Sistema de Gerenciamento de Parceiros e Pets

Documento de Arquitetura de Software

Versão <1.3>

Histórico da Revisão

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| 14/06/2021 | 1.0 | Definido o escopo e a finalidade do documento. | Aline dos Santos  Patrick Tobias Valente |
| 15/06/2021 | 1.1 | Definida a forma de apresentação estrutural da arquitetura do software. | Aline dos Santos  Patrick Tobias Valente |
| 17/06/2021 | 1.2 | Adicionado Casos de Uso, Descrição dos casos de Uso de maior impacto na arquitetura do software. | Aline dos Santos  Patrick Tobias Valente |
| 18/06/2021 | 1.3 | Adicionado Diagramas de Sequência dos casos de uso selecionados. | Aline dos Santos  Patrick Tobias Valente |

Índice Analítico

1. Introdução 4

1.1 Finalidade 4

1.2 Escopo 4

1.3 Definições, Acrônimos e Abreviações 4

1.4 Visão Geral 4

2. Representação Arquitetural 5

2.1 Arquitetura Geral do Sistema 5

2.2 Representação Geral dos Controllers 6

2.3 Representação Geral dos Modelos 6

2.4 Representação Geral dos Templates da camada da View (visão) 7

3. Metas e Restrições da Arquitetura 7

4. Visão de Casos de Uso 8

4.1 Realizações de Casos de Uso 9

4.1.1 Gerenciar Parceiros 9

4.1.2 Gerenciar Animais/Pets 11

4.1.3 Listar animais para Adoção 15

4.2 Representações das Interações das classes de Impacto 16

4.2.1 Gerenciar Parceiro 16

4.2.2 Gerenciar Animais/Pets 17

4.2.3 Listar Animais para Adoção 20

5. Visão Lógica 21

5.1 Visão Geral 21

5.2 Pacotes de Design Significativos do Ponto de Vista da Arquitetura 22

5.2.1 br.com.fatec.ads.es.hopeapi 22

5.2.2 br.com.fatec.ads.es.hopeapi.controller 22

5.2.3 br.com.fatec.ads.es.hopeapi.model 23

5.2.4 br.com.fatec.ads.es.hopeapi.repository 24

5.2.5 br.com.fatec.ads.es.hopeapi.service 24

5.3 Pacotes de Design Significativos do Ponto de Vista da Arquitetura 25

5.3.1 templates 25

5.3.2 templates/animal 26

5.3.3 templates/parceiro 26

5.3.4 templates/pet 26

5.3.5 templates/usuario 27

Documento de Arquitetura de Software

# Introdução

## Finalidade

Este documento oferece uma visão geral do sistema que será desenvolvido na arquitetura MVC dentro do ecossistema Spring Boot, possibilitando uma abordagem geral da arquitetura da web application desenvolvida para atender as necessidades da Hope no gerenciamento de seus Parceiros e Pets. O objetivo fundamental deste documento é capturar e comunicar as decisões arquiteturais mais significativas que foram tomadas em relação ao sistema e destina-se aos desenvolvedores desta web application.

## Escopo

Este documento é influenciado pelo documento de Requisitos REQS2021, e apresenta as visões gerais do sistema como um todo e da visão lógica das classes do modelo de negócio, em forma de blocos e com diagramas de classe. Tem como escopo apresentar a arquitetura a ser utilizada na implantação dos processos de Gerenciamento de Parceiros e de Pets da plataforma solicitada pela iniciativa Hope, utilizando uma web application responsiva.

## Definições, Acrônimos e Abreviações

Vide documento de Glossário GLON2021.

