

Módulo 5 - Atividade Individual

Projeto Arquitetural

Wellington M. Espindula

Abril de 2021

1. Resumo sobre Arquitetura de Software. Com base no podcast que você ouviu, escreva cerca de meia página resumindo o que você aprendeu sobre arquitetura de software, cobrindo ao menos:

- a. O que é arquitetura de software;
- b. Importância da arquitetura de software;
- c. Decisões relacionadas com a arquitetura de software; e
- d. Como representar uma arquitetura de software.

Resposta: Embora existam diversas definições e não uma única consensual, a arquitetura de software rege sobre a estruturação, relacionamento, organização de componentes de um sistema. Segundo Eoin Woods, a Arquitetura de Software é o conjunto de decisões que se tomadas inadequadamente podem causar o cancelamento de um projeto. A partir desta definição, pode-se notar que **a forma como a arquitetura de um sistema é constituída - desde sua concepção até sua evolução - tem o poder de prolongar ou comprometer o ciclo de vida** deste.

Para fins de comparação, podemos imaginar um software *A* sem refatoração constante, repleto de classes deusas altamente acopladas e um software *B* constantemente refatorado para manter módulos com responsabilidades bem definidas e pouco depende entre si. Para modificar o software *A*, adicionando uma nova regra de negócio, o(s) desenvolver(es) do software *A* terão dificuldades maiores para fazê-la do que o(s) de *B*; dado que, em *A*, haverá de procurar-se por todas as implicações da modificação em diversos trechos de códigos, podendo resultar em novos *bugs* e em novos emaranhados de código; ao passo que, em *B*, por tentar manter uma **Arquitetura limpa**, a modificação deverá ser realizada em apenas em trechos específicos de código resultando em pouca (ou nenhuma) implicação fora do componente.

Assim percebe-se que **prezar por manter uma arquitetura limpa no processo de planejamento e desenvolvimento de software resulta em maior manutenibilidade** - e até em maior robustez. Portanto, a fim de obter-se uma Arquitetura Limpa em um processo de desenvolvimento, os desenvolvedores deverão sempre realizar decisões acerca de **(i) Modularização** - A arquitetura deve visar módulos com baixo acoplamento e alta coesão; **(ii) Separação de Interesses** - Cada módulo deverá ter uma única responsabilidade; **(iii) Tecnologias** - *Frameworks*, bibliotecas e afins; **(iv) Requisitos Não-Funcionais** - Requisitos como performance, segurança, etc.

Por fim, a fim de representarmos a arquitetura de um software pode-se utilizar desde uma **notação informal** - tomando um módulo como um quadrado, às vezes organizando-os de forma hierárquica ou às vezes usando setas representando dependências, etc -, mas também pode vir a se utilizar **diagramas de pacotes e diagramas de componentes** da UML bem como **ADLs** (Linguagens de Descrição de Arquitetura).

2. Arquitetura de Sistema de Videoconferência. Faça um diagrama da arquitetura de um sistema controlado por computador para vídeo conferência que permite vídeo, áudio, e dados do computador visíveis para vários participantes ao mesmo tempo. O ponto de partida é entender os concerns (preocupações do sistema) e depois estrutura-los, definindo suas dependências.

Resposta:

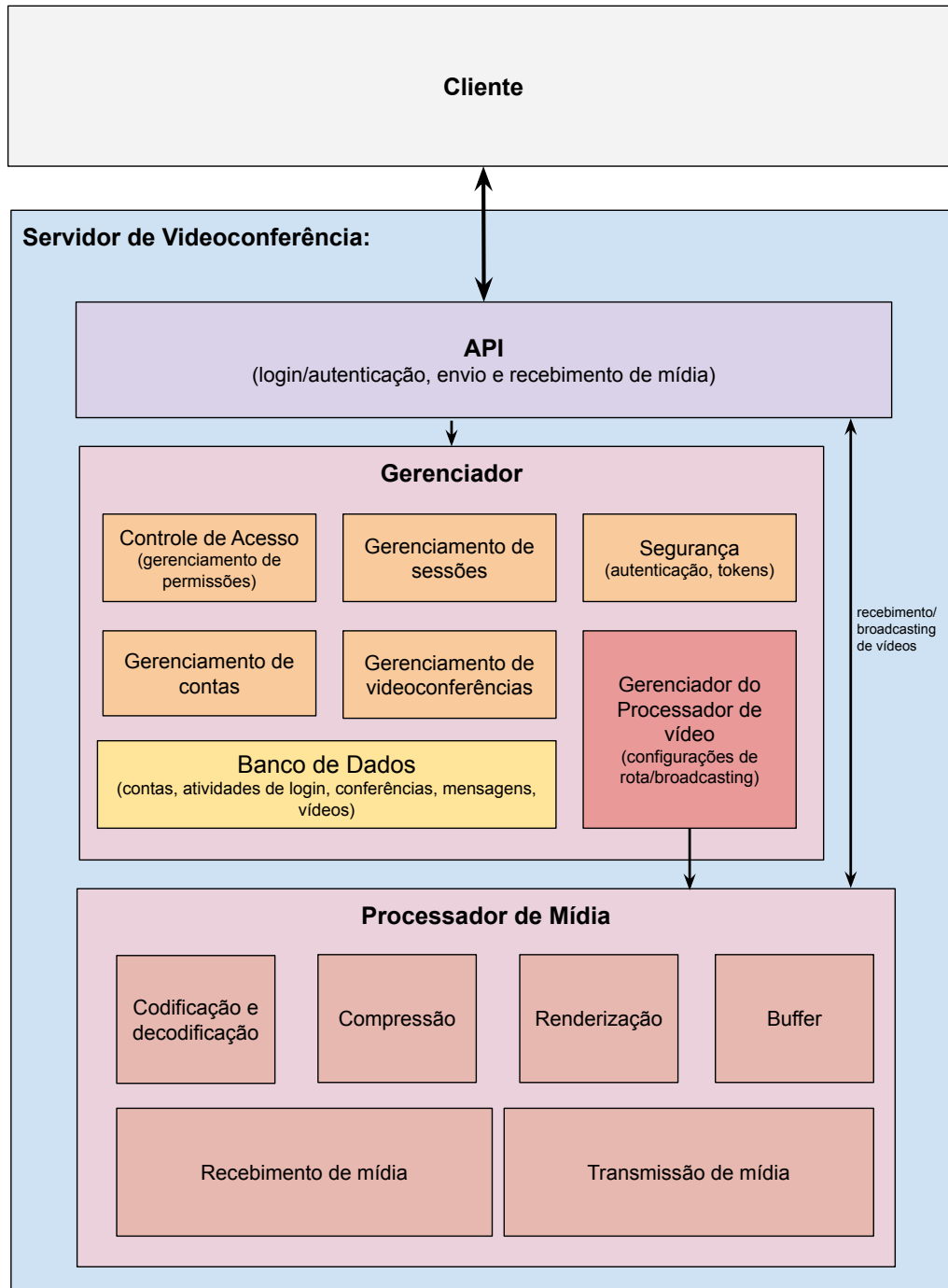


Figura 1: Diagrama de Arquitetura do Sistema de Videoconferência. Fonte: O Autor, 2021.

3. Sistema de Biblioteca. Considere o enunciado do sistema de Biblioteca fornecido no módulo anterior e as atividades realizadas anteriormente e faça o seu projeto arquitetura, representando-o através de um diagrama de pacotes.

Resposta:

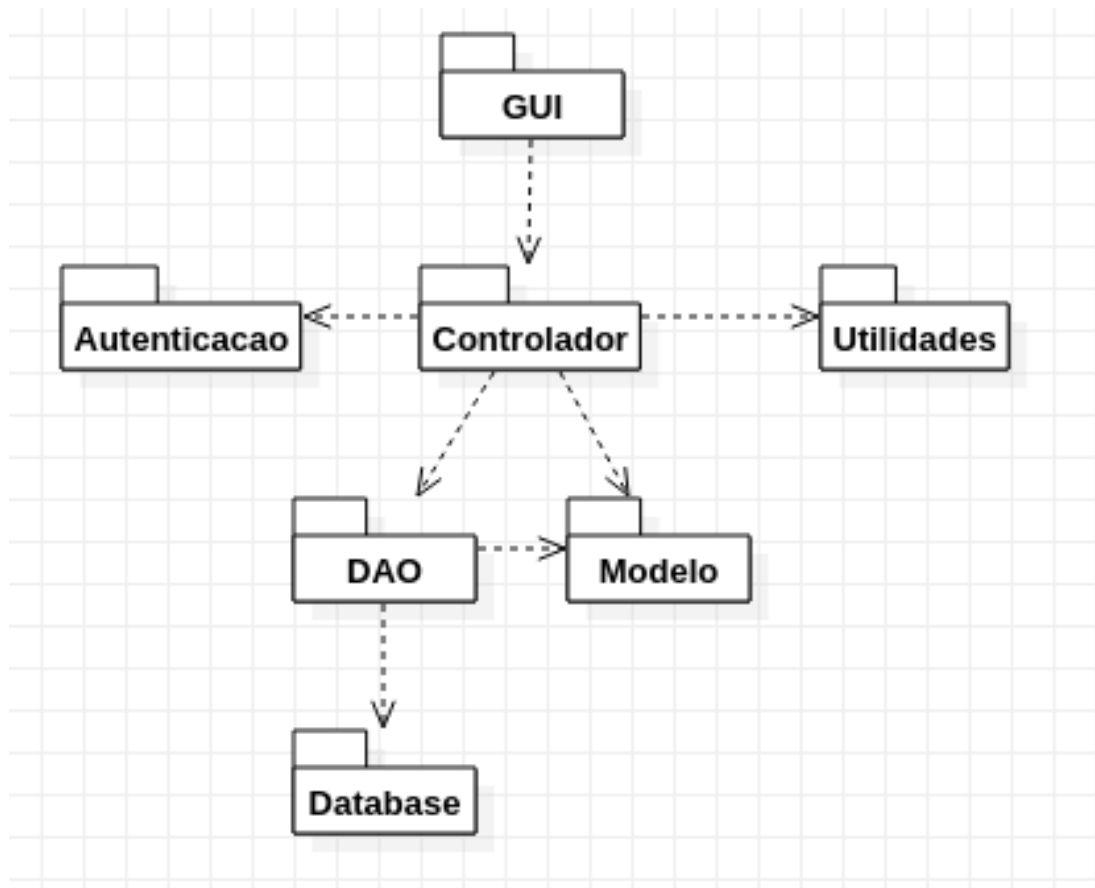


Figura 2: Diagrama de Pacotes do Sistema Biblioteca. Fonte: O Autor, 2021.

4. **Outros Estilos Arquiteturais.** Escolha um dos padrões arquiteturais apresentados no PDF “Modulo 5 -Arquitetura de Software - Estilos Arquiteturais Adicionais”, e escreva um parágrafo resumindo o padrão, destacando sua motivação, problema, solução e consequências. Para fazer isso, busque por detalhes sobre o padrão nos PDFs “Modulo 5 - Leitura - [Chapter] Buschmann - Pattern-oriented software architecture (A System of Pattern)” e “Modulo 5 - Leitura -[Chapter] Sommerville - Software Engineering”, e outros identifiquem.

Resposta: A arquitetura **Pipe and Filter** surgiu do sistema Unix e se baseia em processos de transformações de entrada de dados gerando saídas. Tais processos podem ser combinados através um “Pipe” gerando um fluxo de dados entre processos adequados às necessidades. A motivação de usá-la é em sistemas cujas entradas são processadas em diferentes estágio, isto é, uma arquitetura comum para sistemas de processamento de dados. Uma grande vantagem de usar Pipe and filter é que seu estilo é fácil de entender, facilita o reuso e pode ser utilizado em diversos processos de negócios. Um problema é que sua aplicação restringe-se à sistemas textuais; deste modo, é uma solução pouco escalável para sistemas gráficos. Ademais, para que dois processos se comuniquem, eles não podem discordar em termos de estruturas de dados, o que pode ser resolvido forçando concordância de estruturas de dados entre os processos ou mesmo utilizando um processo intermediário de conversão entre duas estruturas de dados.