Documento de Arquitetura de Software

InvSCP – Inventory - Software para Controle do Patrimônio	
Gestor do Projeto	**Gerente de Projeto**
Elias Batista Ferreira	Hyaygo Vieira de Souza
eliasbf@gmail.com	hagosouzza@hotmail.com

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	Autor	Descrição	Versão
16/10/2018	Estevão Cristino da Silva e Hyago Vieira de Souza	Criação da Documentação de Arquitetura	1.0

Sumário

1. INTRODUÇÃO

1.1 Finalidade

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

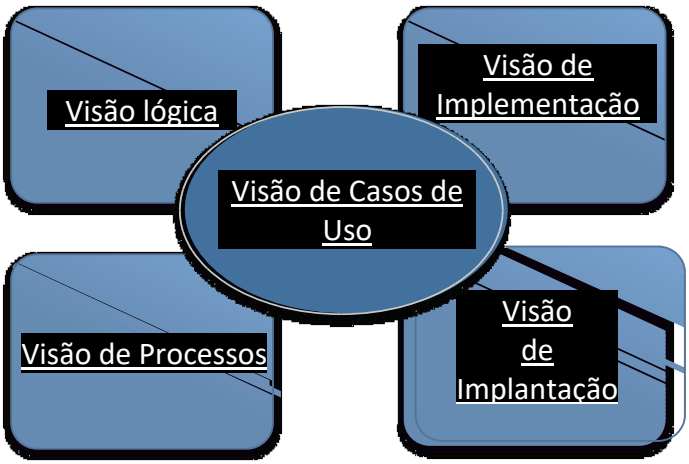

Figura 1 – Arquitetura 4+1

1.2 Escopo

Este Documento de Arquitetura de Software se aplica ao *Inventory*, que será desenvolvido pelo grupo de docentes do Instituto de Informática (UFG) formado por Estevão Silva, Gabriel Menezes, Hyago Souza, João Pedro Pinheiro e Pedro Henrique Coimbra.

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

Arquitetura escolhida Cliente-Servidor com Quatro Camadas (4-Tier)

Analisando os requisitos do software, o sistema será uma aplicação web, a arquitetura escolhida foi a Cliente-Servidor com três camadas.

A arquitetura é dividida em 4 camadas:

- Camada de apresentação: chamada de GUI(Graphical User Interface) que será a camada de interação do usuário com o sistema através de requisições e consultas.

- Camada de Comunicação: Essa camada é responsável por gerir toda comunicação REST entre o cliente e o servidor da aplicação via protocolo HTTP.

- Camada de Negócio: É nessa camada que ficará todas as funções de regras de todo o negócio da ***Inventory***. Localizada no servidor Tomcat 9, terá a responsabilidade gerir todas as requisições de forma segura e sem ferir as regras de negócio e segurança.

- Camada de Dados: Composta pelo repositório das informações e as classes que as manipulam. Tem a responsabilidade de receber as requisições da camada de negócios e as

executam no SGBD PostgreSQL. Uma alteração no banco de dados alteraria apenas nas classes desta camada, logo não afetaria as outras camadas.

As 4 camadas da aplicação, separa as responsabilidades e principalmente a parte lógica da apresentação, assim oferecendo mais segurança pois retiramos as responsabilidades da parte do cliente.

A separação em camadas lógicas torna o sistema mais flexível, permitindo que as camadas possam ser alteradas de forma independentes, o que ajuda manutenção futura do sistema.

As camadas podem ainda ser fatorada, em pacotes ou componentes, reduzindo a dependência entre as classes e pacotes, o que oferece uma coesão e menos acoplamento.

Oferece reuso dos componentes do sistema em diferentes partes do sw.

É a arquitetura mais usada no mundo para sistemas corporativos baseados na web, logo temos bastante recursos e frameworks prontos que podem ajudar a compor o sistema.

