Gestão RecolhaKi

Documento de Arquitetura de Software

1.0

Índice Analítico

[1. Introdução 3](#_Toc321036874)

[1.1 Finalidade 3](#_Toc321036875)

[1.2 Escopo 3](#_Toc321036876)

[1.3 Definições, Acrônimos e Abreviações 3](#_Toc321036877)

[1.4 Referências 3](#_Toc321036878)

[1.5 Visão Geral 3](#_Toc321036879)

[2. Representação Arquitetural 4](#_Toc321036880)

[3. Metas e Restrições da Arquitetura 4](#_Toc321036881)

[4. Visão de Casos de Uso 4](#_Toc321036882)

[4.1 Realizações de Casos de Uso 6](#_Toc321036883)

[5. Visão Lógica 7](#_Toc321036884)

[5.1 Visão Geral 7](#_Toc321036885)

[5.2 Pacotes des Design Significativos do Ponto de Vista da Arquitetura 8](#_Toc321036886)

[5.3 Diagrama de componentes](#_Toc321036887) 9

[6. Visão de Implantação 8](#_Toc321036889)0

[7. Visão de Dados 9](#_Toc321036891)

[8. Tamanho e Desempenho 9](#_Toc321036892)

[9. Qualidade 9](#_Toc321036893)

[10. Exceções 8](#_Toc321036894)2

Documento de Arquitetura de Software

# Introdução

Esse documento provê uma visão de alto nível dos objetivos da arquitetura, dos estilos arquiteturais e componentes que foram selecionados para implementação das funcionalidades propostas pelos casos de usos levantados do Gestão RecolhaKi.

## Finalidade

Este documento oferece uma visão geral arquitetural do sistema Gestão RecolhaKi, usando diversas visões arquiteturais 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.

## Escopo

Este Documento de Arquitetura de Software é aplicado ao Sistema Gestão RecolhaKi, que será desenvolvido pelos alunos Nathan Santos, Antônio Frauzo, Guilherme Andrade e Jessica Portilio do curso de Sistemas de Informação da Universidade Federal de Sergipe como projeto da disciplina Engenharia de Software II.

## Definições, Acrônimos e Abreviações

Todas as definições, juntamente com os termos, acrônimos e abreviações necessárias à adequada interpretação e entendimento deste documento podem ser encontradas no documento de Glossário do Projeto.

## Referências

Os seguintes documentos foram utilizados como referência para a elaboração do documento arquitetura:

* Modelo de Análise
* Modelo de Regra de Negócio
* Modelo de Casos de Uso
* Descrição de Casos de Uso
* Documento de Requisitos Funcionais e Não Funcionais
* *RationalUnifiedProcess*

## Visão Geral

Com o objetivo de cobrir todos os aspectos da arquitetura, esse documento contém as seguintes subseções:

* Subseção 2: Descreve o uso de cada visão;
* Subseção 3: Descreve as restrições arquiteturais do sistema;
* Subseção 4: Descreve os requisitos funcionais que causam significante impacto na arquitetura;
* Subseção 5: Descreve a visão lógica da arquitetura;
* Subseção 6: Descreve a visão de processos;
* Subseção 7: Descreve a visão de implantação;
* Subseção 8: Descreve a visão de implementação;
* Subseção 9: Descreve a visão de dados;
* Subseção 10: Descreve as principais características de dimensionamento do software que têm um impacto na arquitetura;
* Subseção 11: Descreve como a arquitetura do software contribui para todos os recursos.
* Subseção 12: Mostra a hierarquia de exceções.

# Representação Arquitetural

Este documento apresenta a arquitetura como uma série de visualizações, mencionadas acima. Essas visões são apresentadas como Modelos do StarUML e utiliza a Linguagem Unificada de Modelagem (UML – *UnifiedModelingLanguage).*

Para representar a arquitetura do software foram utilizados como base os seguintes estilos arquiteturais:

* Camadas;
* Repositório de banco de dados.