## Visão Geral

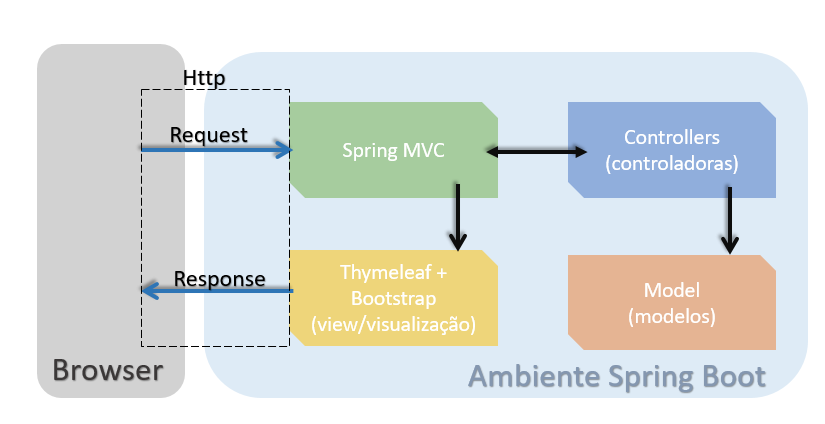
Este documento apresentará os diagramas de bloco contendo a visão geral da arquitetura de implementação, os casos de uso e diagrama de classe na seção de Visão Lógica onde a lógica do negócio será explicitada por meio do diagrama de classes dos modelos do negócio.

A seção de visão lógica apresentará a perspectiva das classes de maior impacto do sistema utilizando diagramas de sequência para evidenciar a interação das classes nesta arquitetura que emprega o ecossistema Spring Boot, SpringBootData, Thymeleaf e Bootstrap na camada de visão. Neste documento serão abordados os casos de uso de maior destaque e impacto no desenvolvimento inicial da web application.

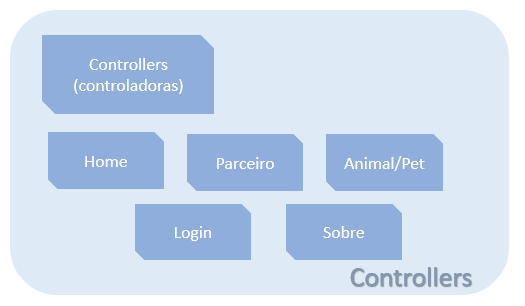
# Representação Arquitetural

O sistema a ser desenvolvido não possuía uma arquitetura previamente implementada, sendo um sistema a ser criado sob a arquitetura baseado no padrão Model-View-Controller, as seções que se seguem apresentarão os casos de uso de maior impacto na arquitetura do sistema, bem como a lógica do negócio. As visões lógicas estão representadas no diagrama de blocos 2.2, 2.3 e 2.4 pelos Controllers, Models e Templates respectivamente.

## Arquitetura Geral do Sistema

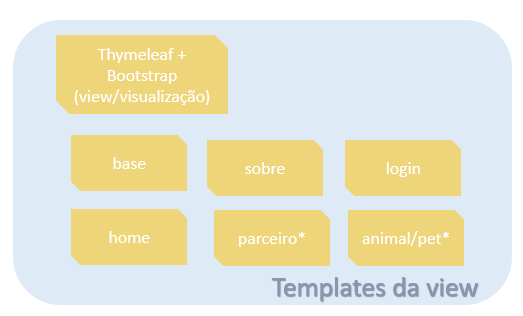


## Representação Geral dos Controllers



## Representação Geral dos Modelos

## Representação Geral dos Templates da camada da View (visão)



* As representações para parceiro\* e para animal/pet\* incluem:
  + View do formulário de cadastros
  + View do formulário de atualização de cadastros
  + View da listagem de cadastros

# Metas e Restrições da Arquitetura

O sistema será desenvolvido utilizando a arquitetura MVC permitindo que as classes acessem o banco de dados por meio de classes auxiliares de Requisição ou Services, bem como classes observadoras. Para gerenciamento das configurações do banco de dados o sistema utilizará o ecossistema SpringJpa com o Hibernate para comunicação com o MariaDB, podendo futuramente trocar o dialeto do banco de dados.

# Visão de Casos de Uso

Os casos de uso que representam funcionalidade central no sistema serão apresentados nesta seção, sendo eles: Gerenciamento de Parceiros, Gerenciamento de Animais/Pet e Visualização de Animais para Adoção, representando os casos de uso de configuração, de condução e de análise respectivamente. Cada subseção apresentará o diagrama do caso de uso, o fluxo principal e o fluxo alternativo do cenário em que o ator envolvido está inserido.

|  |  |
| --- | --- |
| C0001 | Visão Geral do Sistema |
|  |

|  |  |
| --- | --- |
| Caso de Uso de Configuração: Gerenciar Parceiros | |
| Identificação | Descrição |
| C0002 | Cadastrar parceiro |
| C0003 | Atualizar parceiro |
| C0004 | Remover parceiro |
| C0005 | Listar/Exibir parceiro |
| Caso de Uso de Análise: Gerenciar Animais/Pets | |
| Identificação | Descrição |
| C0006 | Cadastrar animal/pet |
| C0007 | Atualizar animal/pet |
| C0008 | Remover animal/pet |
| C0009 | Listar/Exibir animal/pet |

|  |  |
| --- | --- |
| Caso de Uso de Condução: Listar Animais para Adoção | |
| Identificação | Descrição |
| C0010 | Exibir lista dinâmica de animais/pets em status “Aguardando” |

## Realizações de Casos de Uso

Esta seção apresentará a descrição dos casos de uso C002, C0005-C008 e C009.

### Gerenciar Parceiros

|  |  |
| --- | --- |
| Nome do Cenário | Cadastrar Parceiro |
| Ator | Administrador |
| Diagrama | DC001 |
| Caso de Uso | C0001 |
| Pré-condição | Parceiro em potencial aprovado no processo de parceira. Vide RN006 do REGN2021. |
| Fluxo Normal | 1. O sistema exige a autenticação do usuário. 2. Após autenticação o sistema exibe a tela inicial do administrador. 3. O administrador acessa o formulário de cadastros clicando no botão “Novo Cadastro” 4. O administrador insere os dados solicitados. 5. O administrador salva os dados cadastrais clicando no botão salvar. 6. O sistema realiza validação dos dados\*. 7. O sistema salva os dados no banco de dados. 8. O sistema redireciona o administrador para a tela inicial. |
| Fluxo Alternativo | Caso o sistema não consiga validar os dados na etapa 6   1. O sistema exibe a mensagem de erro referente ao campo incorreto. 2. O administrador corrige e/ou insere os dados inválidos. 3. O sistema realiza a validação dos dados cadastrais. 4. O sistema salva no banco de dados. 5. O sistema redireciona o administrador para a tela inicial. |
| Pós-condição | Parceiro está apto a receber login e senha para gerenciar seus animais na plataforma. |

### Gerenciar Animais/Pets

|  |  |
| --- | --- |
| Nome do Cenário | Cadastrar Pet |
| Ator | Parceiro |
| Diagrama | DC001 |
| Caso de Uso | C0005 |
| Pré-condição | Parceiro autenticado na plataforma. |
| Fluxo Normal | 1. O sistema exige a autenticação do usuário. 2. Após autenticação o sistema exibe a tela inicial do parceiro. 3. O administrador acessa o formulário de cadastros clicando no botão “Novo Cadastro” 4. O parceiro insere os dados solicitados. 5. O parceiro salva os dados cadastrais clicando no botão salvar. 6. O sistema realiza validação dos dados\*. 7. O sistema salva os dados no banco de dados. 8. O sistema redireciona o parceiro para a tela inicial. |
| Fluxo Alternativo | Caso o sistema não consiga validar os dados na etapa 6   1. O sistema exibe a mensagem de erro referente ao campo incorreto. 2. O parceiro corrige e/ou insere os dados inválidos. 3. O sistema realiza a validação dos dados cadastrais. 4. O sistema salva no banco de dados. 5. O sistema redireciona o administrador para a tela inicial. |
| Pós-condição | Animal cadastrado está apto para ser visualizado por usuários comuns. |