2.1 Visão Geral da Arquitetura

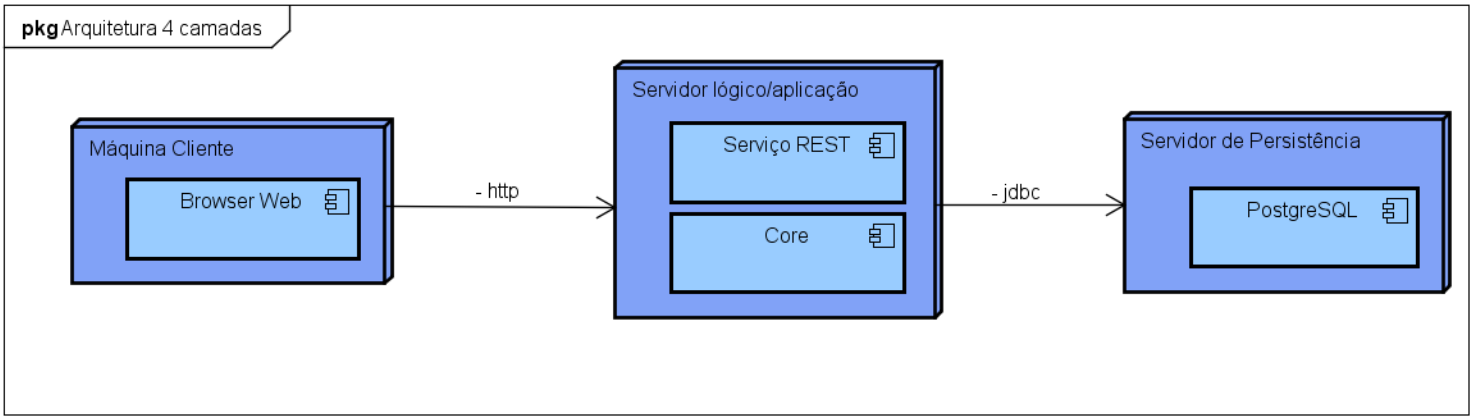

Figura 1. – Visão da Arquitetura

3. REQUISITOS E RESTRIÇÕES ARQUITETURAIS

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

Requisito	Solução
Linguagem	Java 10, TypeScript, PL/SQL, HTML5, CSS3 e JavaScript
Plataforma	Windows e Linux
Segurança	Autenticação de usuário com JWT
Persistência	PostegreSQL via JDBC.

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:

- **CS-RF01 – Movimentação de bem patrimonial (MBP)**
- **CS-RF03 – Registrar Aceite de Saída da Movimentação**
- **CS-RF04 – Registrar Aceite de Entrada do Bem Patrimonial**
- **CS-RF05 – Cancelar Movimentação (MBP)**
- **CS-RF06 – Emitir guia de autorização de transporte**
- **CS-RF07 – Emitir relatório de bens patrimoniais da seção**
- **CS-RF08 – Registrar ordem de serviço**
- **CS-RF09 – Registrar conclusão da ordem de serviço**
- **CS-RF10 – Visualizar histórico do bem patrimonial**
- **CS-RF16 – Emitir inventário**
- **CS-RF17 – Baixar bem patrimonial**
- **CS-RF19 – Pesquisar bem patrimonial usando número de tombamento, denominação ou marca como critério de busca (filtro).**

