

Documento de Arquitetura de Software

[Sigla] – [Nome do Projeto]

Gestor do Projeto	Gerente de Projeto
[nome]	[nome]
[email]	[email]
[telefone]	[telefone]

Objetivo deste Documento

Este documento tem como objetivo descrever as principais decisões de projeto tomadas pela equipe de desenvolvimento e os critérios considerados durante a tomada destas decisões. Suas informações incluem a parte de *hardware* e *software* do sistema.

Histórico de Revisão

Data	Demanda	Autor	Descrição	Versão
[dd/mm/aaaa]	[XX000000]	[Nome do autor]	[Descrever as principais alterações realizadas no documento, evidenciando as seções ou capítulos alterados]	[X]

Sumário

1. INTRODUÇÃO	3
1.1 Finalidade	3
1.2 Escopo	3
1.3 Definições, Acrônimos e Abreviações	3
1.4 Referências	4
2. REPRESENTAÇÃO ARQUITETURAL	4
3. REQUISITOS E RESTRIÇÕES ARQUITETURAIS	4
4. VISÃO DE CASOS DE USO	5
4.1 Casos de Uso significantes para a arquitetura.....	5
5. VISÃO LÓGICA.....	6
5.1 Visão Geral – pacotes e camadas	6
6. VISÃO DE IMPLEMENTAÇÃO	8
6.1 Caso de Uso [00X].....	8
6.1.1 Diagrama de Classes.....	8
6.1.2 Diagrama de Sequência	9
7. VISÃO DE IMPLANTAÇÃO	10
8. DIMENSIONAMENTO E PERFORMANCE	10
8.1 Volume	10
8.2 Performance.....	10
9. QUALIDADE	10

1. INTRODUÇÃO

1.1 Finalidade

Este documento fornece uma visão arquitetural abrangente do sistema [nome do sistema], 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.

O documento irá adotar uma estrutura baseada na visão “4+1” de modelo de arquitetura [KRU41].

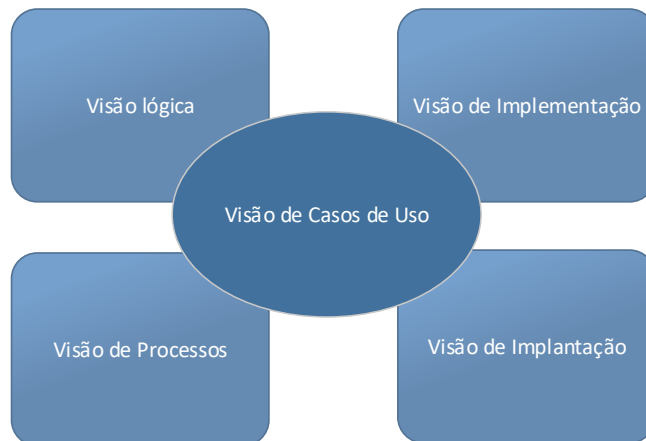


Figura 1 – Arquitetura 4+1

1.2 Escopo

Este Documento de Arquitetura de Software se aplica ao [nome do sistema], que será desenvolvido pela [área / equipe].

1.3 Definições, Acrônimos e Abreviações

QoS – Quality of Service, ou qualidade de serviço. Termo utilizado para descrever um conjunto de qualidades que descrevem as requisitos não-funcionais de um sistema, como performance, disponibilidade e escalabilidade[QOS].

1.4 Referências

- [KRU41]: The “4+1” view model of software architecture, Philippe Kruchten, November 1995, <http://www3.software.ibm.com/ibmdl/pub/software/rational/web/whitepapers/2003/Pbk4p1.pdf>
- [QOS] <https://docs.oracle.com/cd/E19636-01/819-2326/6n4kfe7dj/index.html>

2. REPRESENTAÇÃO ARQUITETURAL

Este documento irá detalhar as visões baseado no modelo “4+1” [KRU41], utilizando como referência os modelos definidos na MDS. As visões utilizadas no documento serão:

