**CENTRO UNIVERSITÁRIO SENAC**

**Disciplina: PROJETO INTEGRADOR: DESENVOLVIMENTO DE SISTEMAS ORIENTADO A DISPOSITIVOS MÓVEIS E BASEADOS NA WEB**

**PROJETO NUTRIPET – ALIMENTAÇÃO NATURAL E PERSONALIZADA PARA CÃES**

**São Paulo**

**2024**

**CENTRO UNIVERSITÁRIO SENAC**

**Disciplina: PROJETO INTEGRADOR: DESENVOLVIMENTO DE SISTEMAS ORIENTADO A OBJETOS**

Júlio Gabriel Moreira Rocha

Laura Cristina Angioletto Martins

Pedro Favoreto Gaya

**PROJETO NUTRIPET – ALIMENTAÇÃO NATURAL E PERSONALIZADA PARA CÃES**

Projeto Integrador Desenvolvimento de Sistemas Orientado a Dispositivos Móveis e Baseados na Web 1 apresentado ao Centro Universitário Senac, como exigência parcial para obtenção de aprovação na disciplina Projeto Integrador Desenvolvimento de Sistemas Orientado a Objetos, do curso de Análise e Desenvolvimento de Sistemas.

Orientador (a): Prof. (ª) Gustavo Moreira Calixto

**São Paulo**

**2024**

**Sumário**

[INTRODUÇÃO E OBJETIVO DO PROJETO INTEGRADOR 4](#_Toc160883248)

[1. PROJETO NUTRIPET – ALIMENTAÇÃO NATURAL E PERSONALIZADA PARA CÃES 5](#_Toc160883249)

[1.1 Concepção da Aplicação 5](#_Toc160883250)

[1.2 Desenvolvimento de Personas 5](#_Toc160883251)

[1.3 Elaboração de um Protótipo Funcional 6](#_Toc160883252)

[2. Tecnologias utilizadas para o desenvolvimento do projeto....7](#_Toc160883253)

[2.1 Backend 7](#_Toc160883254)

[2.2 Frontend 9](#_Toc160883255)

[REFERÊNCIAS 10](#_Toc160883256)

# INTRODUÇÃO E OBJETIVO DO PROJETO INTEGRADOR

Este módulo do projeto integrador dá continuidade aos estudos de Análise e Desenvolvimento de Sistemas, focando em aspectos de sistemas orientados a dispositivos móveis e baseados na web.

O objetivo principal do módulo é revisitar a estrutura do projeto e suas funcionalidades, identificando áreas que tenham a possibilidade de serem aprimoradas e definindo uma prova de conceito que seja capaz de demonstrar a viabilidade de implementações futuras. Para tanto foi necessário preparar o ambiente de desenvolvimento do projeto, incluindo a seleção criteriosa das tecnologias utilizadas tanto no *front* quanto *backend*, bem como a escolha adequada do banco de dado que atendesse aos requisitos de desempenho e escalabilidade.

O *frontend* é responsável por assegurar uma interface de usuário intuitiva e responsiva, que seja capaz de oferecer uma experiência de usuário agradável em dispositivos móveis e páginas web. Já o *backend* precisa ser focado em garantir uma estrutura robusta e escalável que suporte as funcionalidades necessárias para o sistema, incluindo a implementação de um repositório de dados associado para o armazenamento e gerenciamento de informações de forma eficiente e segura.

# PROJETO NUTRIPET – ALIMENTAÇÃO NATURAL E PERSONALIZADA PARA CÃES

## Concepção da Aplicação

* **Título**: "NutriPet - Alimentação Natural e Personalizada para Cães"
* **Público-Alvo:** Donos de cães conscientes da saúde, interessados em nutrição natural e personalizada para seus pets.
* **Missão do Aplicativo:** Facilitar o acesso a dietas balanceadas e naturais para cães, promovendo saúde e bem-estar.
* **Benefícios:**
  1. Planos de alimentação personalizados baseados em necessidades nutricionais específicas.
  2. Informações detalhadas sobre ingredientes e benefícios para a saúde do cão.
  3. Sistema de entrega em domicílio para comodidade.
* **Diferencial:**
  1. Colaboração com veterinários para garantir dietas equilibradas.
  2. Opções de personalização com base em alergias e preferências alimentares do cão.
  3. Interface amigável e educativa com dicas de saúde para cães.

