

# Pet Shop - Relatório

## Alunos:

Alysson Rogerio Oliveira (Nº USP: 9771355)

Guilherme Brunassi Nogima (Nº USP: 9771629)

Leonardo Akel Daher (Nº USP: 9771682)

Link do projeto no GitHub: [https://github.com/gbnogima/pet\\_shop](https://github.com/gbnogima/pet_shop)

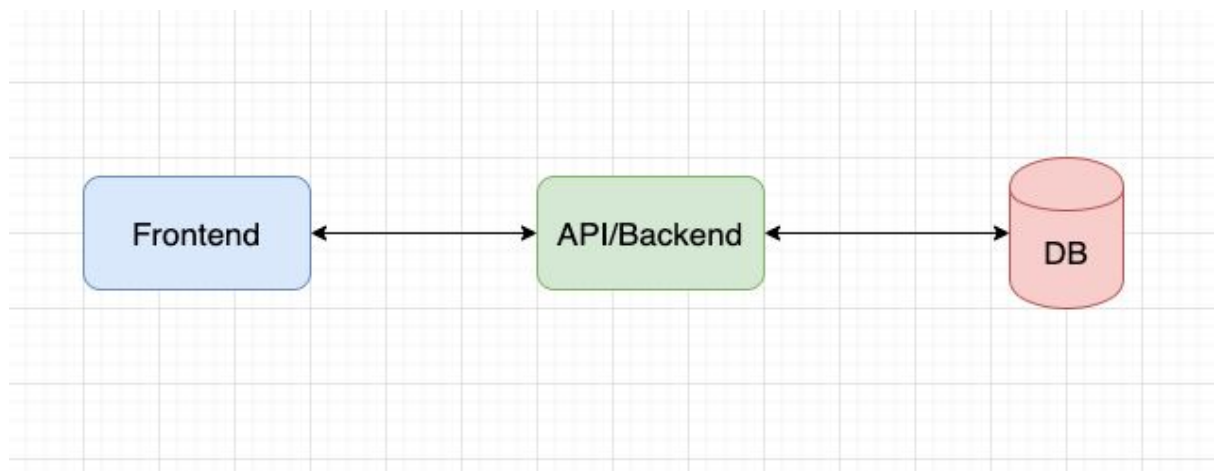
---

## 1. Requisitos

- O sistema possui 2 tipos de usuários: Clientes e Administradores
  - Os usuários utilizam a mesma página de login, no entanto, dependendo do seu tipo, são direcionados para páginas diferentes.
  - Os administradores são responsáveis por registrar e gerenciar administradores, clientes, produtos e serviços.
  - Clientes são usuários que acessam o sistema para registro de animais, agendamento de serviços e compra de produtos.
- Os dados do administrador são: nome, identificação, foto, telefone, email.
- Os dados do cliente são: nome, identificação, endereço, foto, telefone, e-mail
- Cada animal de estimação é registrado sob um cliente. Inclui nome, identificação, foto, raça, idade.
- Os dados do produto são: nome, ID, foto, descrição, preço, quantidade (em estoque), quantidade vendida.
- Os dados dos serviços são: nome, identificação, foto, descrição, preço.
- Os clientes podem agendar um horário para serviços, usando um calendário
- Venda de produtos: os produtos são selecionados, a quantidade escolhida e incluídos em um carrinho. Os carrinhos são esvaziados apenas mediante pagamento ou pelos clientes.
- Os administradores podem visualizar, criar e editar todos os produtos e serviços.

## 2. Descrição do projeto

O projeto consiste de três componentes, um projeto de backend, um de frontend e uma base de dados. O frontend foi implementado utilizando o framework *React*, o backend foi feito com *Node.js*, e a base de dados escolhida foi o *MongoDB*. A seguir temos um diagrama que exibe um design de como os componentes interagem:



A comunicação do frontend, que é o projeto que roda no navegador do usuário, com o backend, que rodaria em um servidor (caso fosse para produção) é feita por requisições HTTP.