4.1 Diagrama de Caso de Uso

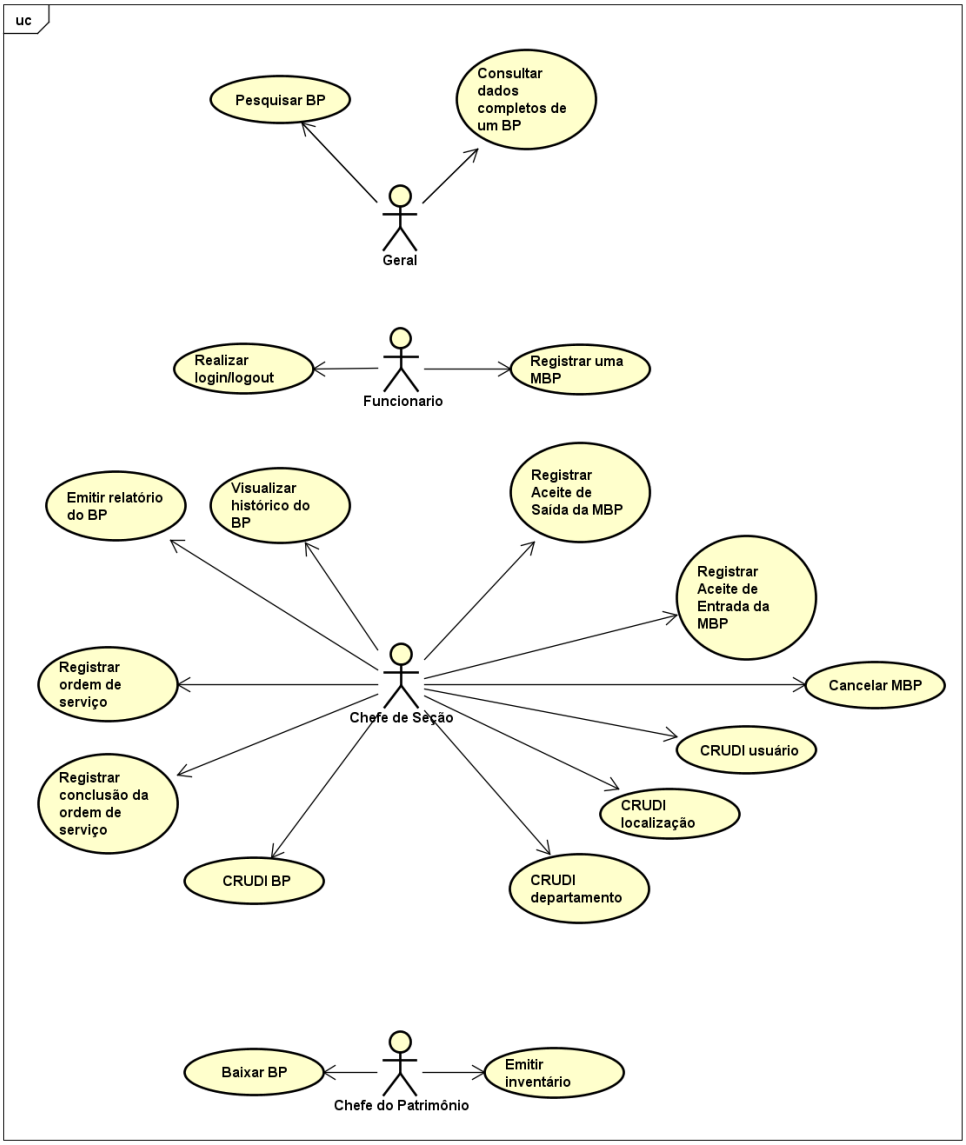

Figura 2 – Diagrama com os casos de uso

5. VISÃO LÓGICA

5.1 Visão Geral – pacotes e camadas

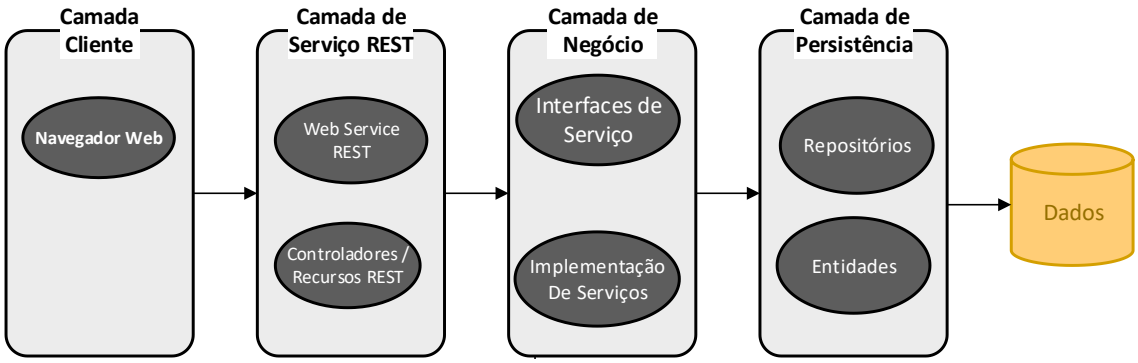

Figura 2.1 – Diagrama de Camadas da Aplicação

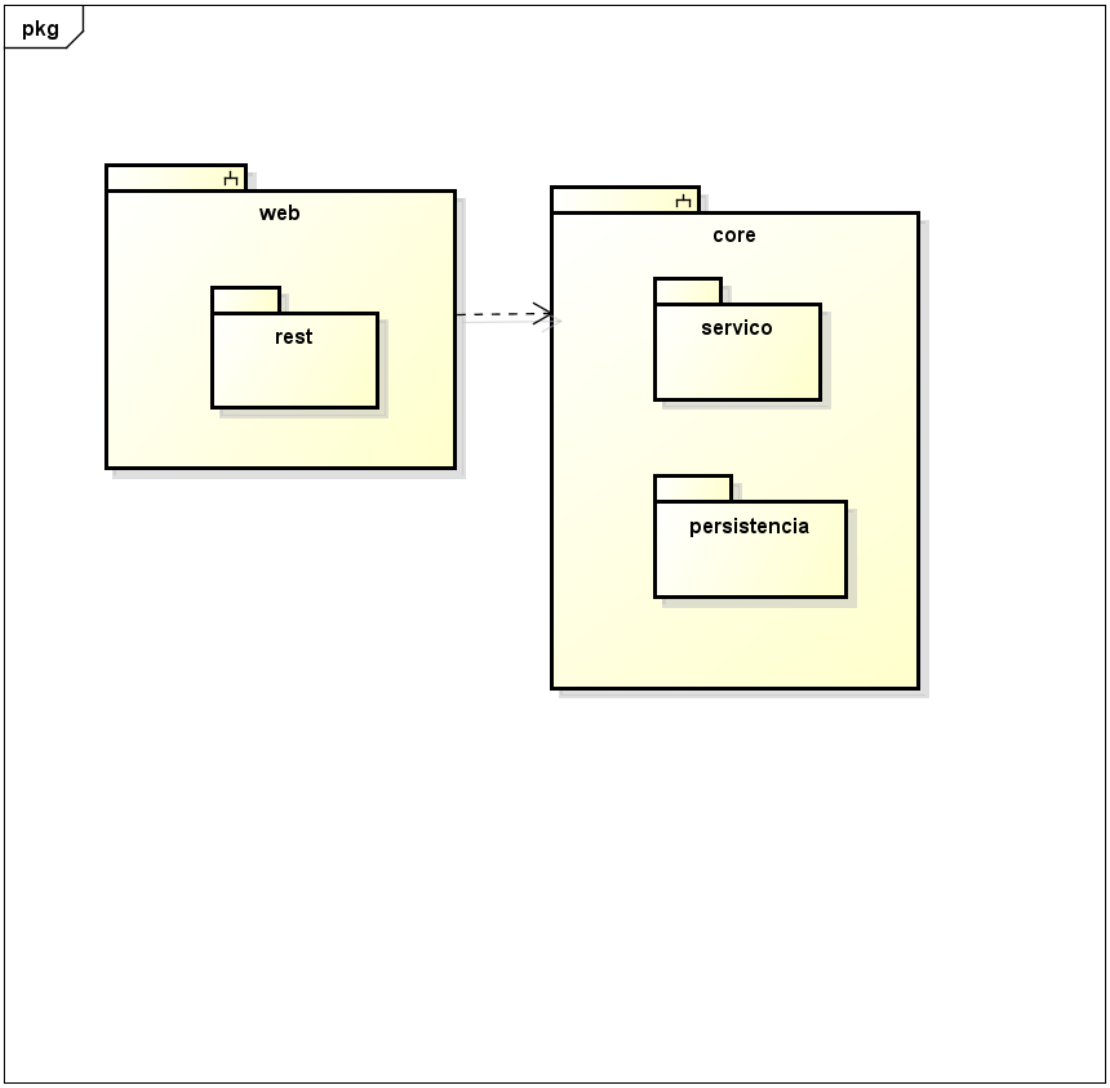