|  |  |
| --- | --- |
| Nome do Cenário | Atualizar Pet |
| Ator | Parceiro |
| Diagrama | DC001 |
| Caso de Uso | C0006 |
| Pré-condição | Pet cadastrado, parceiro autenticado. |
| Fluxo Normal | 1. O parceiro acessa a lista de animais cadastrados clicando no botão “Gerenciar Pets”. 2. O sistema exibe a lista de animais cadastrados contendo os dados básicos no formato de tabela. 3. O parceiro clica no botão “atualizar” junto à linha da tabela correspondente ao animal desejado. 4. O parceiro insere os dados solicitados. 5. O parceiro salva os dados cadastrais clicando no botão salvar. 6. O sistema realiza validação dos dados\*. 7. O sistema salva os dados no banco de dados. 8. O sistema redireciona o parceiro para a tela inicial. |
| Fluxo Alternativo | Caso o sistema não consiga validar os dados na etapa 6   1. O sistema exibe a mensagem de erro referente ao campo incorreto. 2. O parceiro corrige e/ou insere os dados inválidos. 3. O sistema realiza a validação dos dados cadastrais. 4. O sistema salva no banco de dados. 5. O sistema redireciona o administrador para a tela inicial. |
| Pós-condição | Cadastro de animal atualizado. |

|  |  |
| --- | --- |
| Nome do Cenário | Remover Pet |
| Ator | Parceiro |
| Diagrama | DC001 |
| Caso de Uso | C0007 |
| Pré-condição | Pet cadastrado, parceiro autenticado. |
| Fluxo Normal | 1. O parceiro acessa a lista de animais cadastrados clicando no botão “Gerenciar Pets”. 2. O sistema exibe a lista de animais cadastrados contendo os dados básicos no formato de tabela. 3. O parceiro clica no botão “remover” junto à linha da tabela correspondente ao animal desejado. 4. O sistema removerá o cadastro do animal. |
| Fluxo Alternativo | Não se aplica. |
| Pós-condição | Cadastro do animal removido do banco de dados. |

|  |  |
| --- | --- |
| Nome do Cenário | Exibir Pet |
| Ator | Parceiro |
| Diagrama | DC001 |
| Caso de Uso | C0008 |
| Pré-condição | Pet cadastrado, parceiro autenticado. |
| Fluxo Normal | 1. O sistema apresenta na tela inicial todos os animais cadastrados por este parceiro. 2. O parceiro clica em “Gerenciar Pets”. 3. O sistema exibe a lista de animais cadastrados por este usuário contendo os dados básicos do animal no formato de tabela. |
| Fluxo Alternativo | Exibir animais por status   1. O parceiro clica na aba com o Status desejado. 2. O sistema lista os animais cadastrados por este parceiro cujo status é igual ao da aba selecionada listando os animais no formato de *cards*. |
| Pós-condição | Cadastro do animal removido do banco de dados. |

### 

### Listar animais para Adoção

|  |  |
| --- | --- |
| Nome do Cenário | Exibir lista dinâmica de animais/pets em status “Aguardando” |
| Ator | Usuário comum |
| Diagrama | DC001 |
| Caso de Uso | C0009 |
| Pré-condição | Pet com status “Aguardando” |
| Fluxo Normal | 1. O sistema apresenta a tela inicial da plataforma ao usuário. 2. O sistema seleciona dinamicamente 05 animais para serem exibidos ao usuário. 3. O sistema atualiza a lista de exibição considerando todos os animais de todos os parceiros de forma a manter uma lista diferente a cada visita ou atualização da tela. 4. O sistema exibe os animais em formato de cards contendo as informações básicas referentes ao animal. |
| Fluxo Alternativo | Não se aplica. |
| Pós-condição | Exibição randomizada de todos os animais com status “Aguardando” de todos os parceiros. |

## Representações das Interações das classes de Impacto

Esta seção apresentará o diagrama de sequência que ilustra a interação entre as classes ilustrando os casos de uso C0002, C0005-C0008 e C0009

### Gerenciar Parceiro

|  |
| --- |
| C0002 - Cadastrar Parceiro |
|  |

*\* as validações realizadas na camada de View (model:Model) não estão representadas neste diagrama.*