Como optamos pelo uso de um banco de dados local, a comunicação é feita por um *wrapper* dos comandos do MongoDB, chamado *Mongoose*.

### 3. Comentários sobre o código

#### 3.1. Frontend

Para criar o projeto do frontend utilizamos uma ferramenta chamada *create-react-app*, que gera um projeto *React* já com uma estrutura inicial. A partir do projeto gerado, criamos a seguinte estrutura de pastas:

```
|— public/
|— src/
|   |— apps/
|   |— assets/
|   |— components/
|   |— views/
|   |— App.css
|   |— App.js
|   |— App.test.js
|   |— index.css
|   |— index.js
|   |— logo.svg
|   |— serviceWorker.js
```

```
|   |—— setupTests.js
|   |—— styles.css
|   |—— utils.js
|—— README.md
|—— package-lock.json
|—— package.json
```

Dos diretórios e arquivos acima, vale a pena destacar os seguintes:

- *components/* - diretório contendo componentes React que podem ser reutilizados
- *views/* - diretório contendo as páginas, que também são componentes *React*, mas com subcomponentes
- *App.js* - arquivo inicial do *Single Page Application*, é a raiz da árvore de componentes
- *package.json* - arquivo que descreve as dependências e os scripts disponíveis no projeto

### 3.2. Backend

Para o projeto do backend, não utilizamos nenhum gerador, criamos o projeto do zero, e utilizamos a estrutura a seguir:

```
|—— bin
|   |—— server.js
|—— src
|   |—— controllers/
|   |—— images/
|   |—— models/
|   |—— repositories/
|   |—— routes/
|   |—— services/
|   |—— app.js
|   |—— config.js
|   |—— utils.js
|—— README.md
|—— package-lock.json
|—— package.json
```

Assim como no projeto do frontend, vale a pena dar destaque para alguns arquivos e diretórios:

- *routes/* - nesse diretório estão definidos os endpoints disponíveis para o consumo do frontend.
- *controllers/* - os controllers são invocados pelos routes, e contém as regras de negócio do projeto.
- *repositories/* - realizam a comunicação com o banco de dados, utilizando para isso o *Mongoose*.
- *models/* - contém os modelos que são utilizados na base de dados, serve para descrever os campos de cada documento.
- *server.js* - arquivo no qual estão contidas as configurações iniciais que possibilitam a conexão e interação do usuário com a aplicação.

#### 4. Plano de teste

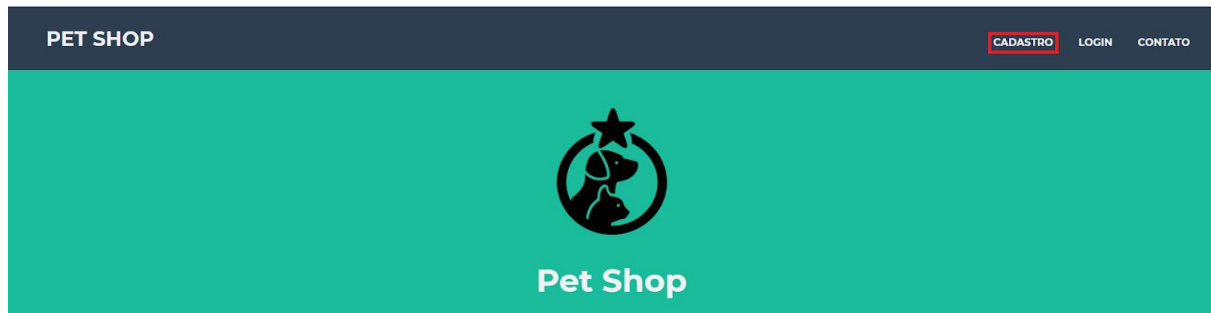
Para evitar a complexidade envolvida na configuração de ferramentas de testes, decidimos executar todos os testes manualmente. Para realizar isso, dividimos o sistema em alguns casos de uso, e assim executamos os testes para garantir a funcionalidade dos mesmos. Os casos de uso são os seguintes:

- Cadastro de usuário
- Autenticação de usuário como admin ou cliente
- Cadastro e visualização de pets
- Agendamento de serviços
- Compra de produtos
- Cadastro de serviços
- Cadastro de produtos
- Alteração das informações da conta

Abaixo, temos a sequência de ações que devem ser realizadas em cada página para reproduzirmos as funcionalidades:

#### 4.1. Cadastro de usuário

A partir da página inicial, clicar em “cadastro”, no cabeçalho da página:



Então, deve-se preencher todos os campos e clicar em “Cadastrar”:

The image shows a registration form titled "CADASTRO" in large, bold, dark blue letters. Below the title are five input fields, each with a light gray border and a light gray background. The first field contains the text "Michael Scarn". The second field contains "michael.scarn@gmail.com". The third field contains "999999999". The fourth field contains "Threat Level Midnight Street". The fifth field contains four dots "....". Below these fields is a green button with the text "Cadastrar" in white.

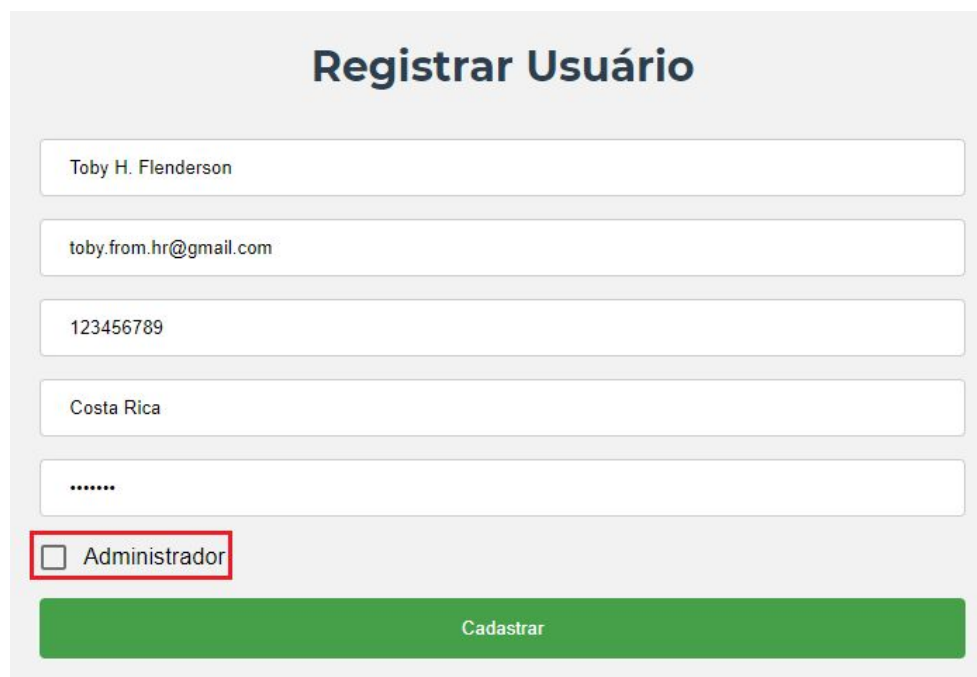
Por fim, será retornada uma mensagem informando se o cadastro foi realizado ou não.

#### 4.1.2 Cadastro de usuário através de um Admin

Uma outra opção disponível é realizar o cadastro de um novo usuário a partir de um Admin. Para tal, basta realizar o login em um cadastro de Admin (ver seção 4.2) e clicar em “Registrar Usuário”:



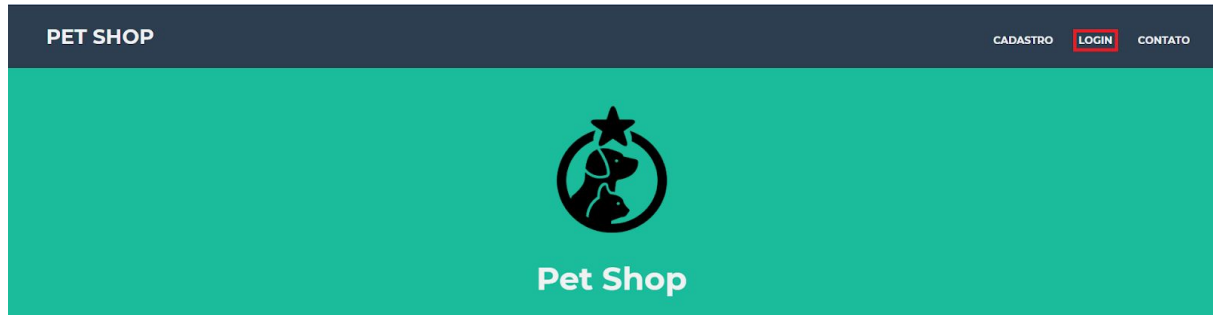
Um formulário será disponibilizado, após preenchê-lo, temos a opção de cadastrar o usuário como Cliente ou Administrador:

A imagem mostra um formulário web intitulado 'Registrar Usuário' em uma caixa cinza. O formulário possui cinco campos de entrada brancos: o primeiro contém 'Toby H. Flenderson', o segundo 'toby.from.hr@gmail.com', o terceiro '123456789', o quarto 'Costa Rica' e o quinto contém pontos para uma senha. Abaixo dos campos, há uma opção de seleção com um checkbox e o texto 'Administrador', que está circulado por um retângulo vermelho. No final do formulário, há um botão verde largo com o texto 'Cadastrar' em branco.

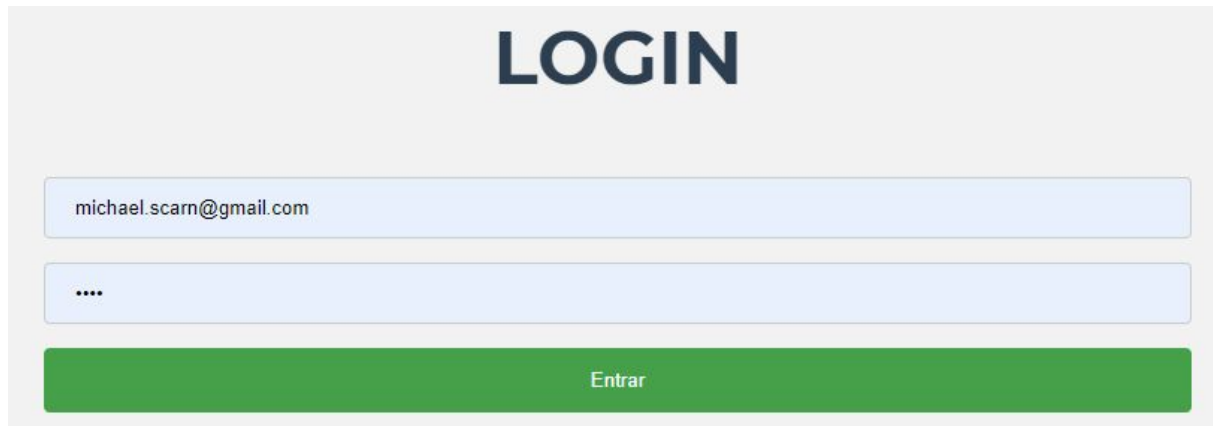
Por fim, clicar em “Cadastrar”.

## 4.2. Autenticação de usuário como Admin ou Cliente

A autenticação é realizada automaticamente pela aplicação, bastando para o usuário apenas clicar em “login”:



Então, inserir os dados requisitados e clicar em “login”:

The image shows a login form on a light gray background. At the top, the word "LOGIN" is displayed in large, bold, dark blue letters. Below it are two light blue input fields. The first field contains the email address "michael.scarn@gmail.com". The second field contains four dots "...." to represent a password. At the bottom of the form is a green button with the text "Entrar" in white.

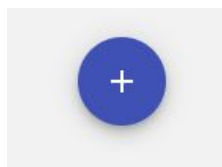
Caso o usuário seja um Cliente, será redirecionado para a área do cliente, caso seja um funcionário, será redirecionado para a área do admin.

### 4.3. Cadastro e visualização de pets

Na área do cliente, basta clicar em “Meus pets”:



Caso o cliente não possua nenhum pet, uma mensagem irá informar que não há pets cadastrados. Para realizar um novo cadastro, basta clicarmos no ícone “+”, no canto inferior direito da tela:



Um pequeno formulário será disponibilizado. Deve-se preencher as opções e clicar em “Cadastrar”:



## Cadastrar pet

Escolher imagemCadastrar

Então, o cadastro do pet será mostrado:



**Snoopy**

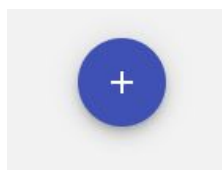
- Raça: Beagle
- Idade: 7
- Peso: 10 Kg
- Espécie: Cachorro

#### 4.4. Agendamento de serviços

Na página de Clientes, clicar em “Agendar Serviço”:



Caso o cliente não possua nenhum serviço agendado, uma mensagem irá informar que não há nenhum agendamento. Portanto, basta clicarmos no ícone “+”, no canto inferior direito da tela:



Um pequeno formulário será disponibilizado. Deve-se preencher as opções e clicar em “Agendar Serviço”:

## Novo agendamento

Pet

Snoopy

Serviço

Tosa

Data

07/28/2020



Horários disponíveis

12h00

Adicionar Serviço

Então, o agendamento será mostrado:

## Meus agendamentos

Pet	Serviço	Data
Snoopy	Tosa	28/07/2020 12:00:00

#### 4.5. Compra de produtos

Na página de clientes, clicar em “Comprar Produtos”:



Então, os itens disponíveis para compra serão mostrados, além da opção de pesquisa de outros produtos:

[Ver Carrinho](#)



**Casinha para Cachorro**

- Preço: R\$ 50
- Descrição: Bela casinha para cachorros, muito confortável



Então, deve-se clicar no ícone de adicionar ao carrinho para prosseguir com a compra:



**Casinha para Cachorro**

- Preço: R\$ 50
- Descrição: Bela casinha para cachorros, muito confortável



Por fim, clicar em “Ver Carrinho” para conferir os itens que serão comprados e então em “Finalizar Compra”:

### Compras

[Ver Carrinho](#)

Produto	Preço	Descrição
Casinha para Cachorro	50	Bela casinha para cachorros, muito confortável

[Finalizar Compra](#)

## 4.6. Cadastro de Serviços

Essa funcionalidade é disponibilizada apenas para Admins, portanto, na área do Admin, clicar em “Gerenciar Serviços”

 [Registrar Usuário](#)

 [Gerenciar Produtos](#)

 [Gerenciar serviços](#)

 [Minha conta](#)

Então, uma página com os serviços existentes será exibida, clicar em “Novo Serviço”

## Gerenciamento de serviços

Novo Serviço

Adicionar Serviço

Preencher o formulário e clicar em “Adicionar Serviço”:

## Gerenciamento de serviços

Novo Serviço

Adicionar Serviço

Por fim, a tabela contendo os serviços disponíveis é atualizada e disponibilizada:

Serviço	Preço
Tosa	20
Banho	25

#### 4.7. Cadastro de Produtos

Essa funcionalidade é disponibilizada apenas para Admins, portanto, na área do Admin, clicar em “Gerenciar Produtos”:





Então, uma página com os produtos existentes será exibida, clicar em “Novo Produto”:


## Controle de estoque

Novo Produto

Pesquisar item

Produto	Preço
Casinha para Cachorro	50

Um pequeno formulário irá aparecer, após preenchê-lo, clicar em “Adicionar Produto”:



Cama para gatos

Cama confortável para gatos

55

10

0

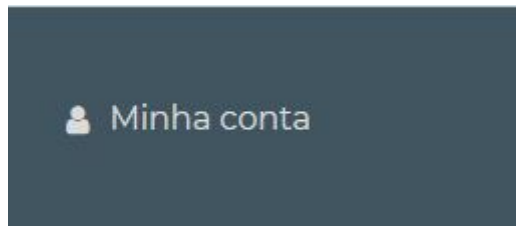
Adicionar Produto

O produto adicionado será mostrado no estoque:

Produto	Preço
Casinha para Cachorro	50
Cama para gatos	55

#### 4.8. Alteração das informações de conta

Essa funcionalidade está disponível tanto para Clientes quanto para Admins. Após realizar o login, basta clicar em “Minha Conta”:



Os dados do usuários serão disponibilizados na tela:

**Agent Michael Scarn**

- Email: michael.scarn@gmail.com
- Telefone: 999999999
- Endereço: Threat Level Midnight Street

Editar Cadastro

Ao clicar em “Editar Cadastro”, um formulário irá ser disponibilizado. Após preenchê-lo, clicar em “Salvar Alterações”:

Editar Cadastro

Michael Scott

michael.scarn@gmail.com

987654321

Orville Tootenbacher Street

Salvar alterações

Os dados cadastrais serão atualizados:

Michael Scott

- Email: michael.scarn@gmail.com
- Telefone: 987654321
- Endereço: Orville Tootenbacher Street

Editar Cadastro

## 5. Procedimentos para build

### 5.1. node.js

Para instalar o *node.js* recomendamos instalar o *nvm*, que é uma ferramenta de gerenciamento de versões do *node.js*. Para instalar o *nvm*, basta executar o seguinte comando:

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.35.3/install.sh | bash
```

Após instalado o *nvm*, basta instalar a última versão do *node.js* com o seguinte comando:

```
nvm install node
```

### 5.2. MongoDB

Para a base de dados, utilizamos o MongoDB, e para instalá-lo localmente, basta seguir as instruções do link a seguir:

<https://docs.mongodb.com/manual/administration/install-community/>

Como a instalação é diferente para cada sistema operacional e distribuição do Linux, não foi possível incluir no relatório um passo a passo.

### 5.3. Código-fonte

O código fonte do projeto está em um repositório do Github, portanto, basta executar o seguinte comando para clonar:

```
git clone https://github.com/gbnogima/pet_shop.git
```

## 5.4. Dependências

Como foi explicado na seção "descrição do projeto", o código foi dividido em duas pastas diferentes, uma para o frontend e outra para o backend, portanto, cada uma das partes tem suas próprias dependências.

Para instalar as dependências do frontend, basta rodar o seguinte comando, estando na raiz do projeto:

```
cd frontend && npm install
```

Para instalar as dependências do backend, basta rodar o seguinte comando, estando na raiz do projeto:

```
cd api && npm install
```

## 5.5. Execução

Para executar, basta estar na pasta raiz do projeto, abrir duas janelas de um emulador de terminal, e executar os seguintes comandos, um em cada janela:

```
cd frontend && npm start           # Para o frontend
```

```
cd api && npm start                 # Para o backend
```

Agora basta abrir algum navegador e acessar a URL "<http://localhost:3000>", e será possível acessar o Pet Shop.

## 6. Desafios e Comentários Adicionais

Esta seção é dedicada a apresentar os obstáculos enfrentados durante o desenvolvimento do projeto e alguns comentários finais. Primeiramente, é importante ressaltar a complexidade do projeto, que conta com diversas funcionalidades, que tiveram implementações distintas para clientes e administradores. Além disso, foi também um desafio a utilização de diversas ferramentas e tecnologias, as quais os integrantes não tinham experiência prévia. Por fim, outro fator foi o tempo de execução, que, apesar de ser extenso, quando

somado a outras atividades e trabalhos, não foi suficiente para entregar o projeto com a qualidade desejada inicialmente pelo grupo.

Apesar das dificuldades enfrentadas, somadas ao desafio de nos adaptar ao ensino à distância, o grupo obteve sucesso ao implementar as funcionalidades especificadas e conseguiu também manter a boa comunicação e trabalho em equipe. A conclusão final, portanto, é de que o projeto foi bastante exigente, assim como foi também uma oportunidade para aprender novas tecnologias, desenvolvendo todas as etapas de um sistema completo.