Figura 3 – Diagrama de Pacotes da Aplicação

6. VISÃO DE IMPLEMENTAÇÃO

6.1 Diagrama de Classes

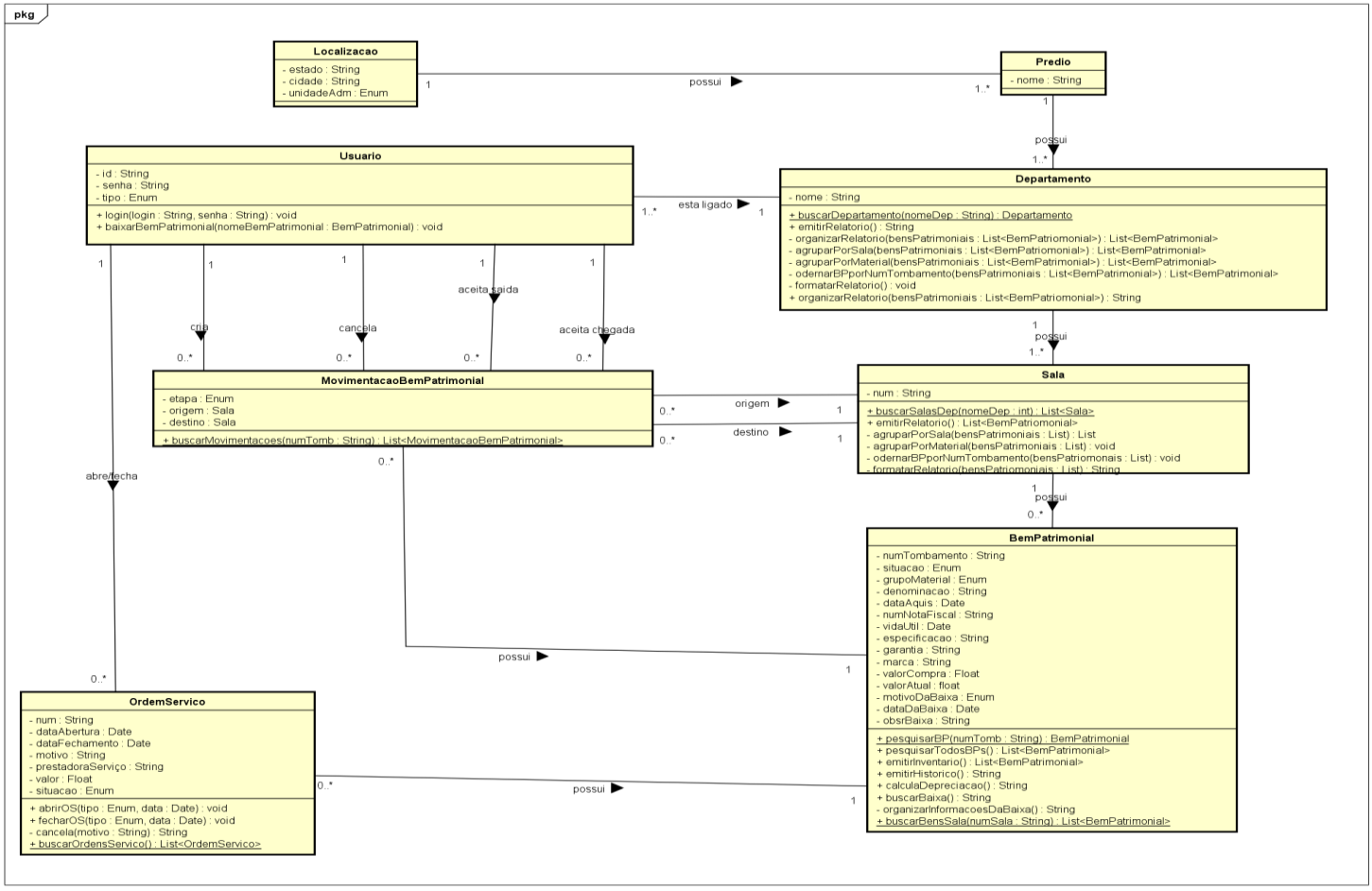

Figura 20 – Exemplo de Diagrama de Classes

6.1.1 CS-RF01

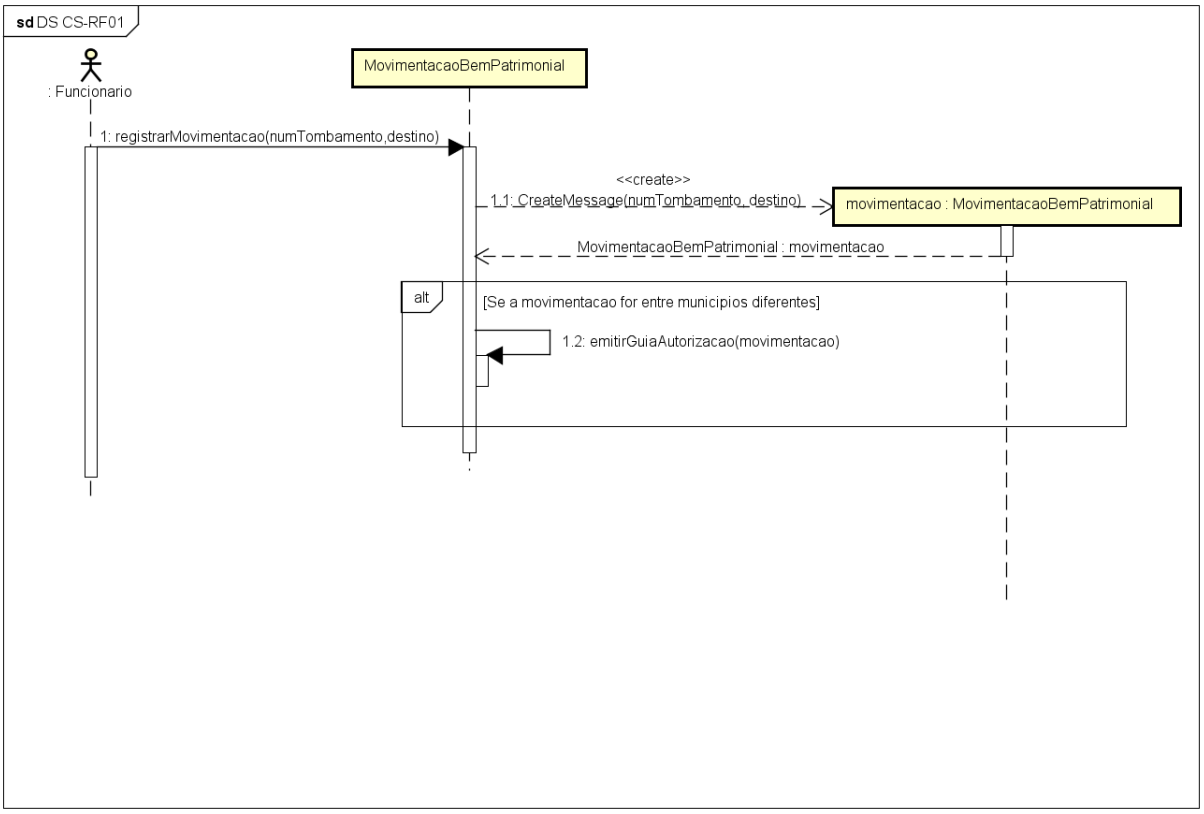

Figura 20 – Diagrama de Sequência Caso de Uso 01

6.1.2 CS-RF03

