

Inventory – Software para Controle do Patrimônio
Documento de Arquitetura de Software

Versão <1.1>

Histórico da Revisão

Data	Versão	Descrição	Autor
16/102018	1.0	Criação do documento de Arquitetura	Rodolpho Hiroshi Takahashi
21/10/2018	1.1	Alterações e Complementações	Rafael Reis

Índice Analítico

1. Introdução	
1.1 Finalidade	4
1.2 Escopo	4
1.3 Visão Geral	4
2. Representação Arquitetural	
2.1 Metas e Restrições	5
3. Perspectivas - Pontos de Vista (ViewPoints)	
3.1 Casos de Uso	
3.2 Lógico	5
3.3 Implementação	6
3.4 Processos	7
3.5 Implantação	8

Documento de Arquitetura de Software

1. Introdução

1.1 Finalidade

Este documento possui como objetivo definir os aspectos da Arquitetura do Inventory – Software para Controle do Patrimônio e é direcionado aos stakeholders do software a ser desenvolvido, tais como: Gerentes do Projeto, Clientes e equipe técnica, possuindo grande foco para os Desenvolvedores e a Equipe de implantação.

1.2 Escopo

Este documento se baseia no documento de requisitos Sistema: InvSCP para definir os atributos de qualidade a serem priorizados, bem como, os estilos arquiteturais que favorecem tais atributos e as representações das visões arquiteturais e seus sub-produtos.

1.3 Visão Geral

Os próximos tópicos descrevem quais serão os requisitos e restrições utilizados para definir a arquitetura a ser implementada, bem como, quais atributos de qualidades serão priorizados e o porquê da escolha. Quais os padrões arquiteturais serão utilizados conforme os atributos de qualidade selecionados e como funcionará o trade-off entre esses padrões arquiteturais, bem como o porquê da escolha dos padrões arquiteturais. Quais e como as visões arquiteturais serão detalhadas e quais os pontos de vista da arquitetura serão utilizados para descrever as visões.

2. Representação Arquitetural

A Arquitetura do Software a ser desenvolvido, por escolha do professor, adotará o estilo arquitetural MVC. Para representar como ela será estruturada será adotado os pontos de vista do 4 + 1(RUP), formulados por Philippe Kruchten, e que considera que esses pontos de vista seriam o mínimo para representar uma arquitetura.

2.1 Metas e Restrições da Arquitetura

Nenhum orçamento para aquisição de serviços ou produtos está previsto para o projeto. Os recursos alocados ao projeto devem ser aqueles já disponíveis na Fábrica de Software. Essa restrição se aplica particularmente à utilização da mão-de-obra de discentes, que não poderá adicionar custo ao projeto.

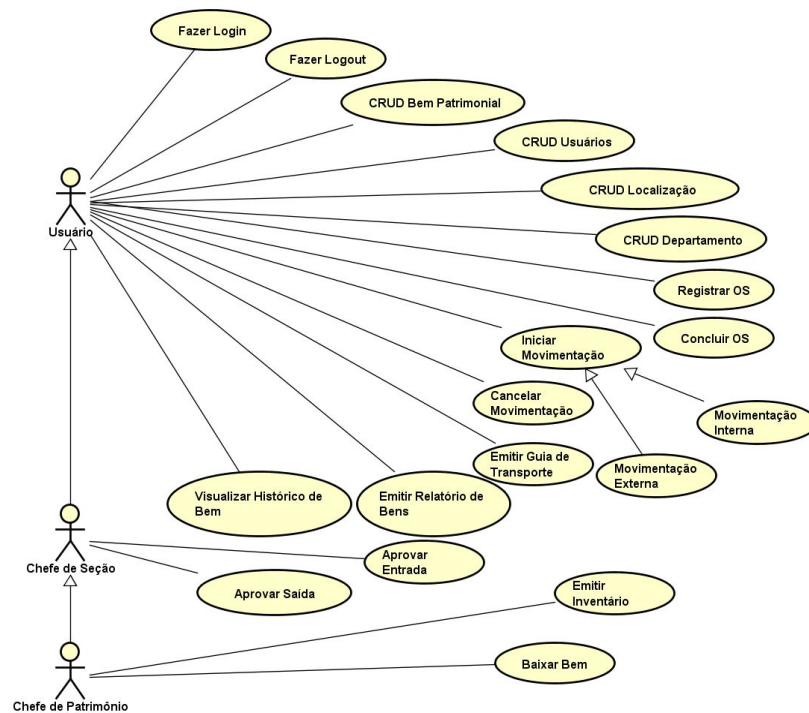
3. Perspectivas - Pontos de Vista (ViewPoints)

Para representar bem uma arquitetura, dada sua complexidade, torna-se necessário apresentá-la sob diferentes perspectivas. Para tanto, o critério mais utilizado para esse fim é separar pelos tipos de informações relevantes para cada interessado. Essa separação reduz a quantidade de informação que o arquiteto precisa apresentar/lidar ao apresentá-la. O RUP sugere alguns Pontos de vista considerados imprescindíveis para apresentar minimamente uma Arquitetura de um Software.