## 1.2 Desenvolvimento de Personas

**Persona 1: Laura, a Dona Cuidadosa**

* **Descrição:** Laura, 34 anos, trabalha em home office. Ela é muito atenta à saúde de seu Labrador, Max, e prefere produtos orgânicos e sustentáveis.
* **Necessidades:** Uma dieta natural para Max que apoie seu estilo de vida ativo e ajude na manutenção de seu peso ideal.
* **Objetivo:** Encontrar uma dieta saudável e natural para Max.
* **Ações:** Explora o aplicativo, lê sobre os benefícios dos ingredientes, personaliza a dieta de Max, faz um pedido.
* **Pensamentos:** "Quero o melhor para a saúde do Max, e NutriPet parece oferecer exatamente isso".
* **Emoções:** Sentimento de alívio e satisfação ao encontrar uma solução conveniente e saudável.

**Persona 2: Rodrigo, o Pai de Pet Consciente**

* **Descrição:** Rodrigo, 28 anos, um jovem profissional urbano. Ele tem um Bulldog Francês, Bella, que sofre de alergias alimentares.
* **Necessidades**: Uma opção de alimentação que possa ser personalizada para evitar alergênicos e melhorar a saúde da Bella.
* **Objetivo:** Achar uma alimentação que se adapte às alergias da Bella.
* **Ações:** Utiliza a opção de personalização para excluir alergênicos, consulta um veterinário através do app, faz o pedido.
* **Pensamentos:** "É difícil achar alimentos que a Bella possa comer sem problemas. NutriPet me dá essa opção".
* **Emoções:** Gratidão por encontrar uma solução adaptável e preocupação com a saúde da Bella sendo atendida.

## 1.3 Elaboração de um Protótipo Funcional

**Funcionalidades do Aplicativo**

* **Personalização de Dieta**: Usuários inserem informações sobre a raça, idade, nível de atividade e preferências alimentares do cão.
* **Consulta com Veterinários:** Opção de consultar um veterinário para uma recomendação de dieta personalizada.
* **Pedido e Entrega**: Sistema integrado para pedidos e entrega em domicílio.
* **Educação do Usuário:** Seção de dicas de saúde e nutrição para cães.

# TECNOLOGIAS UTILIZADAS PARA O DESENVOLVIMENTO DO PROJETO

## Backend

Para a realização do projeto em backend, inicialmente escolhemos desenvolvê-lo em Java com a utilização de Spring Boot por vários motivos que se alinham à agilidade, à eficiência e à simplicidade no desenvolvimento de aplicações Java. Primeiramente, o Spring Boot oferece uma vasta gama de funcionalidades autoconfiguráveis que aceleram significativamente o processo de desenvolvimento. Isso significa que posso me concentrar mais na lógica de negócios sem perder tempo configurando aspectos como conexão com banco de dados, segurança, ou a exposição de endpoints HTTP.

Após a definição das tecnologias do serviço para o backend, escolhemos desenvolve-lo em arquitetura hexagonal, principalmente pela necessidade de criar um sistema altamente desacoplado e testável. A principal motivação foi a separação clara entre a lógica de negócio do aplicativo e os detalhes de infraestrutura, como banco de dados, interfaces de usuário e comunicações de rede. Isso nos permitiu focar no desenvolvimento das regras de negócio sem me preocupar inicialmente com aspectos externos.

A arquitetura hexagonal, também conhecida como padrão Ports and Adapters, favorece a adaptabilidade e a evolução do sistema. Ao definir interfaces (ports) para as interações externas, conseguimos isolar a lógica de negócio de mudanças em tecnologias ou em estratégias de persistência e apresentação. Isso nos proporcionou a flexibilidade de modificar ou substituir componentes externos sem grandes impactos na lógica central do aplicativo.