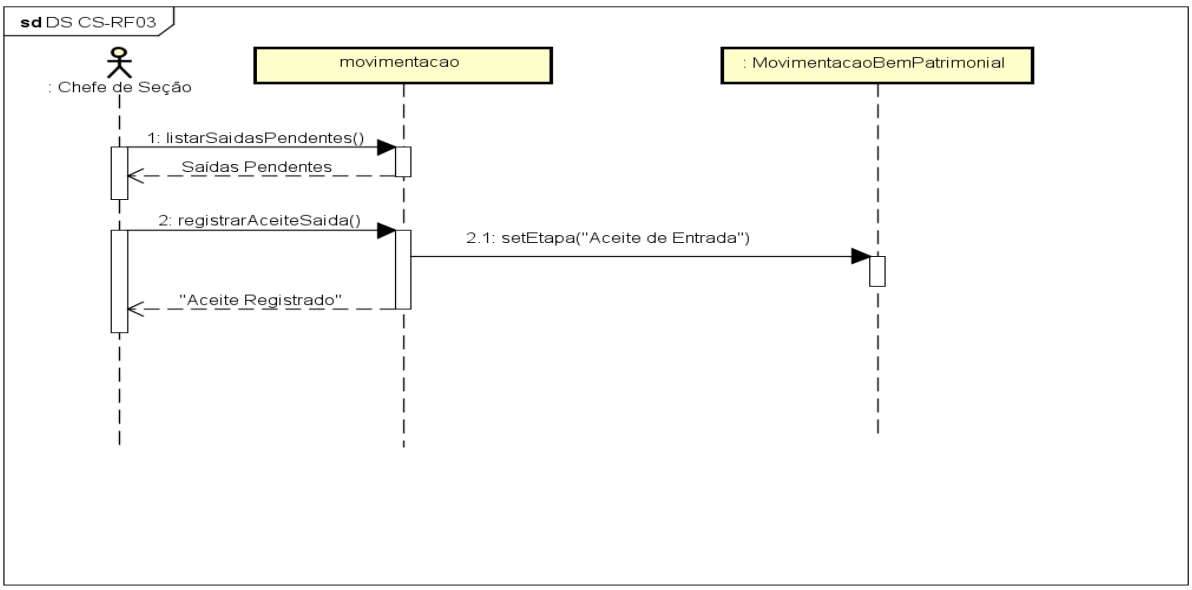

Figura 21 – Diagrama de Sequência Caso de Uso 03

6.1.3 CS-RF04

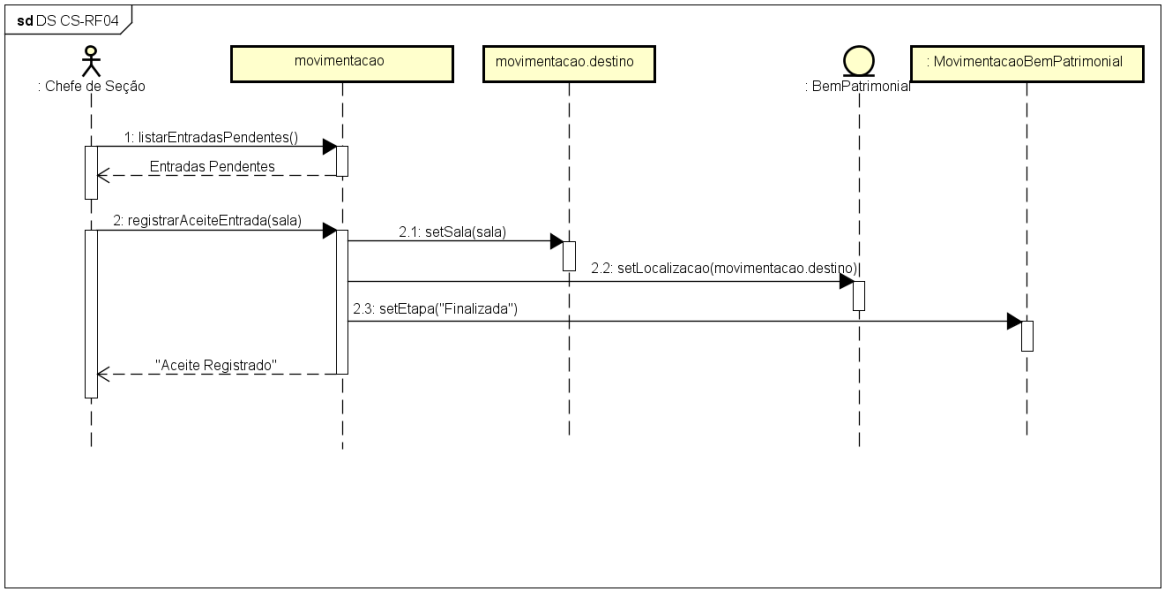

Figura 22 – Diagrama de Sequência Caso de Uso 04

6.1.4 CS-RF05

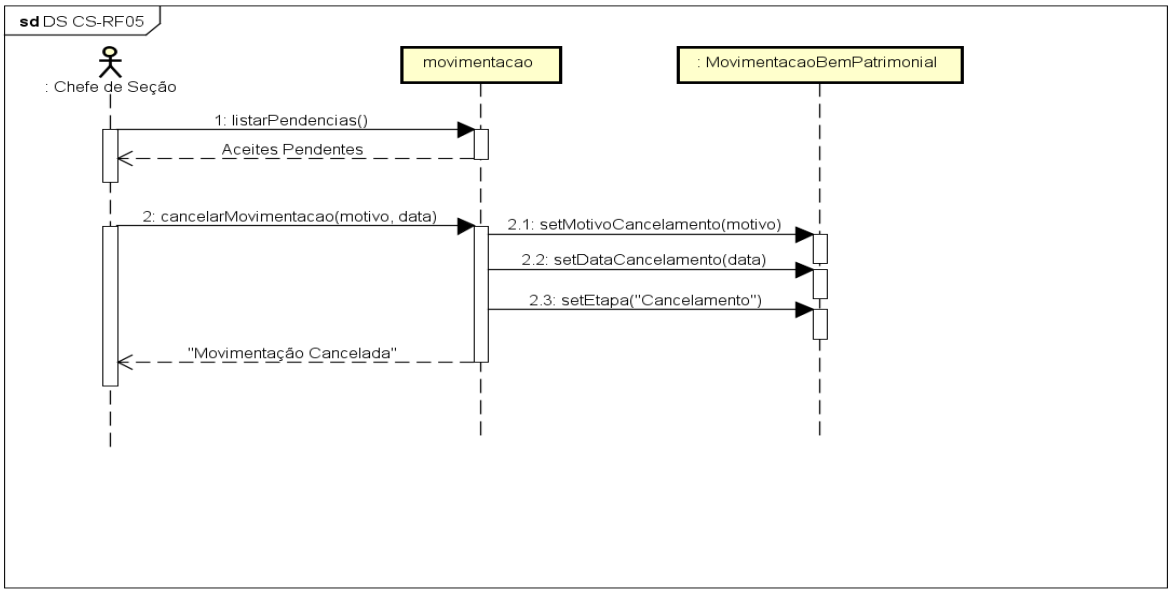

Figura 23 – Diagrama de Sequência Caso de Uso 05

6.1.5 CS-RF07

