netlify** e back upado no **Heroku**, vide link próximo slide.
- Nossas requisições api já estão apontadas para o endereço do heroku:
<https://serrafunko.herokuapp.com/ecommerce/>
- Nossa api está publicada no seguinte repositório:
<https://github.com/marinapsvreis/serratec-apirestfull-grupo5>
obs: Informações como DDL, DML, Json's e etc... estão presentes no repositório acima, na pasta arquivos extras.
- Nosso front está publicado no seguinte repositório:
<https://github.com/marinapsvreis/serratec-reactjs-trabalhofinal-grupo5>

OBRIGADO(A)!



Conheça nosso sistema:

<https://serrafunko.netlify.app/>