Residential Life

Documento de Arquitetura de Software

<Versão 1.0>

Índice Analítico

[1. Introdução 4](#_Toc321036874)

[1.1 Finalidade 4](#_Toc321036875)

[1.2 Escopo 4](#_Toc321036876)

[1.3 Definições, Acrônimos e Abreviações 4](#_Toc321036877)

[1.4 Referências 4](#_Toc321036878)

[1.5 Visão Geral 4](#_Toc321036879)

[2. Representação Arquitetural 5](#_Toc321036880)

[3. Metas e Restrições da Arquitetura 5](#_Toc321036881)

[4. Visão de Casos de Uso 5](#_Toc321036882)

[4.1 Realizações de Casos de Uso 7](#_Toc321036883)

[5. Visão Lógica 8](#_Toc321036884)

[5.1 Visão Geral 8](#_Toc321036885)

[5.2 Pacotes des Design Significativos do Ponto de Vista da Arquitetura 9](#_Toc321036886)

[5.3 Diagrama de componentes 10](#_Toc321036887)

[6. Visão de Processos 10](#_Toc321036888)

[7. Visão de Implantação 11](#_Toc321036889)

[8. Visão da Implementação 11](#_Toc321036890)

[9. Visão de Dados (opcional) 11](#_Toc321036891)

[10. Tamanho e Desempenho 11](#_Toc321036892)

[11. Qualidade 11](#_Toc321036893)

[12. Exceções 11](#_Toc321036894)

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 Residential Life.

## Finalidade

Este documento oferece uma visão geral arquitetural abrangente do sistema Residential Life, 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 Residential Life, que será desenvolvido pelos alunos Leonardo Santos, Nathanael Oliveira, Tauany Santana Thiago Almeida, Cleverton dos Santos e Andrés S. Menéndez, 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;
* Cliente-Servidor;
* Subrotinas;
* Baseado em Eventos;
* 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:

* O sistema deve permitir a execução nos principais browsers disponíveis, como por exemplo: *Mozilla Firefox, Chrome, Safari, Opera e Internet Explorer*;
* O sistema será implementado segundo a arquitetura .NET, usando como linguagem o C#;
* O sistema terá os seus dados persistidos no banco de dados SQL Server;
* Apenas usuários autenticados podem ser acessar o sistema.

# 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- Gerenciar Condomínios;
* CSU02- Gerenciar Blocos;
* CSU03- Gerenciar central de Atendimento;
* CSU04- Gerenciar Despesas/Receitas;
* CSU05- Gerenciar Moradias;
* CSU06- Gerenciar Cadastro do Proprietário;
* CSU07- Gerenciar Funcionários;
* CSU08- Gerenciar Pessoas que terão Acesso ao Prédio;
* CSU09- Publicar em Listas de Discussão;
* CSU10- Registrar Ocorrência;
* CSU11- Reservar Ambiente;
* CSU12- Publicar no Mural Eletrônico;
* CSU13- Votar em Enquete;
* CSU14- Gerenciar Veículos;
* CSU15 - Gerenciar Áreas Públicas;
* CSU16 - Gerenciar Ocorrências;
* CSU17 - Gerenciar Reservas de Ambientes;
* CSU19 - Gerenciar Enquetes,
* CSU20 - Gerenciar Email;
* CSU21 - Controlar Acesso ao Prédio;
* CSU22 - Autenticar no Sistema;
* CSU23 – Visualizar Movimentação Financeira;
* CSU24 – Central de Atendimento;
* CSU25 – Pontuar Pessoa;
* CSU26 – Analisar Movimentação Financeira;