### Gerenciar Animais/Pets

|  |
| --- |
| C0005 - Cadastrar Animal/Pet |
|  |
|  |
| C0006 – Atualizar Animal/Pet |
|  |
|  |
| C0007 – Remover Animal/Pet |
|  |
|  |
| C0008 – Listar Animais Cadastrados |
|  |

### Listar Animais para Adoção

|  |
| --- |
| C0009 – Exibir lista dinâmica de animais/pets em status “Aguardando” |
|  |

# Visão Lógica

O sistema sob a perspectiva da lógica com foco nas classes de negócio está representado no diagrama de classes DC001. Classes auxiliares como Service, Controllers, Repositories, DTOs não serão representadas no diagrama de classes, mas suas responsabilidades serão destacadas na seção 5.2 – Pacotes de Design Significativos do Ponto de Vista da Arquitetura.

|  |  |
| --- | --- |
| DC001 |  |

## Visão Geral

Nesta seção os pacotes do modelo em termos de design de camadas e de hierarquia de pacotes serão apresentados separando as seções utilizando a hierarquia de pacotes do projeto Maven, onde src/main/java apresentará as classes das camadas de Controle e Modelo, e em src/main/resources os templates necessários para a renderização da camada de visão.

## Pacotes de Design Significativos do Ponto de Vista da Arquitetura

Os pacotes de design significativos que serão apresentados estão contidos no mesmo nível de hierarquia que se repete seguindo a hierarquia de pacotes em comum: *br.com.fatec.ads.hopeapi* as subseções apresentam o nome do pacote contido a partir deste nível. Nesta seção serão apresentados os pacotes e classes relativos à camada de modelo e controle das classes, sob o pacote src/main/java. Uma breve descrição de suas responsabilidades será apresentadas.

### br.com.fatec.ads.es.hopeapi

|  |  |
| --- | --- |
| Classe | Descrição |
| HopeApiApplication | Classe principal que inicia a aplicação. |
| WebSecurityConfig | Apresenta as configurações de autenticação (configurações básicas) |

### br.com.fatec.ads.es.hopeapi.controller

|  |  |
| --- | --- |
| Classe | Descrição |
| AnimalController | Responsável por mapear as ações/requisições referentes ao CRUD dos Pets |
| HomeController | Responsável por apresentar a lista dinâmica de Animais aos usuários comuns. |
| LoginController | Responsável por mapear a autenticação do usuário. |
| ParceiroController | Responsável por mapear ações/requisições referentes ao CRUD dos Parceiros |
| \*PetController | Responsável por mapear as ações/requisições referentes ao CRUD dos Pets |
| SobreNosController | Responsável por mapear as requisições referentes às informações sobre a ONG e seus parceiros. |
| \*UsuárioController | Responsável por mapear as ações/requisições referentes ao parceiro. |

*\* as classes em destaques são experimentais apenas para fins de implementação conforme acordado na aula, estão incompletas pois demandavam maior complexidade ao lidar com o usuário autenticado e suas ações sobre o CRUD.*

### br.com.fatec.ads.es.hopeapi.model

|  |  |
| --- | --- |
| Classe | Descrição |
| Animal | Responsável por modelar os animais, objetos dos cadastros de animas. |
| Parceiro | Responsável por modelar o parceiro, objeto dos cadastros e usuário autenticado. |
| \*Pet | Responsável por modelar os animais, objetos dos cadastros de animas. |
| PortePet | Enum responsável por manter as constantes com os portes dos animais. |
| SexoPet | Enum responsável por manter as constantes com os gêneros dos animais. |
| StatusPet | Enum responsável por manter as constantes referentes aos estados (status) dos animais dentro do processo de adoção. |
| TipoPet | Enum responsável por manter as constantes referente a categoria do animal. |
| \* User | Responsável por modelar o usuário autenticado (vinculado ao Parceiro). |

