

**UNIVERSIDADE DE SÃO PAULO**  
**INSTITUTO DE CIÊNCIAS**  
**MATEMÁTICAS E**  
**COMPUTAÇÃO**

SCC0219 Introduction to Web Development

Professor: Dilvan Moreira

Alunos:

3672382 - Richard Carvalho Calderan  
958350- Jonatan Ricardo Catai

Pet Shop functional client side

(segunda entrega)

## Objetivos

Desenvolver uma aplicação client side de um Pet Shop utilizando baseado na entrega anterior aplicando conceitos aprendidos em sala.

## Descrição do projeto

O projeto consiste em uma aplicação estilo Single-Page de um Pet Shop, onde o usuário poderá obter informações, contratar serviços de banho/tosa (grooming) e consulta veterinária. O usuário também poderá visualizar e adquirir produtos do Pet Shop. Haverá também uma área dedicada apenas à administração, onde somente usuários com privilégio de administrador poderão ter acesso.

## Estrutura e funcionalidades do projeto

Inicialmente, o aplicativo mostra uma visão geral do pet shop e oferece opções para o usuário interagir. Estas opções estão disponíveis principalmente no MENU do aplicativo. Cada opção deste menu leva o usuário para determinada parte da aplicação de acordo com seu interesse.

A aplicação está dividida em HEADER, NAV, CONTENT e FOOTER.

O Header contém o logo, uma barra de pesquisa e informações do usuário (se está logado ou não).

No Nav é exibido o MENU da aplicação com suas possíveis opções de caminho (Home, services, shop, contact e about).

No Content é a área que o conteúdo pertinente é exibido. Ao logar, por exemplo, todo o conteúdo do tipo de usuário estará disponível.

No FOOTER contém um campo para o usuário pode marcar se quer receber notícias do aplicativo (datas e ofertas do shop, etc) e ainda é exibido informações de rodapé.

Para esta entrega foi decidido converter toda a estrutura do projeto anterior para Vue.js, onde os dados que alimentam o sistema estão definidos em uma instância do VUEX (store)

## Estrutura de arquivos e descrição

ROOT

- | -node\_modules (é criado ao executar o npm)

- | -public

- | | -index.html (nosso arquivo estático index.html)

- | -src

- | | -assets (todos arquivos de imagem)

- | | -components (componentes do vue)

- | | | -managements (componentes de administrador)

- | | -database (arquivos de database definidos para a proxima parte do projeto)

- | | -router

- | | | -index.js (arquivo de rota do vue-router)

- | |--tests (arquivos de testes de unidades para próxima fase do trabalho)
- | |--views (Componentes das rotas do vue-router)
- | |--app.vue (componente raiz do vue)
- | |--main.js (arquivo onde é instanciado o vue)

## Componentes do Vue

*Managements (Management,finance, Person, Product,Service):* Responsáveis pela administração.

*Animals*

*Calendar*

*Cart*

*Content*

*Footer*

*Header*

*Login*

*Nav*

*Service*

*Shop*

Componentes do router: *About, Admin, Contact, Home, Login,Service, Shop*

## Instalação

Este projeto utiliza gerenciador de pacotes do node.js (npm), portanto é preciso ter o node instalado. Foi usado a versao 4.0.5 do *@vue/cli* para desenvolvimento.

Todas as dependências estão listadas no arquivo package.json do npm, portanto para instalar basta estar na raiz do projeto e executar:

*npm install.*

Em seguida, execute:

*npm run serve*

O *@vue/cli* vai dispor o projeto em <http://localhost:8080/>

## Visão geral de funcionamento

O vue é instanciado no arquivo main.js. Neste arquivo também instanciamos o *vuex*, onde adicionamos os nossos MOCKs que representará os objetos do banco de dados e alimentará nossa aplicação para esta fase.