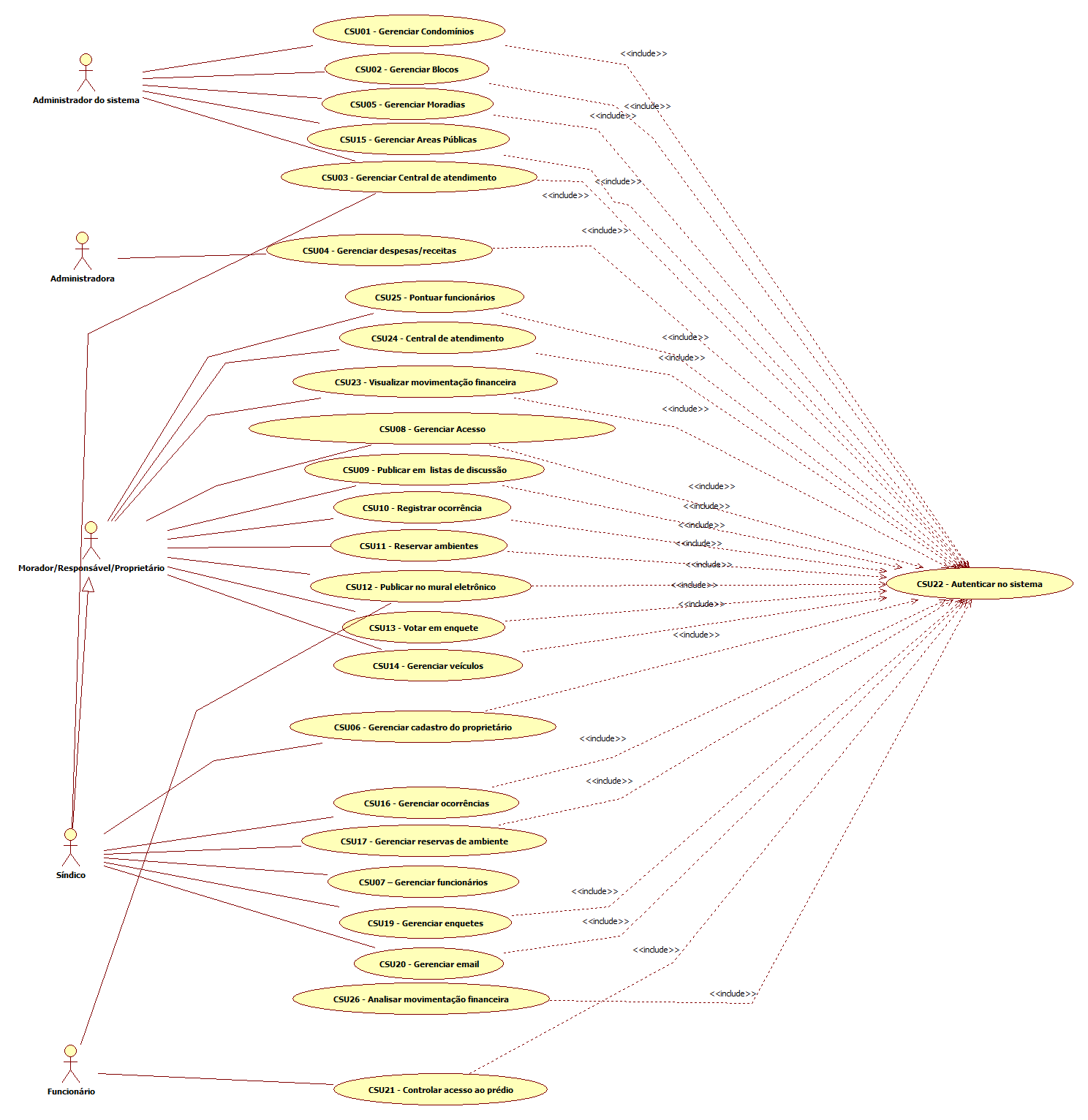


Figura 1 – Diagrama de Casos de Uso

## Realizações de Casos de Uso

A descrição de cada caso de uso contido no diagrama da Figuras 1 encontra-se detalhada no documento CSU – Descrição a Nível de Usuário.

# Visão Lógica

## Visão Geral

A visão lógica do Residential Life é composta principalmente por três pacotes:

* Dados: Esse pacote representa a implementação da parte lógica do domínio da aplicação. Aqui serão armazenadas classes que representação entidades da aplicação e classes responsáveis pela persistência dos dados da aplicação;
* Views: Nesse pacote são armazenados os componentes que fazem parte da interface gráfica da aplicação;
* Negócio: Nesse pacote são armazenados os componentes do tipo Controller. Esses componentes são responsáveis por integrar os Models com as Views, de acordo com a interação com usuário.

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



Figura 4 – Diagrama de Pacotes

## Diagrama de componentes

Figura 5 – Diagrama de Diagrama de Componentes

# Visão de Processos

À ser definida.

