<Nome do Projeto>

Documento de Arquitetura de Software

Versão <1.0>

[Observação: O template a seguir é fornecido para uso com o Rational Unified Process (RUP). O texto em azul exibido entre colchetes e em itálico (style=InfoBlue) foi incluído para orientar o autor e deve ser excluído antes da publicação do documento. Um parágrafo digitado após esse estilo será automaticamente definido como normal (style=Body Text).]

[Para personalizar campos automáticos no Microsoft Word (que exibem um fundo cinza quando selecionados), escolha File>Properties e substitua os campos Title, Subject e Company pelas informações apropriadas para este documento. Depois de fechar a caixa de diálogo, para atualizar os campos automáticos no documento inteiro, selecione Edit>Select All (ou Ctrl-A) e pressione F9 ou simplesmente clique no campo e pressione F9. Isso deve ser feito separadamente para Cabeçalhos e Rodapés. Alt-F9 alterna entre a exibição de nomes de campos e do conteúdo dos campos. Consulte a ajuda do Word para obter mais informações sobre como trabalhar com campos.]

Histórico da Revisão

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| <dd/mmm/aa> | <x.x> | <detalhes> | <nome> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Índice Analítico

1. Introdução 4

1.1 Finalidade 4

1.2 Escopo 4

1.3 Definições, Acrônimos e Abreviações 4

1.4 Referências 4

1.5 Visão Geral 4

2. Representação Arquitetural 4

3. Metas e Restrições da Arquitetura 4

4. Visão de Casos de Uso 5

4.1 Realizações de Casos de Uso 5

5. Visão Lógica 5

5.1 Visão Geral 5

5.2 Pacotes de Design Significativos do Ponto de Vista da Arquitetura 5

6. Visão de Processos 5

7. Visão de Implantação 5

8. Visão da Implementação 5

8.1 Visão Geral 5

8.2 Camadas 6

9. Visão de Dados (opcional) 6

10. Tamanho e Desempenho 6

11. Qualidade 6

Documento de Arquitetura de Software

# Introdução

[A introdução do **Documento de Arquitetura de Software** fornece uma visão geral do documento inteiro. Ela inclui a finalidade, o escopo, as definições, os acrônimos, as abreviações, as referências e a visão geral do **Documento de Arquitetura de Software**.]

## 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.

[Esta seção define o papel ou finalidade do **Documento de Arquitetura de Software**, na documentação do projeto como um todo, e descreve rapidamente a estrutura do documento. O público-alvo específico do documento é identificado, com uma indicação de como ele espera usar o documento.]

## Escopo

[Uma breve descrição da utilidade do Documento de Arquitetura de Software, do que é afetado por esse documento ou influenciado por ele.]

## Definições, Acrônimos e Abreviações

[Esta subseção contém as definições de todos os termos, acrônimos e abreviações necessários para interpretar corretamente o **Documento de Arquitetura de Software**.  Essas informações podem ser fornecidas fazendo referências ao Glossário do projeto.]

## Referências

[Esta subseção fornece uma lista completa dos documentos mencionados em outra parte do **Documento de Arquitetura de Software**. Identifique cada documento por título, número do relatório (se aplicável), data e organização de publicação. Especifique as fontes a partir das quais as referências podem ser obtidas. Essas informações podem ser fornecidas por um anexo ou outro documento.]

## Visão Geral

[Esta subseção descreve o que o restante do **Documento de Arquitetura de Software** contém e explica como o documento está organizado.]

# 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.]

# 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.]

# 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.]

## 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.]

# 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.]

## Visão Geral

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

## 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.]

# 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.]

# 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.]

# 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.]

## 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. ]

## Camadas

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

# 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.]

# 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.]