# Metas e Restrições da Arquitetura

Para que o software tenha um comportamento esperado pelos stakeholders ele deve seguir as seguintes restrições:

* A versão Mobile do Gestão RecolhaKi dará suporte para execução apenas em sistemas Android.
* Já versão web do sistema deverá permitir a execução nos principais browsers disponíveis, como por exemplo: *Mozilla Firefox, Chrome, Safari, Opera e Internet Explorer*;

# Visão de Casos de Uso

Nessa seção serão listados os casos de uso que são representados no modelo de casos de uso. Esses casos de uso são:

* CSU01 – Manter Pessoa;
* CSU02 – Disponibilizar Material Reciclável para Coleta Seletiva;
* CSU03 – Avaliar Coleta;
* CSU04 – Notificar Problema;
* CSU05 – Manter Coletor;
* CSU06 - Coletar Material Reciclável;
* CSU07 - Confirmar Interesse Material Reciclável;
* CSU08 – Manter Empresa;
* CSU09 – Notificar Coletor;
* CSU10 – Avaliar Problemas Notificados;
* CSU11 - Visualizar Avaliações de Pessoas;
* CSU12 – Autorizar Coletor;
* CSU13 – Autorizar Empresa Coletora;
* CSU14 – Autenticar Usuário;

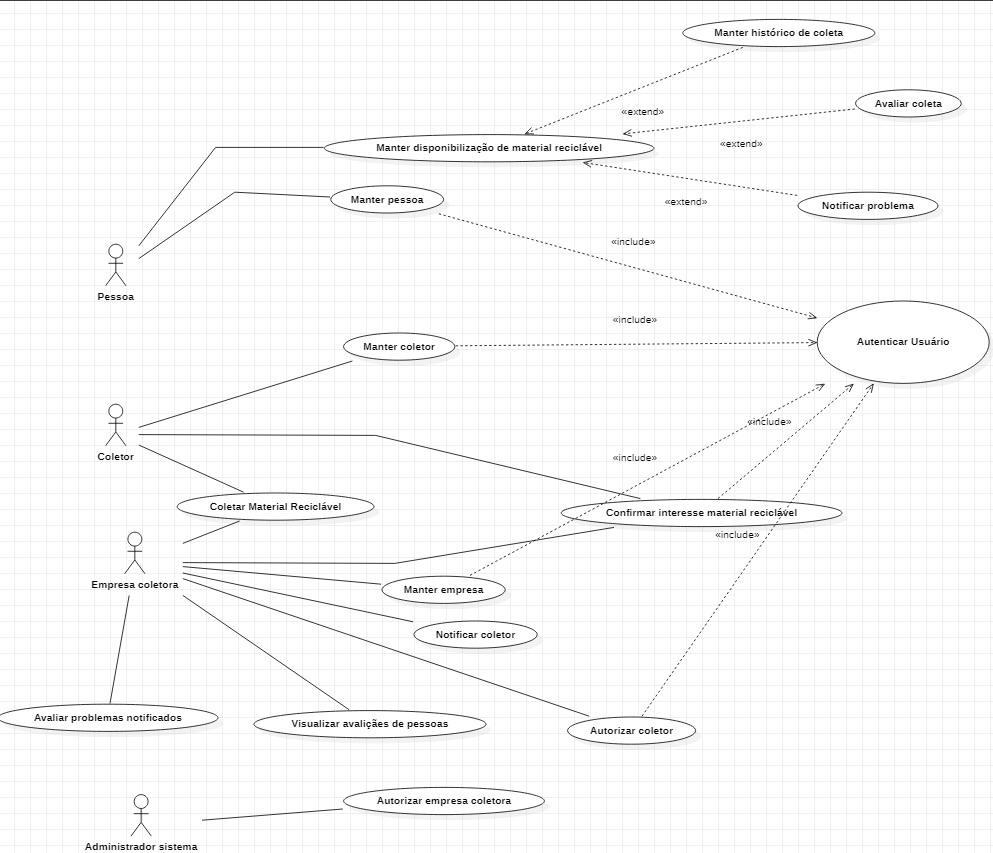


Figura 1 – Diagrama de Casos de Uso

## Realizações de Casos de Uso

A descrição de cada caso de uso contido no diagrama da Figura 1 encontra-se detalhada no diretório Requisitos > Casos de Uso.

# Visão Lógica

## Visão Geral

A visão lógica do Gestão Animal é composta principalmente por três pacotes:

* Service: Esse pacote representa a implementação da parte lógica do domínio da aplicação. Aqui serão armazenadas classes responsáveis pela persistência dos dados da aplicação;
* RecolhaKiWeb: Nesses pacotes são armazenados os componentes que fazem parte da interface gráfica da aplicação, as Controladoras e as ViewModels do RecolhaKi;
* Core: Nesse pacote são armazenadas as classes que representarão entidades da aplicação, e que poderão ser manipuladas pelos demais componentes da nossa arquitetura.