# Visão de Implantação

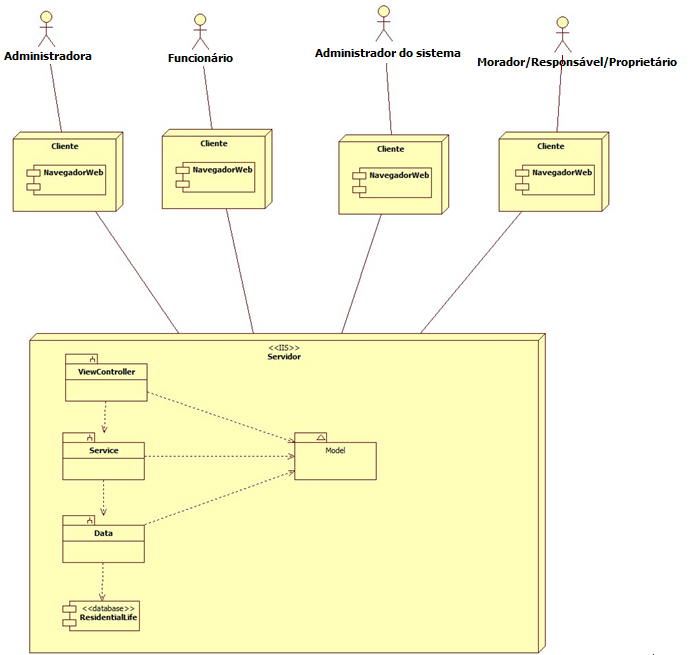
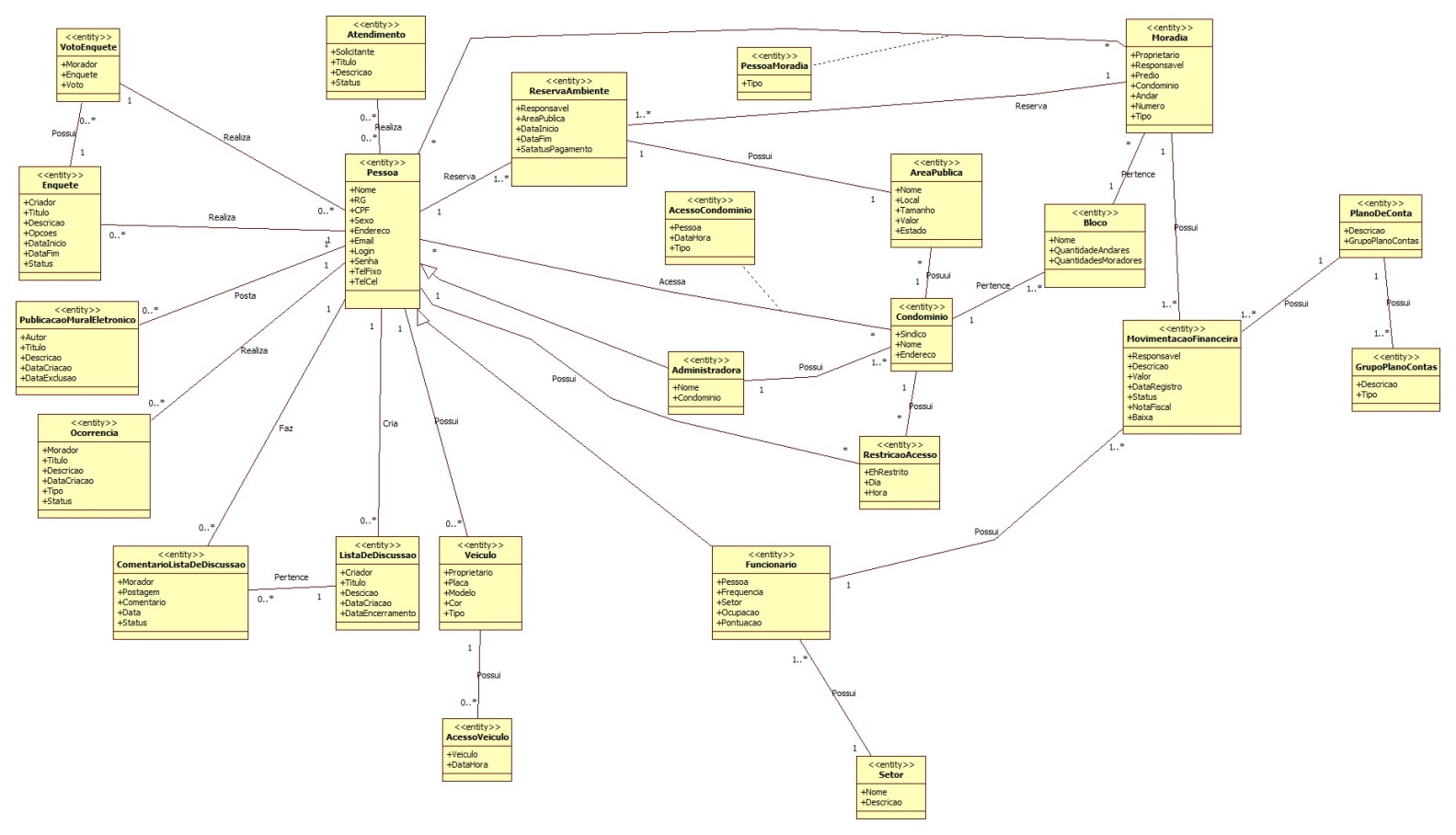


Figura 6 – Diagrama de Implantação

# Visão da Implementação

A visão de implementação encontra-se detalhada no documento Guia de Análise e Projeto.

# Visão de Dados (opcional)



# 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

O tratamento de exceções se dará em camadas. O sistema terá tratamento de exceções específicas definidas em *Application.Exception* utilizará os tratamentos de exceções genéricos definidos em *System.Exception.*

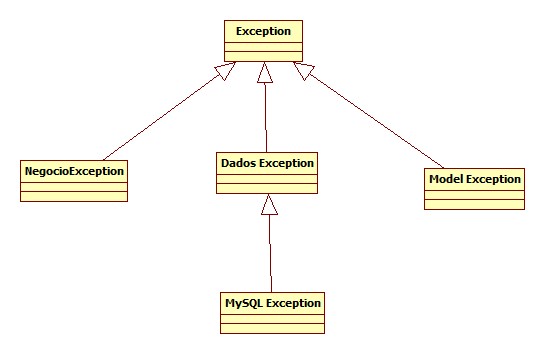


Figura 7 – Diagrama da Arquitetura de Exceções