

Especificação de Arquitetura de Software

## **Introdução**

Este documento tem como objetivo descrever a arquitetura do Sistema <Nome do Sistema>, seguindo o Modelo 4+1[[1]](#footnote-0) para representar as Visões de Arquitetura de Software.

<Inclua um parágrafo com a justificativa da sua equipe para escolher o modelo 4+1 para representação das visões arquiteturais.>

## Escopo

Este documento auxilia os envolvidos no projeto a compreender os aspectos arquiteturais do sistema que são necessários para desenvolver uma solução que atenda as necessidades do proponente. Além de auxiliar equipes futuras no entendimento do projeto.

### Restrições do sistema

<Incluir uma listagem de restrições e decisões do sistema e suas justificativas. >

<Exemplo: Framework CakePHP: A escolha do CakePHP foi feita, devido a restrições do servidor de produção e visando facilidade de manutenção do sistema, uma vez que o ambiente de produção já possui outros sistemas desenvolvidos com o framework CakePHP.

Padrão arquitetural: O padrão MVC (Model-View-Controller) é uma decisão arquitetural desencadeada pela escolha da utilização do framework CakePHP. >

## Representação arquitetural

### Visão Lógica

Descreve como o sistema é estruturado, em termos de unidades de implementação. Os elementos são pacotes, classes e interfaces. O relacionamento entre os elementos mostra as dependências, as realizações de interface, os relacionamentos parte-todo e assim por diante. Para representar essa visão arquitetural foi utilizado <incluir os nomes dos diagramas utilizados para essa visão.>

<Incluir as imagens geradas dos diagramas.

Nota: Esta visão é obrigatória quando do uso das Visões 4+1 da Arquitetura de Software.

São recomendações de diagramas para essa visão:

* Diagrama de classes de projeto: contém todas as classes implementadas do sistema. Não é obrigatório a representação de métodos, mas é recomendável a representação dos atributos;
* Diagrama de classes de entidade: uma parte do diagrama de classes de projeto que contém apenas as classes persistentes;
* Diagrama de entidade-relacionamento: representa o banco de dados.

Todos os diagramas são opcionais, porém é obrigatório a escolha de pelos uma diagrama para representar essa visão.

Fique atento para otimizar a visualização evitando espaços em branco desnecessários.

Prefira layout vertical para o diagrama se encaixar na página do documento.

Sugestão de ferramentas UML: Astah

Não é recomendado a utilização de ferramentas que não fazem validação de restrições da UML, como draw.io, diagrams.net, etc.

>

### Visão de Implementação

Descreve como os artefatos de desenvolvimento estão organizados no sistema de arquivos. Os elementos são arquivos e diretórios (quaisquer itens de configuração). Isto inclui os artefatos de desenvolvimento e os artefatos de implantação. É a descrição da implementação dos módulos e suas dependências. Representamos essa visão com <incluir os nomes dos diagramas utilizados para essa visão.>

<Incluir as imagens geradas dos diagramas.

Essa visão é opcional. Porém é recomendável a utilização de pelo menos um dos diagramas abaixo:

* Diagrama de pacotes dos subsistemas: contém os pacotes e dependências; e/ou
* Diagrama de componentes: representa os componentes do sistema e deve ser usado quando o sistema é composto de componentes ou microsserviços.>

### Visão de Processos

Trata a divisão do sistema em processos e fluxos. O sistema é dividido em linhas de execução de processos concorrentes (threads). Esta visão de concorrência deverá mostrar como se dá a comunicação e a concorrência destas threads. Esta visão é opcional quando do uso das Visões 4+1.

<Essa visão é opcional é necessária apenas quando há processos complexos ou importantes para o entendimento do sistema.

São recomendações de diagramas para essa visão:

* Modelagem de processos de negócio, usado BPMN;
* Modelagem dos processos utilizando UML (diagrama de atividades ou sequência).>

### Visão de Implantação

Descreve como o sistema é mapeado para o hardware, incluindo as dependências de software e topologia de rede. Representa os dispositivos de hardware e software, os subsistemas e a forma de comunicação entre eles.

<Incluir o diagrama de implantação.

O diagrama de implantação não é necessário quando o sistema em desenvolvimento é monolítico. **Porém é obrigatório nos demais casos**.

Este diagrama de implantação deve ser feito utilizando o diagrama de implantação da UML. Um nó de implantação é algo como infraestrutura física (por exemplo, um servidor ou dispositivo físico), infraestrutura virtualizada (por exemplo, IaaS, PaaS, máquina virtual), infraestrutura em container (por exemplo, um container Docker), um ambiente de execução (por exemplo, servidor de banco de dados, Java EE servidor web / de aplicativos), etc. Os nós de implantação podem ser aninhados.>

### Visão de Casos de Uso / Cenários

Descreve a funcionalidade do sistema, suas interfaces externas, e seus principais usuários. Esta visão é obrigatória no uso das Visões 4+1, porque todos os elementos da arquitetura devem ser derivados dos requisitos.

<O sistema deve ser representado na forma de casos de uso ou estórias de usuários

Casos de uso: incluir um diagrama de casos de uso.

Estórias de usuário: incluir referência para a lista de estórias.>

## 

1. Referencia: KRUCHTEN, Philippe. *Architectural Blueprints – The “4+1” View Model of Software Architecture*. IEEE Software 12 (6) Novembro 1995, pp. 42-50. Disponível em <https://www.cs.ubc.ca/~gregor/teaching/papers/4+1view-architecture.pdf> [↑](#footnote-ref-0)