
PONTU

PONTU
Documento de Arquitetura de Software

Versão 1.0

Histórico da Revisão

Data	Versão	Descrição	Autor
10/09/18	1.0	Introdução do documento	Rogério Amorim, Marcelo Alves, Hitallo Flavyo, Matheus França, Reinaldo Albernaz

Índice Analítico

Introdução

Finalidade

Escopo

Definições, Acrônimos e Abreviações

Referências

Visão Geral

Representação Arquitetural

Metas e Restrições da Arquitetura

Visão de Casos de Uso

Realizações de Casos de Uso

Visão Lógica

Visão Geral

Pacotes de Design Significativos do Ponto de Vista da Arquitetura

Visão de Processos

Visão de Implantação

Visão da Implementação

Visão Geral

Camadas

Visão de Dados (opcional)

Tamanho e Desempenho

Qualidade

Documento de Arquitetura de Software

1. Introdução

1.1 Finalidade

Este documento oferece uma visão geral arquitetural abrangente do sistema, 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.

1.2 Escopo

O documento de arquitetura de software se aplica ao Sistema de gerenciamento de ponto - PONTU que será desenvolvido por essa equipe.

1.3 Definições, Acrônimos e Abreviações

CSU - Caso de uso.

N/A - Não aplicável.

1.4 Referências

<https://github.com/RogérioAmorim/Arquitetura-PONTU>

1.5 Visão Geral

O Documento irá descrever a Representação arquitetural, as metas e restrições da arquitetura, as camadas arquiteturais, ou seja as visões 4+1(visão lógica, visão de processos, visão de implementação, visão de implantação e visão de casos de uso), e os dados do tamanho e restrições do desempenho.

2. Representação Arquitetural

[Esta seção descreve qual é a arquitetura de software do sistema atual e como ela é representada. Da Visão de Casos de Uso, Visão Lógica, Visão de Processos, Visão de Implantação e Visão de Implementação, enumera as visões necessárias e, para cada visão, explica quais tipos de elementos de modelo ela contém.]

3. Metas e Restrições da Arquitetura

[Esta seção descreve os requisitos e objetivos do software que têm algum impacto sobre a arquitetura; por exemplo, segurança, garantia, privacidade, uso de um produto desenvolvido internamente e pronto para ser usado, portabilidade, distribuição e reutilização. Ela também captura as restrições especiais que podem ser aplicáveis: estratégia de design e implementação, ferramentas de desenvolvimento, estrutura das equipes, cronograma, código-fonte legado e assim por diante.]

4. Visão de Casos de Uso

[Esta seção lista casos de uso ou cenários do modelo de casos de uso quando eles representam

funcionalidade central e significativa do sistema final ou, quando têm uma grande cobertura arquitetural — eles experimentam muitos elementos arquiteturais ou quando enfatizam ou ilustram um ponto complexo e específico da arquitetura.]

4.1 Realizações de Casos de Uso

[Esta seção ilustra o funcionamento do software, apresentando algumas realizações (ou cenários) de casos de uso selecionadas e explica como os diversos elementos do modelo de design contribuem para a respectiva funcionalidade.]

5. Visão Lógica

[Esta seção descreve as partes significativas do ponto de vista da arquitetura do modelo de design, como sua divisão em subsistemas e pacotes. Além disso, para cada pacote significativo, ela mostra sua divisão em classes e utilitários de classe. Apresente as classes significativas do ponto de vista da arquitetura e descreva suas responsabilidades, bem como alguns relacionamentos, operações e atributos de grande importância.]

5.1 Visão Geral

[Esta subseção descreve toda a decomposição do modelo de design em termos de camadas e de hierarquia de pacotes.]

5.2 Pacotes de Design Significativos do Ponto de Vista da Arquitetura

[Para cada pacote significativo, inclua uma subseção com o respectivo nome, uma breve descrição e um diagrama com todos os pacotes e classes significativos nele contidos.]

Para cada classe significativa no pacote, inclua o respectivo nome, uma breve descrição e, opcionalmente, uma descrição de algumas das suas principais responsabilidades, operações e atributos.]

6. Visão de Processos

[Esta seção descreve a decomposição do sistema em processos leves (threads simples de controle) e processos pesados (agrupamentos de processos leves). Organize a seção em grupos de processos que se comunicam ou interagem. Descreva os modos principais de comunicação entre processos, como transmissão de mensagens e interrupções.]

7. Visão de Implantação

*[Esta seção descreve uma ou mais configurações da rede física (hardware) na qual o software é implantado e executado. Ela é uma visão do Modelo de Implantação. No mínimo, para cada configuração, ela deve indicar os nós físicos (computadores, CPUs) que executam o software e suas interconexões (barramento, LAN, ponto a ponto, etc.) É incluído também um mapeamento dos processos da **Visão de Processos** nos nós físicos.]*

8. Visão da Implementação

[Esta seção descreve a estrutura geral do modelo de implementação, a divisão do software em camadas e os subsistemas no modelo de implementação e todos os componentes significativos do ponto de vista da arquitetura.]

8.1 Visão Geral

[Esta subseção nomeia e define as diversas camadas e o seu conteúdo, as regras que determinam a inclusão em uma camada específica e as fronteiras entre as camadas. Inclua um diagrama de componentes que mostre os relacionamentos entre as camadas.]

8.2 Camadas

[Para cada camada, inclua uma subseção com o respectivo nome, uma lista dos subsistemas localizados na camada e um diagrama de componentes.]

9. Visão de Dados (opcional)

[Uma descrição da perspectiva de armazenamento de dados persistentes do sistema. Esta seção será opcional se os dados persistentes forem poucos ou inexistentes ou se a conversão entre o Modelo de Design e o Modelo de Dados for trivial.]

10. 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.]

11. 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.]