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

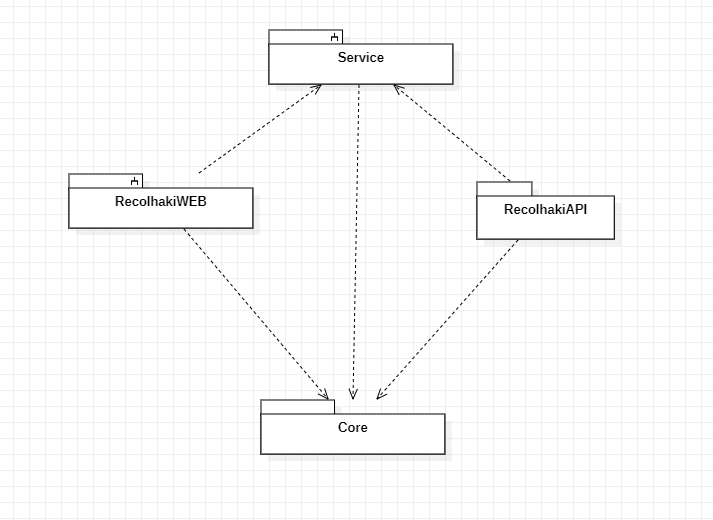


Figura 2 – Diagrama de Pacotes

## Diagrama de componentes

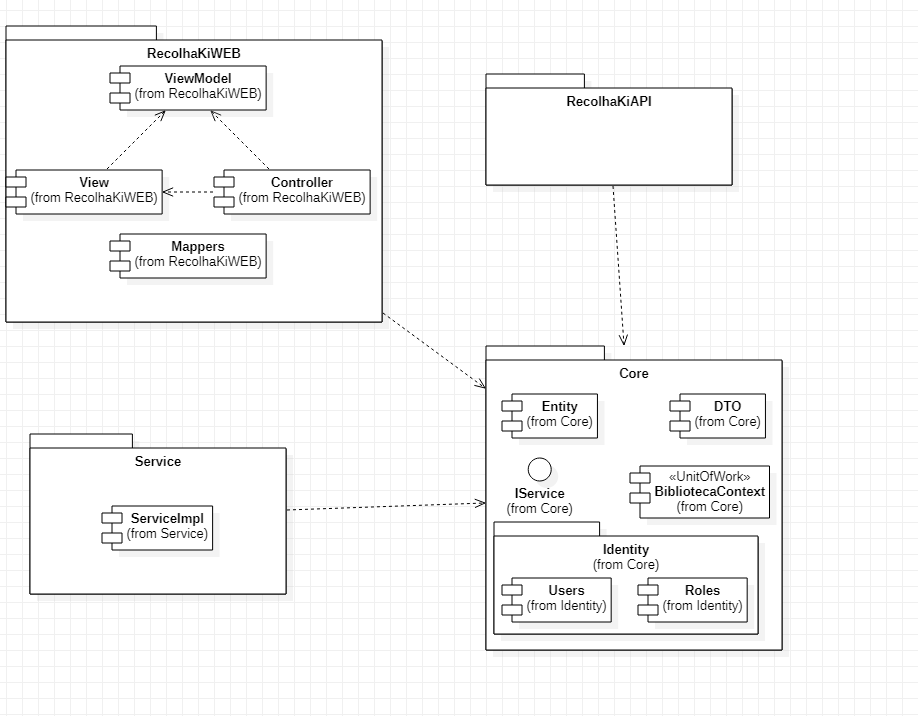


Figura 3 – Diagrama de Diagrama de Componentes

# Visão de Implantação

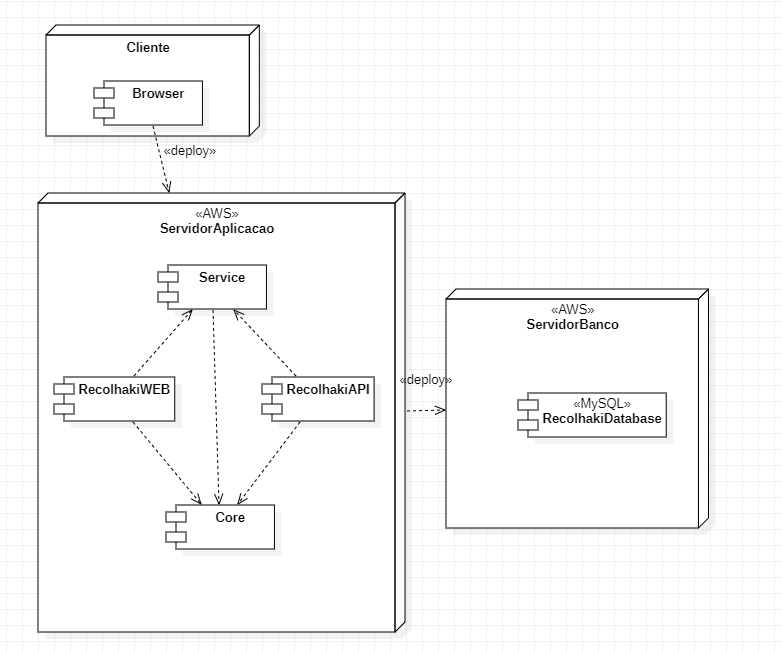


Figura 4 – Diagrama de Implantação

# Visão de Dados (Parcial)

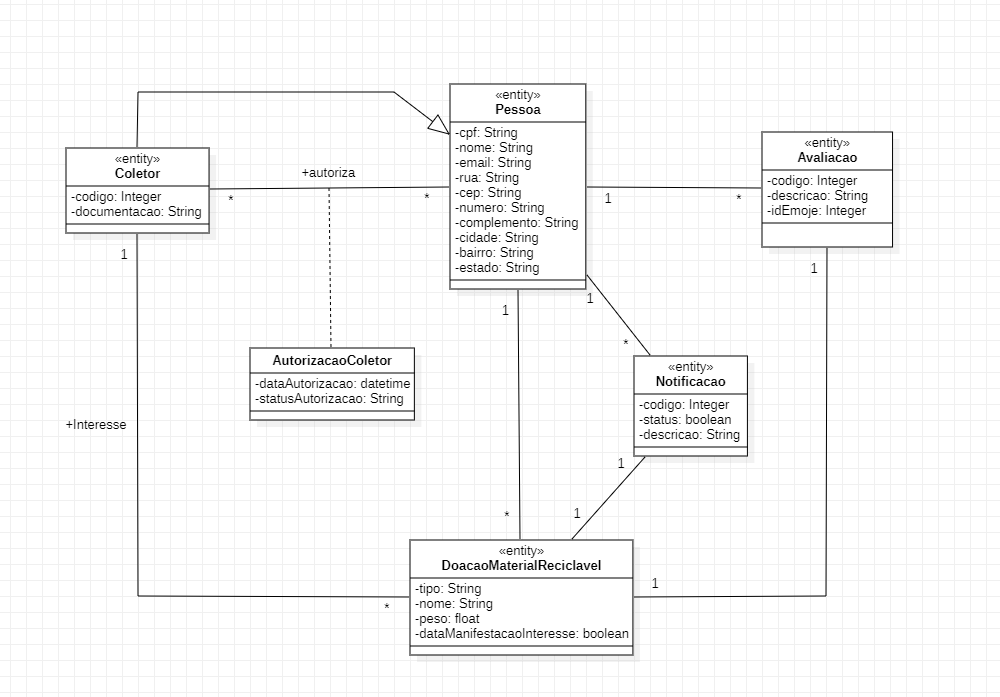


Figura 5 – Diagrama Entidade Relacionamento

# Tamanho e Desempenho

[Uma descrição das principais características de dimensionamento do software que têm um impacto na arquitetura, bem como as restrições do desempenho desejado].

# Qualidade

[Uma descrição de como a arquitetura do software contribui para todos os recursos (exceto a funcionalidade) do sistema: extensibilidade, confiabilidade, portabilidade e assim por diante. Se essas características possuírem significado especial, como implicações de segurança, garantia ou privacidade, elas deverão ser delineadas claramente].

# Exceções

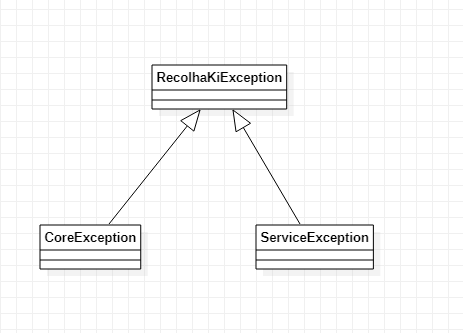


Figura 6 – Diagrama de Exceções