

Implementações Full Stack

1) API

A API foi criada com o framework Spring Boot no ambiente de desenvolvimento IntelliJ em conjunto com as dependências Spring Web, Lombok e Firebase. Esta última permitiu a utilização do banco de dados não relacional Firebase do Google, com acesso via chave de serviço em formato JSON.

Foram desenvolvidas 3 coleções sendo elas: bdAnimais, bdTratamentos e bdDetalhesAnimais. Na coleção bdAnimal foram implementados os métodos POST, GET, PUT e DELETE e nas coleções bdTratamento e bdDetalhe foi implementado o método GET. Os métodos foram testados inicialmente com o aplicativo Postman.

A coleção bdAnimal apresenta os atributos nome, espécie, raça, altura, peso, tipo pelagem/plumagem e tipo de tratamento. Já a coleção bdTratamento apresenta os atributos nomeTratamento e valorTratamento, enquanto que a coleção bdDetalhe apresenta os atributos especieDetalhe, pelPlumDetalhe (array de Strings) e racaDetalhe (array de Strings).

2) Front JS, HTML, CSS

O front-end foi desenvolvido com o editor Visual Code Studio e o acesso as coleções foi realizado utilizando importação de funções via CDN. O sistema apresenta 2 telas, sendo elas Home e Cadastro e ambas as telas possuem header com navbar e footer fixos, sendo injetadas via JavaScript. Foram criadas 2 artes simples (logo SuperPet e imagem de fundo com patinha) com o objetivo de conhecer o editor gráfico Figma.

As configurações de layout foram realizadas através de seletores e principalmente com o emprego do framework Bootstrap. Foram utilizados também alguns ícones da coleção Icofont e fonte padrão Roboto.

Na tela de cadastro as opções disponíveis nos campos "Tratamentos" e "Especie" são carregados automaticamente nos respectivos combo boxes, sendo obtidos nas coleções bdTratamento e bdDetalhe. Quando o usuário escolhe a espécie, ocorre o carregamento das opções de mais 2 atributos, sendo eles "Tipo de Pelo/Pelagem" e "Raça" nos respectivos combo boxes e diretamente relacionados a espécie escolhida. Ambos atributos são obtidos também na coleção bdDetalhe.

Após os testes e validações necessários foi gerada uma build da aplicação, sendo hospedada com o recurso de hospedagem do Firebase no endereço: <https://bdpetshop-90019.web.app/>.

A implementação completa está disponível no repositório: <https://github.com/cristianofreese78/pagePetShop-HTML-CSS-JS>

3) Front Angular e Typescript

O front-end foi desenvolvido com o editor Visual Code Studio e o acesso as coleções via API (ainda em fase de conclusão). O sistema apresenta a mesma visualização da implementação utilizando JS, HTML e CSS, ou seja, com as telas Home e Cadastro injetadas entre o header e o footer, porém a integração destes elementos seguiram uma estrutura diferente.

Foram criados inicialmente 2 componentes, header e footer e depois foram criados 2 módulos, sendo eles Home e Cadastro, os quais são acessados através de roteamento, sendo Home a rota padrão quando o sistema é carregado.

Todos os componentes e módulos possuem estilizações em css e bootstrap de forma individual dentro dos respectivos componentes. As estilizações em comum com os componentes foi realizado no arquivo style.css. O framework Bootstrap é utilizado através de dependência.

A camada de serviço será responsável pelas requisições a API implementada inicialmente com Spring Boot, conforme descrito no item 1.

As primeiras validações foram realizadas com o auxílio do arquivo animal.ts, contendo uma estrutura de dados equivalente a coleção bdAnimal e uma pequena lista com esta estrutura, a qual é injetada em uma tabela quando a página de Cadastro é carregada.

A implementação completa está disponível no repositório no GitHub no endereço: <https://github.com/cristianofreese78/pagePetShop-SpringAngular>.

4) Idéias de implantação futura

A seguir são descritas algumas sugestões de implementações futuras:

- Acesso restrito via Login;
- Geração de relatório detalhado por tratamento;
- Geração de nota fiscal com a possibilidade de inclusão de mais de um tratamento;
- Inclusão de atributos na coleção bdAnimal relacionados a agendamento de tratamentos tais como lembrete de vacinas, serviços de banho e tosa;
- Possibilidade de maior detalhamento dos tratamentos, com a criação de cadastro de medicamentos, procedimentos, tipos de vacinas;
- Aplicações relacionadas a IoT – criação de um novo modulo no sistema com o objetivo de rastrear animais domésticos em tratamento. Estas informações aplicadas em conjunto a Machine Learning e Inteligência Artificial poderiam validar diagnósticos ou mesmo processos de recuperação de procedimentos. Neste caso poderiam ser empregados sensores wireless inerciais nos animais, com transmissão de dados via gateways, em muitos casos o próprio smartphone do tutor.