Visão	Público	Área	Modelo da MDS
Lógica	Analistas	Realização dos Casos de Uso	
Processo	Integradores	Performance, Escalabilidade, Concorrência	
Implementação	Programadores	Componentes de Software	
Implantação	Gerência de Configuração	Nodos físicos	
Caso de Uso	Todos	Requisitos funcionais	
Dados	Especialistas em dados Administradores de dados	Persistência de dados	

Tabela 1 – Visões, Público, Área e Artefatos da MDS

3. REQUISITOS E RESTRIÇÕES ARQUITETURAIS

Esta seção descrever os requisitos de software e restrições que tem um impacto significativo na arquitetura.

Requisito	Solução
Linguagem	[Especificar a(s) linguagem(ns) utilizada(s) no desenvolvimento.]
Plataforma	[Especificar o servidor de aplicações utilizado.]
Segurança	[Especificar a necessidade de segurança e as características básicas da segurança.]
Persistência	[Especificar a necessidade de persistência e qual o mecanismo de persistência que será adotado.]
Internacionalização (i18n)	[Especificar a necessidade de internacionalização/localização na aplicação.]

Tabela 2 – Exemplo de requisitos e restrições

4. VISÃO DE CASOS DE USO

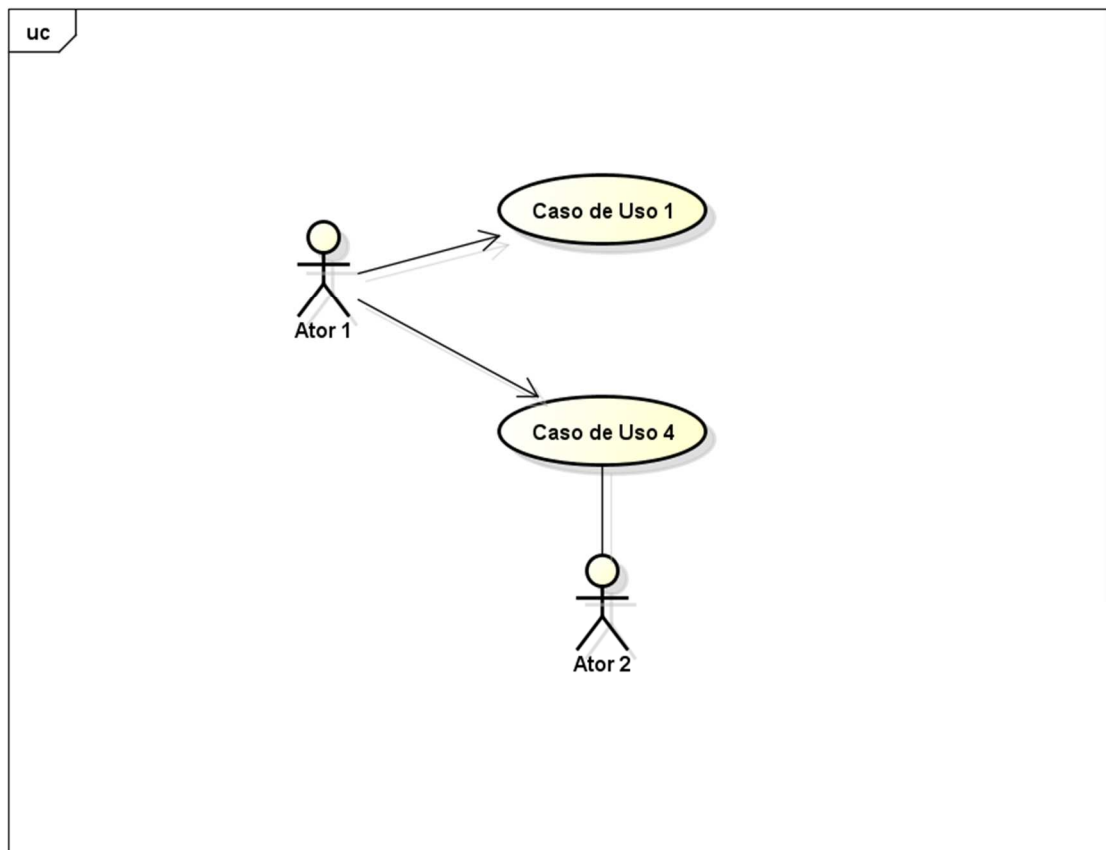
Esta seção lista as especificações centrais e significantes para a arquitetura do sistema.