3.1 Casos de Uso

Para fornecer uma base para o planejamento da arquitetura e de todos os outros artefatos que serão gerados durante o ciclo de vida do software, é gerada, na análise de requisitos, uma visão chamada visão de casos de uso. Só existe uma visão de casos de uso para cada sistema. Ela ilustra os casos de uso e cenários que englobam o comportamento, as classes e riscos técnicos significativos do ponto de vista da arquitetura. A visão de casos de uso é refinada e considerada inicialmente em cada iteração do ciclo de vida do software.

Visão:

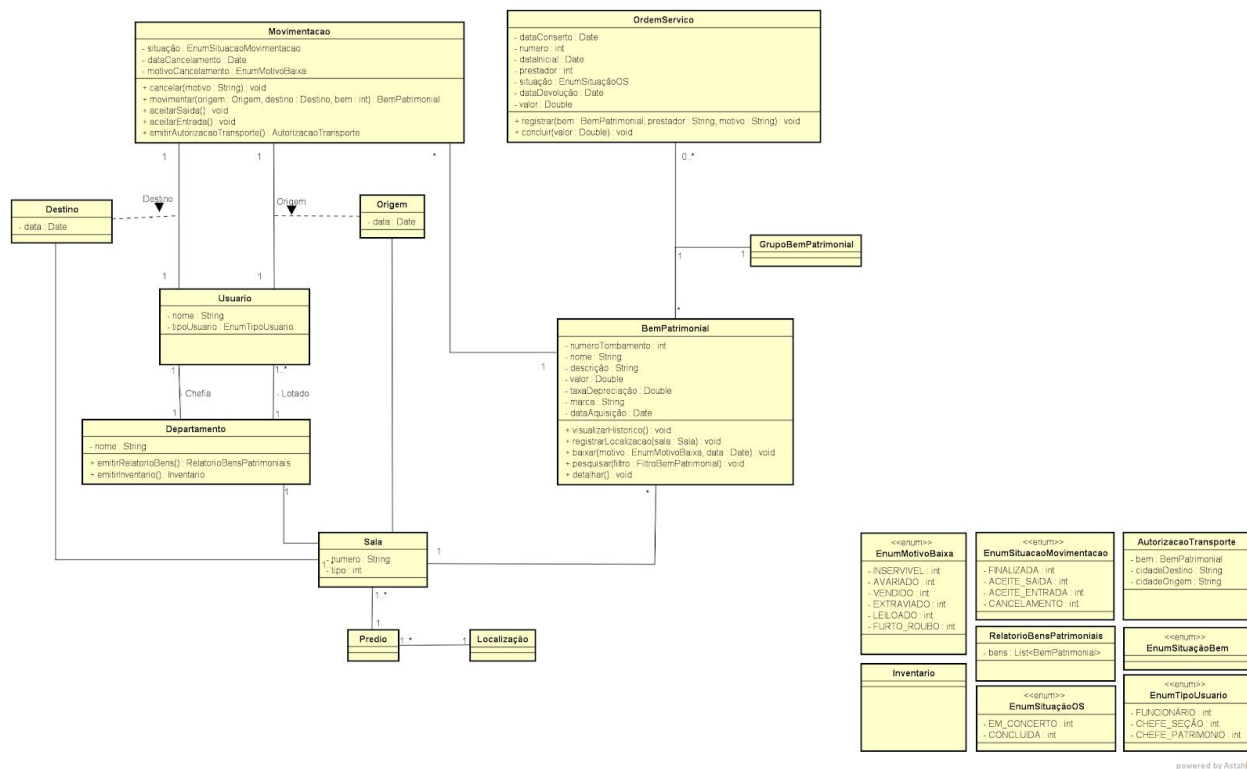


powered by Astah

3.2 Lógico

Descreve requisitos comportamentais e a decomposição do sistema em um conjunto de abstrações. Diagramas de classes, sequência e colaboração mostram os relacionamentos entre esses elementos.

Visões:



3.3 Desenvolvimento

Descreve os módulos do sistema. Módulos são elementos mais abstratos que classes e objetos. Pacotes e biblioteca de classes são exemplos de módulos em alguns ambientes de programação.

Visões:

3.4 Processos

Descreve os processos do sistema e como eles se comunicam. Útil quando se tem múltiplos processos ou threads concorrentes. Permite avaliar requisitos não funcionais relacionados à execução e comunicação: – Desempenho, disponibilidade. Diagramas de atividades são úteis para descrever esta visão.

Visões:

3.5 Implantação

Descreve como a aplicação é instalada e como executa em uma rede de computadores. Componentes executáveis são alocados a nós processadores. Permite avaliar requisitos não-funcionais - desempenho, disponibilidade, confiabilidade, escalabilidade.

Visões:

