**Nanny’s Pet**

**Documento de Arquitetura do Sistema**

**Histórico da Revisão**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| 19/09/2023 | 2.0 | Revisão Geral | Anna Carolinne, Renato Bernardino |
| 03/10/2023 | 2.1 | Revisão Geral | Anna Carolinne, Renato Bernardino, Virginia Claudia, Gabriel Ricardo |
| 02/01/2024 | 2.2 | Revisão Geral | Anna Carolinne, Renato Bernardino, Virginia Claudia, |

# **1 Arquitetura**

# Introdução

Este documento fornece uma visão arquitetural abrangente do sistema Nanny’s Pet, usando diversas visões de arquitetura para representar diferentes aspectos do sistema. O objetivo deste documento é capturar e comunicar as decisões arquiteturais significativas que foram tomadas em relação ao sistema.

# Termos e Abreviações

|  |  |
| --- | --- |
| - Tutor | Dono do pet, responsável pelo animal que será hospedado. |
| - Cuidador | Aquele que hospeda o animal temporariamente em sua residência. |
| - Pet | Animal de estimação do tutor que será hospedado. |
| - Hospedagem | Procedimento temporário de acolhimento do animal feito pelo cuidador. |
| - Residência | Local onde o pet será hospedado pelo cuidador. |

# Requisitos Significantes

[[Requisitos mais significantes que nortearam a arquitetura do sistema, obtidos através da metodologia do Mini-QAW ]]

# Restrições Arquiteturais

## Restrições técnicas

|  |  |  |
| --- | --- | --- |
|  | **Restrição** | **Contexto e/ou Motivação** |
| ***Restrição de software e programação*** | | |
| RT1 |  |  |
| RT2 |  |  |
| ***Restrição de sistema operacional*** | | |
| RT3 |  |  |
| RT4 |  |  |
| ***Restrições de Hardware*** | | |
| RT5 |  |  |
| RT6 |  |  |
|  |  |  |

# Escopo do Sistema e Contexto

## Diagrama de Casos de Uso

Diagrama

Descrição gerada automaticamente

## Um usuário realiza seu cadastro na plataforma, como tutor ou como cuidador.

## O tutor faz uma busca por um cuidador que preencha os requisitos desejados.

## Diagrama de Contexto

Diagrama

Descrição gerada automaticamente

* O tutor se cadastra na plataforma e busca por um cuidador que atenda suas necessidades.
* O cuidador após se cadastrar na plataforma é quem será procurado pelo tutor, para hospedar pets sob condições pré acordadas.
* A plataforma conecta tutores e cuidadores.
* A plataforma acessa a api dos correios ViaCep para localizar o endereço cadastrado através do CEP.

## Diagrama de Containers

Diagrama

Descrição gerada automaticamente

# Diagramas Conceituais

## Visão Lógica

Uma imagem contendo Diagrama

Descrição gerada automaticamente

|  |  |
| --- | --- |
| **Conceito** | **Descrição** |
| Pessoa | Mantém informações sobre o usuário |
| Tutor | Indica que o usuário é tutor. |
| Cuidador | Indica que o usuário é um cuidador cadastrado e mantem informações sobre o cuidador. |
| Imagens | Imagens do local onde o cuidador hospedará os pets |
| Características | Características referentes ao perfil de cuidador. |
|  |  |

# Detalhamento da Implementação e Ambiente Físico

## Visão de Implementação

[[inserir o diagrama de componentes do CDU principal e dos CDUs detalhados]]

[[Listar os componentes e suas respectivas descrições]]

|  |  |
| --- | --- |
| **Componente** | **Descrição** |
|  |  |
|  |  |

## Diagramas de Componente – C4

## Diagrama de Distribuição(Opcional)

[[inserir o diagrama de Implantação]]

[[inserir uma descrição breve de cada nó. ]]

## Persistência

[[inserir a descrição superficial da implementação da persistência]]

[[motivação]]

[[sgbd e a motivação]]

[[tecnologia da implementação (ex. pacotes do django responsáveis pela persistência]]

[[módulos implementados que solicitam a persistência]]

## Interface de Usuário

[[inserir a descrição superficial da implementação da interface]]

[[tecnologia e a motivação]]

# Anexos

## API

**Exemplo. Pode ser indicado a documentação Swagger, mas ela deve ser complementada para além do que é gerado automaticamente, indicando o que cada endpoint faz.**

**Agendamento**

Listar os agendamentos

Uma requisição GET que listará todos os agendamentos.

*Parâmetros da requisição*

|  |  |
| --- | --- |
| **Parâmetros** | **Descrição** |
| ... | ... |
|  |  |

*Estrutura da resposta*

|  |  |  |
| --- | --- | --- |
| **Path** | **Tipo** | **Descrição** |
| [ ] | Array | Todos os agendamentos |
| [ ].id | Number | Id de um agendamento |
| [ ].data | Date | Data da consulta |
| ... | ... | ... |

*Exemplo de requisição*

GET '<http://edu.ifrn.cnat.clinica/api/agendamento>'

*Exemplo de resposta*

[ {

"id": 78,

"data": "2021-02-17"

}, {

"id": 79,

"data": "2021-03-01"

} ]

# Referências

[1] <https://biking.michael-simons.eu/docs/index.html>

[2] <https://hsc.aim42.org/documentation/hsc_arc42.html>

[3] <https://resources.sei.cmu.edu/library/asset-view.cfm?assetID=513862>

[4] <http://static.codingthearchitecture.com/documenting-software-architecture.pdf>