Por fim, a escolha pela arquitetura hexagonal alinhou-se à estratégia de crescimento e escalabilidade do projeto. Com o desacoplamento e a modularidade promovidos, pudemos facilmente expandir ou modificar funcionalidades existentes, além de integrar novos serviços e tecnologias, acompanhando as evoluções do mercado e as necessidades do negócio. Este design arquitetônico provou ser uma decisão acertada para o sucesso e sustentabilidade do projeto.

Quanto ao banco, optamos pelo uso do MongoDB em nosso projeto por várias razões estratégicas que se alinham com as necessidades e objetivos específicos do sistema que estávamos desenvolvendo, principalmente por causa da escalabilidade, flexibilidade e performance superior para o manejo de dados não estruturados ou semiestruturados.

A escolha foi fortemente influenciada pela natureza do MongoDB como um banco de dados NoSQL orientado a documentos. Isso nos permitiu armazenar dados de forma mais natural e flexível, usando BSON (uma variação de JSON), o que facilita a modelagem de dados complexos e heterogêneos. Essa flexibilidade é especialmente útil em cenários onde o esquema dos dados pode mudar rapidamente ou precisa suportar uma grande variedade de tipos de dados, algo comum em nossas aplicações.

A escalabilidade horizontal oferecida pelo MongoDB também foi um fator determinante. Sua capacidade de distribuir dados através de sharding permite o gerenciamento eficiente de grandes volumes de dados, garantindo alta disponibilidade e balanceamento de carga. Para nós, que prevemos um crescimento acelerado e lidamos com grandes quantidades de acessos e operações, essa característica é crucial.

Além disso, valorizamos a performance superior do MongoDB para consultas complexas e operações de leitura e escrita, graças a recursos como índices secundários, consultas ad-hoc e agregações. Isso assegura tempos de resposta rápidos e uma experiência de usuário otimizada, essencial para nossas aplicações de uso intensivo de dados.

A funcionalidade avançada de agregação do MongoDB, que oferece uma interface poderosa para processamento e análise de dados, também foi uma razão importante para nossa escolha. Caso o projeto venha a se tornar um serviço grande, a escalabilidade de processos, dados no banco, se tornam muito mais ágeis em um sistema NoSQL e o mongo vem a se tornar, muito mais prático e útil que um banco SQL.

Em resumo, nossa escolha pelo MongoDB reflete a necessidade de um sistema de gerenciamento de banco de dados que nos ofereça flexibilidade na modelagem de dados, eficiente escalabilidade horizontal, alta performance para grandes volumes de dados, e suporte robusto para análises e operações complexas.

## Frontend

A escolha pela utilização do React.js no desenvolvimento do *frontend* teve como respaldo alguns aspectos, como componentização, performance e escalabilidade. Sob a ótica da componentização, a possibilidade de dividir a interface do usuário em componentes independentes e reuitilizáveis facilita a manutenção do código e permite um desenvolvimento mais rápido.

O modelo de renderização virtual do React.js atualiza somente as partes da página que foram alteradas, ou seja, resulta em uma performance mais eficiente em comparação com outros *frameworks*. Na prática, os aplicativos React têm maior tendência a serem rápidos e responsivos, implicando em uma melhor experiência do usuário.

Existe também uma comunidade muito ativa que utiliza o React.js, resultando em um grande ecossistema de bibliotecas e ferramentas. Desta forma, é possível encontrar uma ampla variedade de recursos, itens pré-construídos, documentação e suporte online.

Outros dois pontos muito importantes são a flexibilidade e a escalabilidade. Além destas características, o React.js ainda pode ser facilmente integrado com outras tecnologias e *frameworks*, como o Reduz e o React Router.

Por fim, o React.js é mantido pelo Facebook e utilizado por muitos dos seus produtos próprios, como Instagram e WhatsApp, o que garante contínua atualização, suporte e novas *features*.

# 

# REFERÊNCIAS