Lista de casos de uso do sistema:

- Caso de Uso [00X]
- Caso de Uso [00X]

4.1 Casos de Uso significantes para a arquitetura

[Exemplo:



powered by Astah

Figura 2 – Exemplo de Diagrama com os casos de uso significantes e atores

5. VISÃO LÓGICA

Descrever uma visão lógica da arquitetura. Descrever as classes mais importantes, sua organização em pacotes de serviços e subsistemas, e a organização desses subsistemas em camadas. Também descreve as realizações dos casos de uso mais importantes, por exemplo, aspectos dinâmicos da arquitetura. Diagramas de classes e sequência devem ser incluídos para ilustrar os relacionamentos entre as classes significativas na arquitetura, subsistemas, pacotes e camadas.

5.1 Visão Geral – pacotes e camadas

[Exemplo:

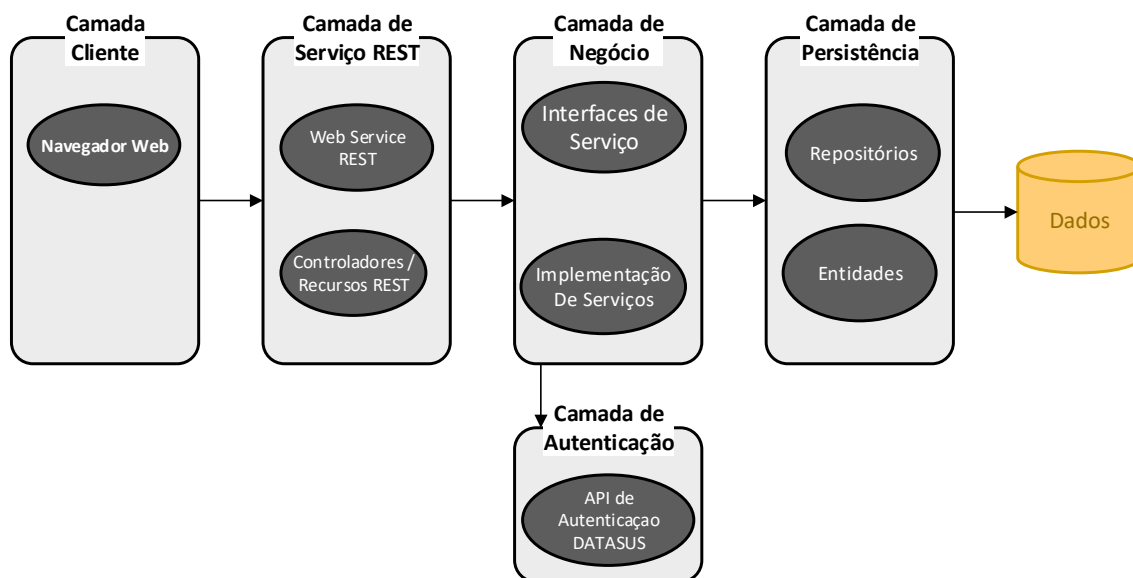
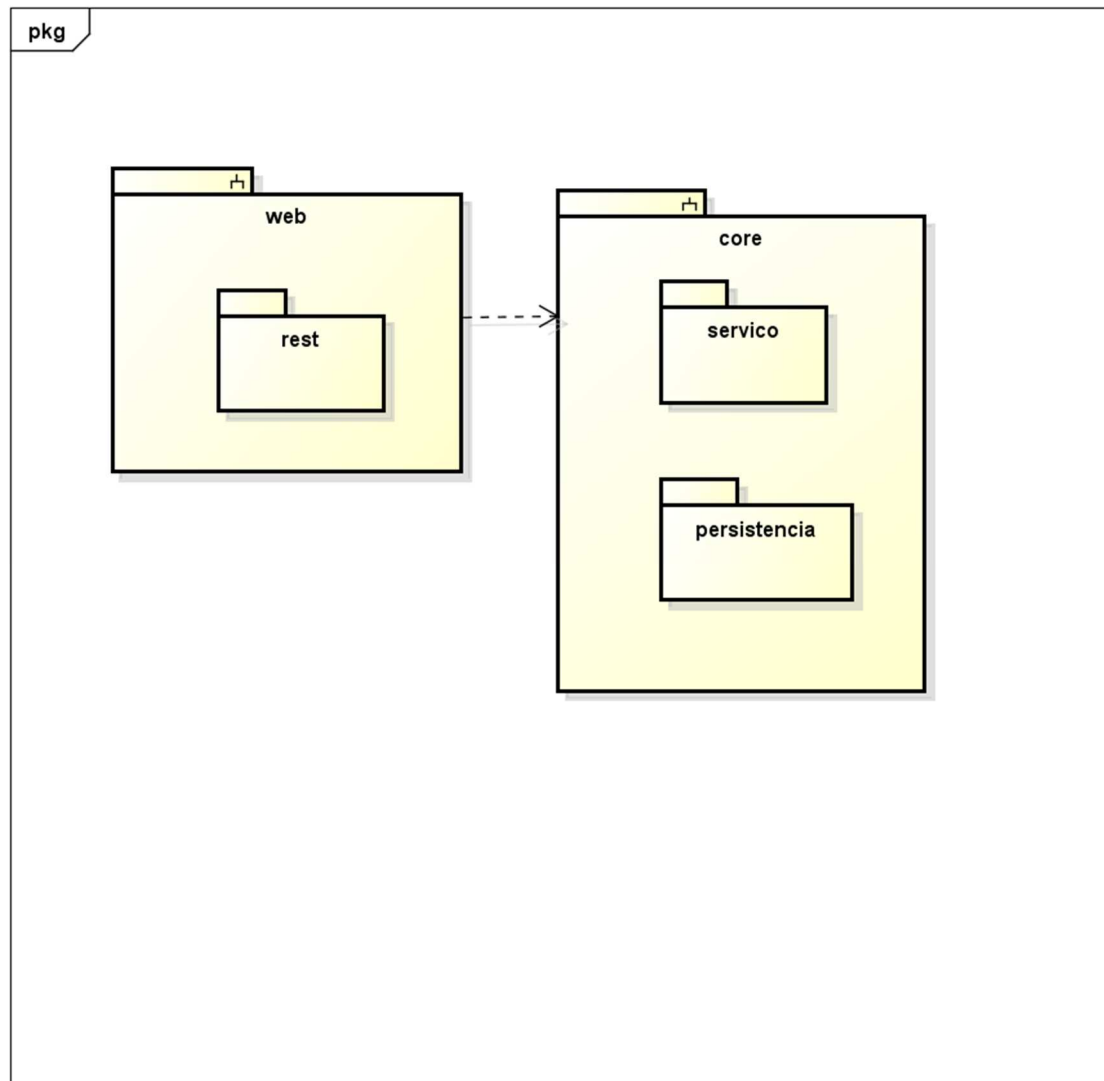