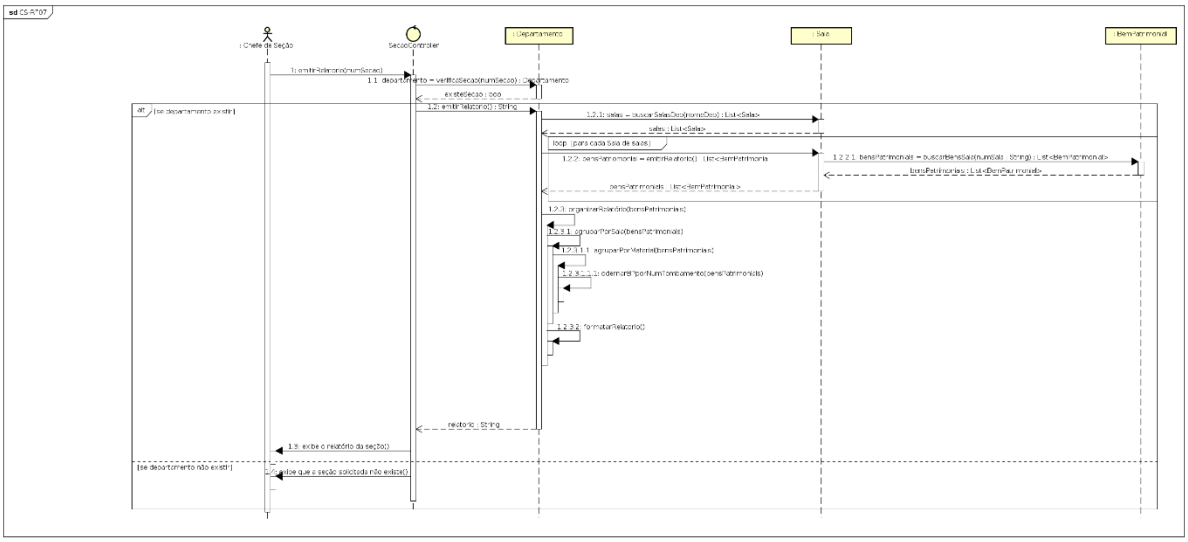

Figura 24 – Diagrama de Sequência Caso de Uso 07

6.1.6 CS-RF10

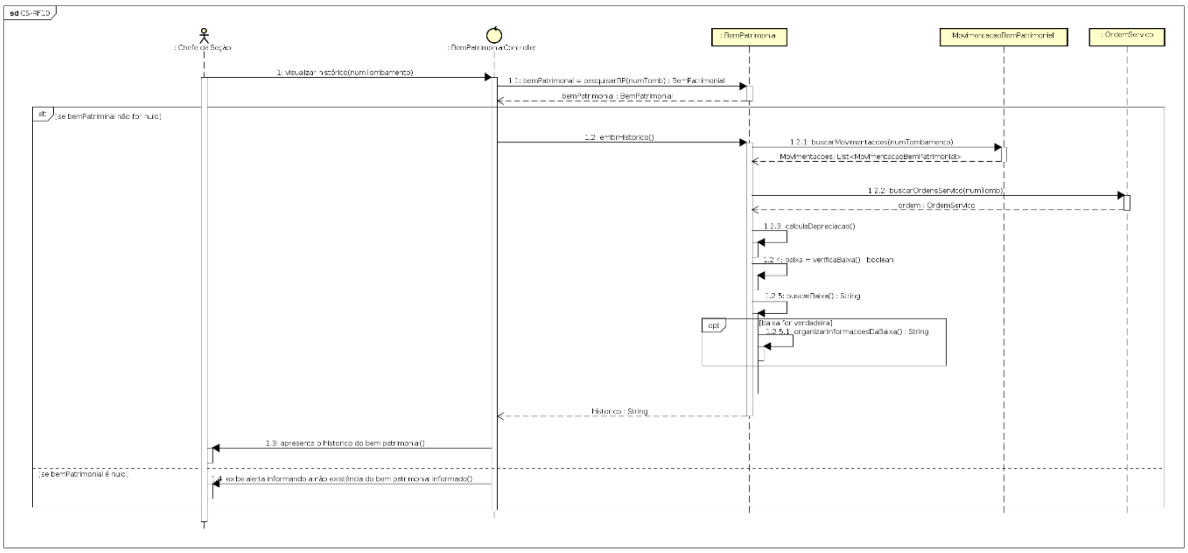

Figura 25 – Diagrama de Sequência Caso de Uso 10

6.1.7 CS-RF16

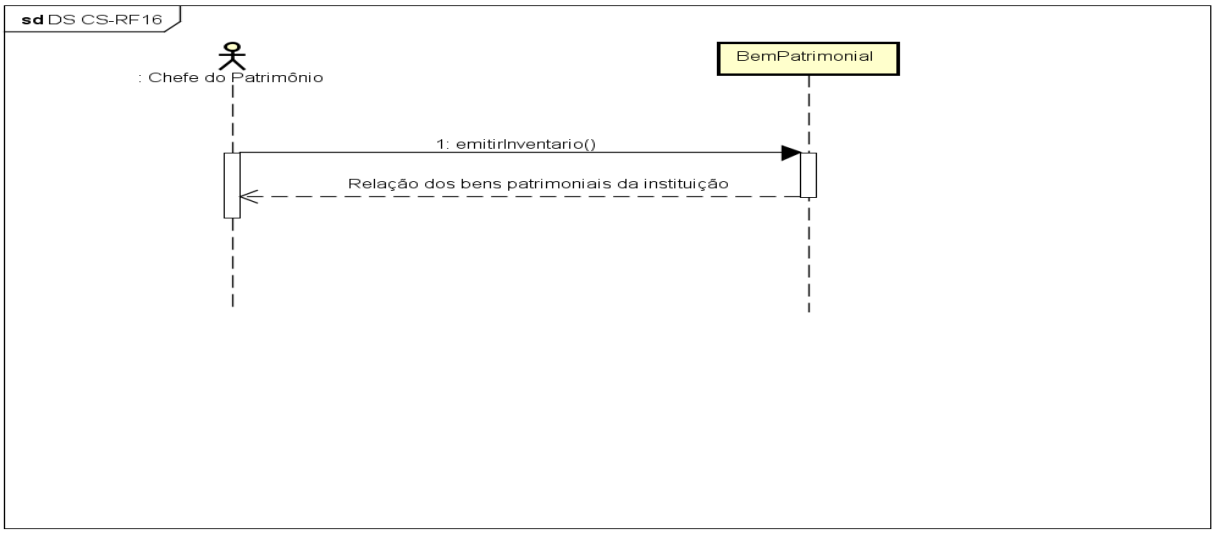

Figura 26 – Diagrama de Sequência Caso de Uso 16