*\* as classes em destaques são experimentais apenas para fins de implementação conforme acordado na aula, estão incompletas pois demandavam maior complexidade ao lidar com o usuário autenticado e suas ações sobre o CRUD.*

### br.com.fatec.ads.es.hopeapi.repository

|  |  |
| --- | --- |
| Classe | Descrição |
| AnimalRepository | Interface responsável pelas ações/requisições da entidade Animal no banco de dados. |
| ParceiroRepository | Interface responsável pelas ações/requisições da entidade Parceiro no banco de dados. |
| \*PetRepository | Interface responsável pelas ações/requisições da entidade Pet no banco de dados, apresenta queries específicas relacionadas ao usuário autenticado. |
| \*UserRepository | Interface responsável pelas ações/requisições da entidade Usuário no banco de dados, apresenta queries específicas relacionadas ao usuário autenticado. |

*\* as classes em destaques são experimentais apenas para fins de implementação conforme acordado na aula, estão incompletas pois demandavam maior complexidade ao lidar com o usuário autenticado e suas ações sobre o CRUD.*

### br.com.fatec.ads.es.hopeapi.service

|  |  |
| --- | --- |
| Classe | Descrição |
| AnimalService | Interface responsável por auxiliar nas operações CRUD, será injetada na classe AnimalController. |
| AnimalServiceImpl | Classe responsável por implementar a interface AnimalService. |
| ParceiroService | Interface responsável por auxiliar nas operações CRUD, será injetada na classe ParceiroController. |
| ParceiroServiceImpl | Classe responsável por implementar a interface ParceiroService. |
| PetService | Interface responsável por auxiliar nas operações CRUD, será injetada na classe PetController. |
| PetServiceImpl | Classe responsável por implementar a interface PetService. |

## Pacotes de Design Significativos do Ponto de Vista da Arquitetura

Esta seção apresenta a hierarquia dos pacotes referentes à camada de visão (view) do sistema. Os pacotes apresentados em cada subseção estão contidos em: src/main/resources/ . Os arquivos de *application.properties* e *messages.properties* encontram-se neste nível de hierarquia, não possuindo um pacote específico.

### templates

|  |  |
| --- | --- |
| Arquivo | Descrição |
| base.html | Apresenta os componentes de estilo que são reutilizáveis no sistema. |
| home.html | Apresenta a camada de visão da home principal do usuário não autenticado e do usuário comum. |
| login.html | Apresenta a interface para autenticação do usuário. |
| sobre.html | Apresenta mais informações sobre a ONG e informações adicionais sobre os parceiros da Hope. |

### templates/animal

|  |  |
| --- | --- |
| Arquivo | Descrição |
| animal\_atualizado.html | Apresenta o formulário com os dados cadastrais do animal selecionado para eventuais atualizações. |
| index.html | Apresenta e lista as informações para gerenciar os animais cadastrados. |
| novo\_animal.html | Apresenta o formulário de cadastrar um novo animal. |

### templates/parceiro

|  |  |
| --- | --- |
| Arquivo | Descrição |
| index.html | Apresenta e lista as informações para gerenciar os parceiros cadastrados. |
| novo\_parceiro.html | Apresenta o formulário de cadastro de novo parceiro. |
| parceiro\_atualizado.html | Apresenta o formulário com os dados cadastrais do parceiro selecionado para eventuais atualizações. |

### templates/pet

|  |  |
| --- | --- |
| Arquivo | Descrição |
| index.html | Apresenta e lista as informações para gerenciar os animais cadastrados. |
| novo\_pet.html | Apresenta o formulário de cadastrar um novo animal. |
| pet\_atualizado.html | Apresenta o formulário com os dados cadastrais do animal selecionado para eventuais atualizações. |

*\* este template está vinculado às classes experimentais Pet e User e não está implementada por completo.*

### templates/usuario

|  |  |
| --- | --- |
| Classe | Descrição |
| home.html | Apresenta a tela inicial do parceiro autenticado. |

*\* este template está em fase de adaptação para incluir a classe não experimental Animal.*