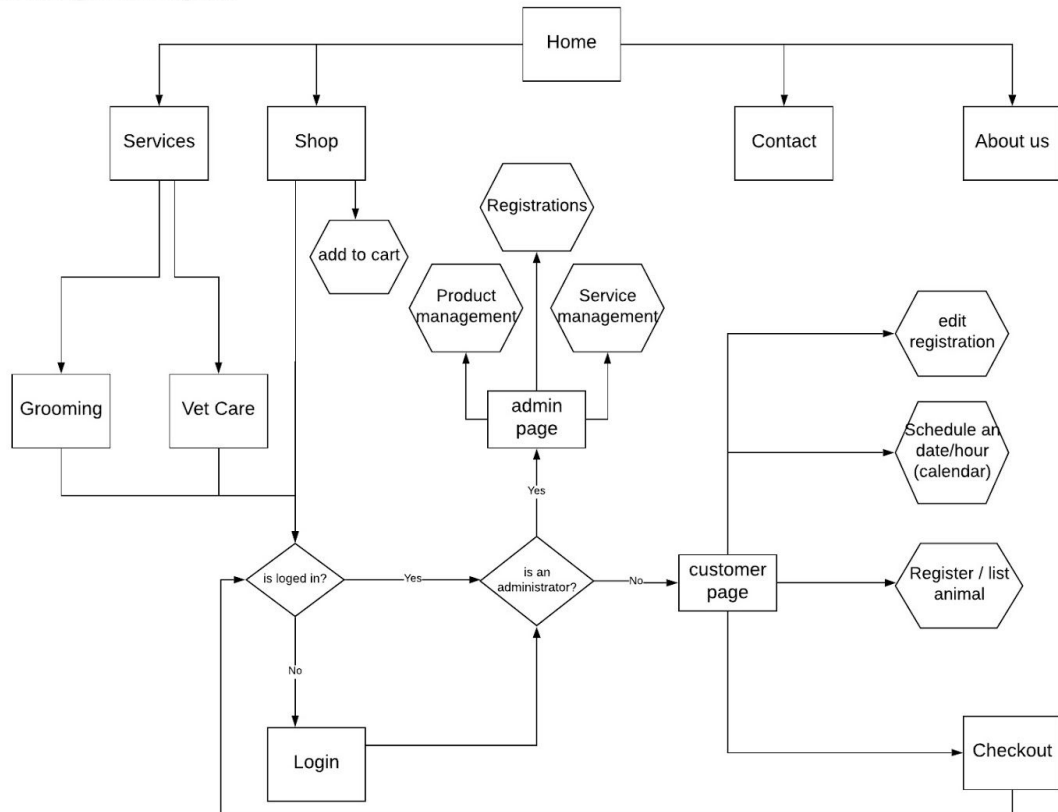
As rotas são definidas pelo vue-router, sendo que este sugeriu a criação da pasta views e com os devidos componentes de rota lá.

A instancia do vue chama o componente raiz app.vue, e este chama a estrutura base (header, content(router-view) e footer. O *Router-view* carrega as demais rotas baseado na requisição do usuário. As rotas estão definidas dentro da pasta router no arquivo *index.js*.

## Diagrama

A forma como o usuário navega no site pode ser visualizada no diagrama abaixo.

User Navigation Diagram



3672382 - Richard Carvalho Calderan  
958350- Jonatan Ricardo Catai

## Mocks

Para representar os dados que alimentam o sistema foi utilizado objetos do javascript que estão instanciados no arquivo main.js. Os objetos são:

- Person: Representa o usuario;
- Persons: Representa todos os usuarios registrados, incluindo administradores;
- Animals: Representa todos os animais cadastrados;
- Products: Representa todos os produtos vendidos no shop;
- Services: Representa todos os serviços oferecidos;
- Carts: Representa todos os carrinhos de compra;
- Schedules: Representa todos os horários reservados;
- Finances: Representa o movimento de caixa da aplicação;

## Considerações finais

Para fim de teste foi criado 2 atalhos de teste para simular o login dos usuários. Estes se encontram no canto superior direito do header. A aplicação já inicia com um usuário cadastrado.

Alguns arquivos foram adicionados para serem usados na parte final de entrega, como os testes de unidade e todo conteúdo da pasta database. Portanto, temos instalado o mocha e o mongoose, utilizados para fim de aprendizado, sendo assim não são utilizados nesta parte do projeto.

Para controle de datas foi utilizado o plugin v-calendar. Este plugin é chamado no componente Service.vue e é renderizado quando o usuário está cadastrado como “customer”.