6.1.8 CS-RF17

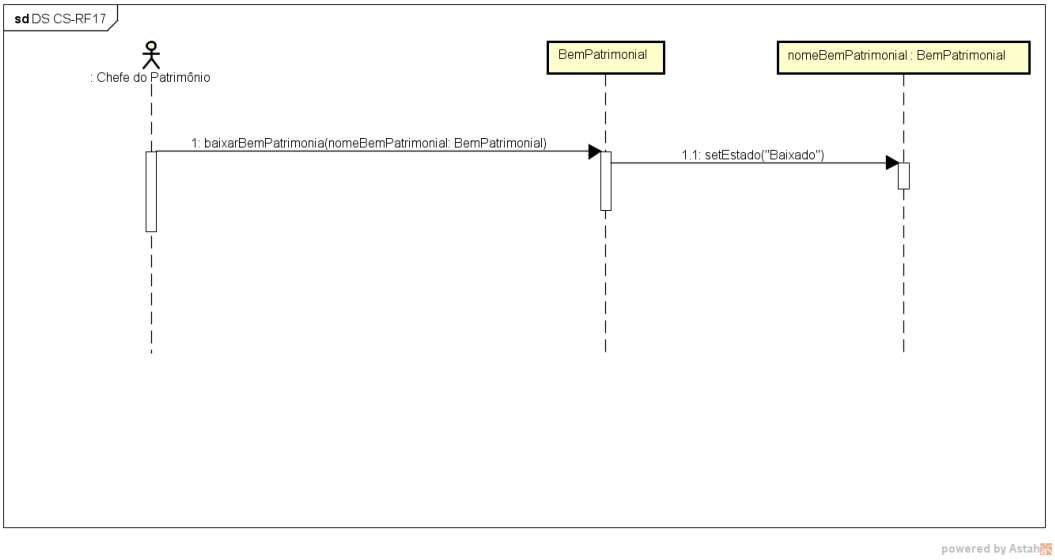

Figura 27 – Diagrama de Sequência Caso de Uso 17

7. VISÃO DE IMPLANTAÇÃO

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

[Exemplo:

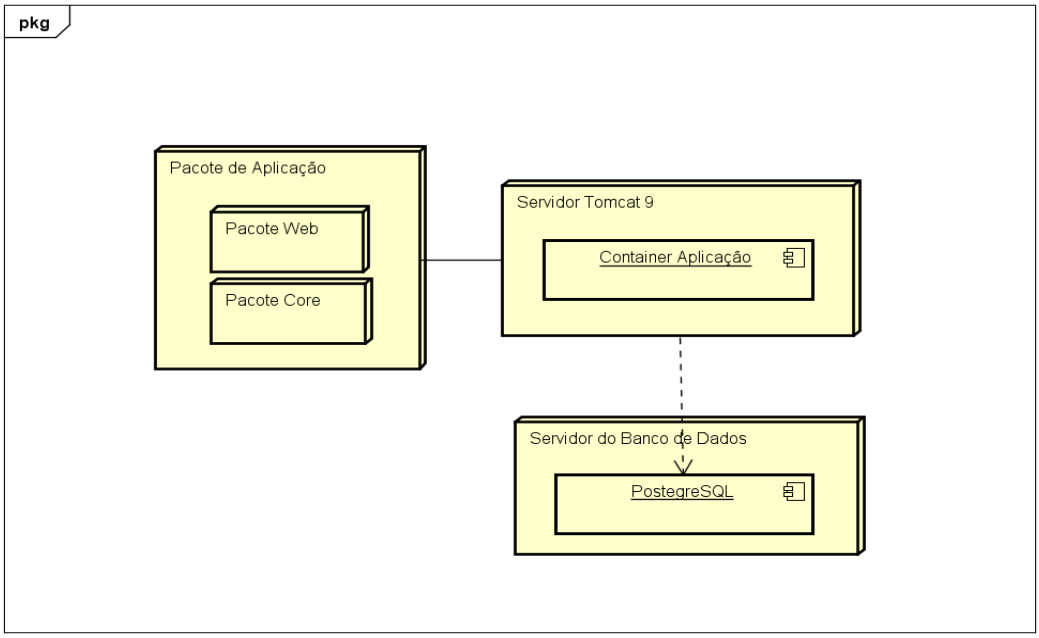

Figura 28 – Diagrama de Implantação Java

8. DIMENSIONAMENTO E PERFORMANCE

8.1 Volume

- Número de estimados usuários: 400
- Número estimado de acessos diários: 50
- Número estimado de acessos por período: 20
- Tempo de sessão de um usuário: 20

8.2 Performance

Enumerar os itens referentes à resposta esperada do sistema:

- Tempo máximo para a execução de determinada transação: 30