Figura 2 – Exemplo de Diagrama de Camadas da Aplicação



powered by Astah

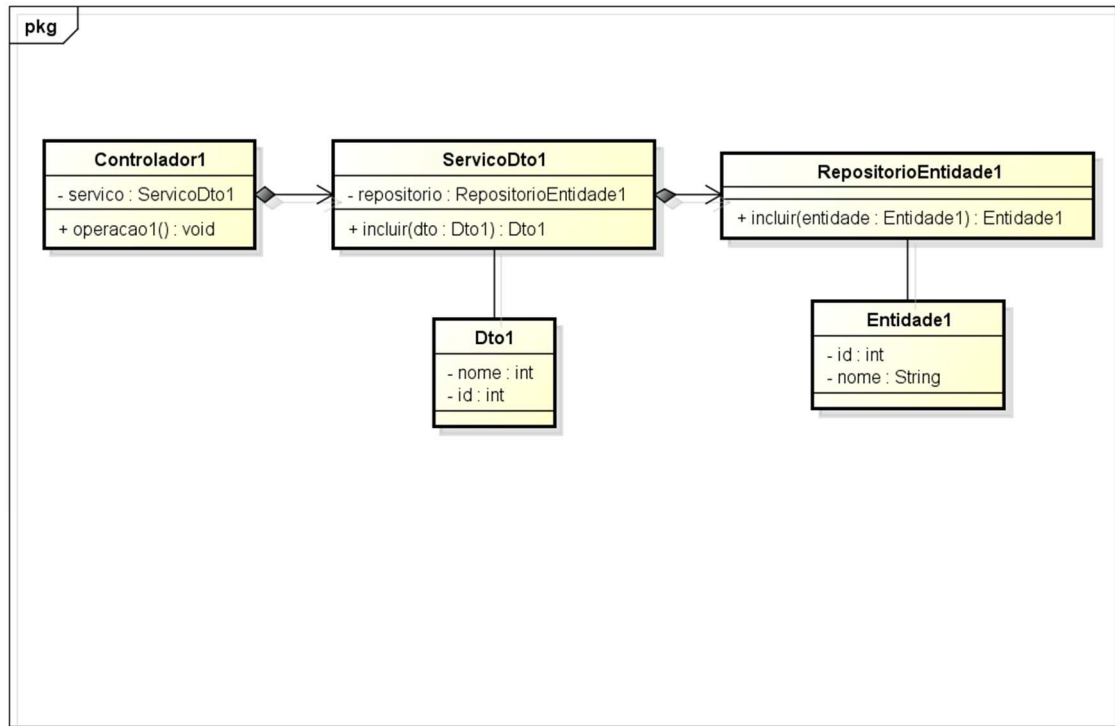
Figura 3 – Exemplo de Diagrama de Pacotes da Aplicação

6. VISÃO DE IMPLEMENTAÇÃO

6.1 Caso de Uso [00X]

6.1.1 Diagrama de Classes

[Exemplo:



powered by Astah

Figura 20 – Exemplo de Diagrama de Classes

6.1.2 Diagrama de Sequência

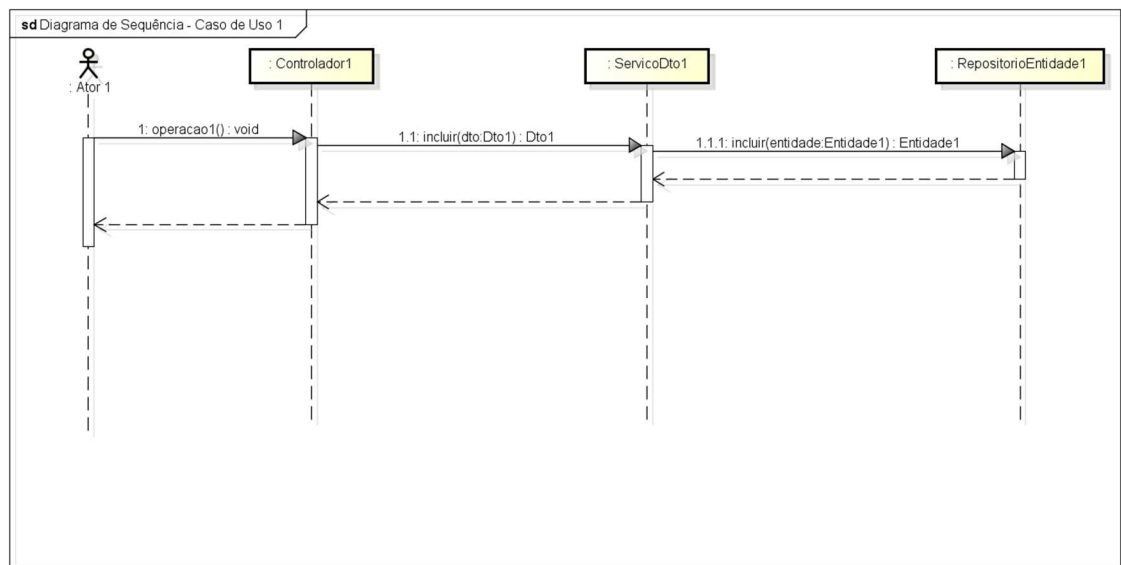


Figura 20 – Exemplo de Diagrama de Sequência

7. VISÃO DE IMPLANTAÇÃO

Descrever os nodos físicos, as configurações e os artefatos que serão implantados.

[Exemplo:

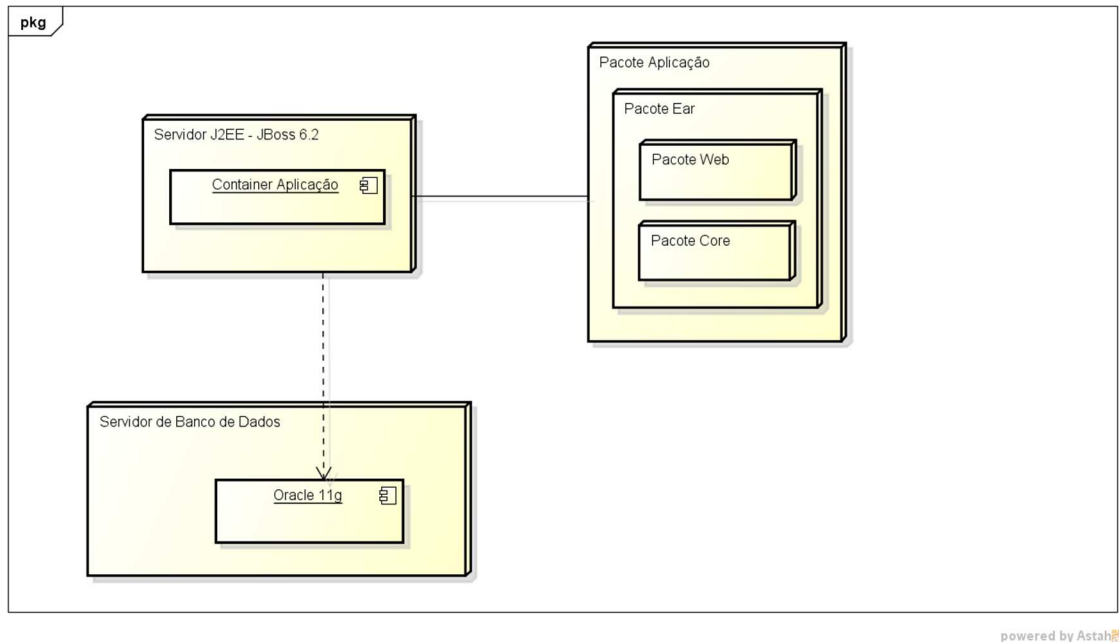


Figura 20 – Exemplo de Diagrama de Implantação Java

8. DIMENSIONAMENTO E PERFORMANCE

8.1 Volume

Enumerar os itens relativos ao volume de acesso aos recursos da aplicação:

- Número de estimado usuários: [000]
- Número estimado de acessos diários: [000]
- Número estimado de acessos por período: [000]
- Tempo de sessão de um usuário: [000]

8.2 Performance

Enumerar os itens referentes à resposta esperada do sistema:

- Tempo máximo para a execução de determinada transação: [000]

9. QUALIDADE

Enumerar os itens de qualidade de software [QOS] significativos para a aplicação:

Item	Descrição	Solução
Escalabilidade	[Breve Descrição]	[Breve descrição da Solução]
Confiabilidade, Disponibilidade	[Breve Descrição]	[Breve descrição da Solução]
Portabilidade	[Breve Descrição]	[Breve descrição da Solução]
Segurança	[Breve Descrição]	[Breve descrição da Solução]