

Faculdade de Tecnologia de Sorocaba

Tecnologia em Análise e Desenvolvimento de Sistemas

Programação para WEB

Arquitetura de Software

Alícia Munhóz Franco de Camargo

Sorocaba

Agosto/2025

**SUMÁRIO**

[INTRODUÇÃO 3](#_Toc206068008)

[1. COMPONENTES ESSENCIAIS DA ARQUITETURA 3](#_Toc206068009)

[2. IMPORTÂNCIA DA ARQUITETURA DE SOFTWARE 3](#_Toc206068010)

[3. PRINCIPAIS PADRÕES ARQUITETURAIS 4](#_Toc206068011)

[4. ESCOLHA DA ARQUITETURA 4](#_Toc206068012)

[5. TENDÊNCIAS E EVOLUÇÃO 5](#_Toc206068013)

[6. CONCLUSÃO 5](#_Toc206068014)

[REFERÊNCIAS BIBLIOGRÁFICAS 6](#_Toc206068015)

# INTRODUÇÃO

A arquitetura de software é a estrutura fundamental de um sistema, a qual serve para definir quais serão os componentes, as características e as interações. Ela organiza o sistema, separando o sistema em partes menores, que são chamados de componentes, estabelece suas responsabilidades e como eles se comunicarão, com o objetivo de atender os requisitos funcionais e não funcionais do projeto.

Segundo a ISO/IEC/IEEE 42010:2022, arquitetura é a “estrutura fundamental ou esqueleto do sistema, definindo seus componentes, suas relações e princípios de projeto e evolução”. Ela serve para mapear as decisões estratégicas que podem afetar a qualidade, manutenção, escalabilidade e segurança do software.

Quando o projeto possui uma boa arquitetura, facilita a comunicação entre a equipe, reduz os riscos, orienta o desenvolvimento e ajuda a evitar retrabalho, o que torna o projeto mais eficiente e sustentável.

# COMPONENTES ESSENCIAIS DA ARQUITETURA

Os componentes são partes modulares do sistema, podemos citar como exemplo as classes, os módulos, os serviços e as APIs.

Temos as propriedades externas, que são características visíveis que influenciam o uso, por exemplo o desempenho, segurança e escalabilidade.

Os relacionamentos são a forma como os componentes interagem, definindo o fluxo de dados e a comunicação interna.

# IMPORTÂNCIA DA ARQUITETURA DE SOFTWARE

Seguir um padrão de arquitetura de software é muito importante, devido aos seguintes itens:

* Facilita a comunicação entre todos os envolvidos no projeto;
* Reduz riscos ao identificar e tratar problemas desde o início;
* Organiza o desenvolvimento ao dividir o sistema em partes menores e mais gerenciáveis;
* Permite a reutilização de componentes bem definidos;
* Reduz custos e tempo ao evitar retrabalho e otimizar recursos.

# PRINCIPAIS PADRÕES ARQUITETURAIS

Os principais padrões de arquitetura de software utilizados atualmente são:

* Arquitetura em camadas (Layered Architecture): divide o sistema em níveis como apresentação, lógica de negócios e dados.
* Arquitetura orientada a serviços (SOA): componentes independentes que podem ser reutilizados em diferentes aplicações.
* Microsserviços: pequenos serviços independentes que se comunicam por APIs, favorecendo escalabilidade e autonomia.
* Arquitetura orientada a eventos (EDA): o sistema reage a eventos disparando ações e processamentos.
* Pipes and Filters (Pipeline de dados): dados passam por etapas de processamento sequenciais.
* Arquitetura MVC (Model-View-Controller): separa dados, interface e lógica de controle para melhor organização.

# ESCOLHA DA ARQUITETURA

Para escolher a arquitetura mais indicada para o projeto, deve-se levar em consideração os requisitos funcionais e não funcionais do projeto em si, a capacidade técnica da equipe que irá executar, a escalabilidade e a manutenção esperadas e os custos e prazos estipulados.

# TENDÊNCIAS E EVOLUÇÃO

Temos alguns destaques de tendências mais relevantes, como:

* Arquiteturas cloud-native que utilizam microsserviços, containers e serverless;
* Platform Engineering que padroniza e acelera o trablho das equipes;
* Data Mesh que é um exemplo de uma arquitetura orientada a dados;
* Micro frontends no desenvolvimento de interfaces modulares.

# CONCLUSÃO

Uma boa arquitetura define a estrutura e os componentes, orienta como o sistema evoluirá e será mantido, logo a sua escolha deve levar em conta as necessidades do negócio, os requisitos técnicos e a realidade da equipe.

Se ela for bem planejada, pode garantir a qualidade, escalabilidade, segurança e eficiência, garantindo a base do sucesso de um sistema, facilitando a comunicação e reduzindo os riscos.

# REFERÊNCIAS BIBLIOGRÁFICAS

PESSÔA, C. **Padrões arquiteturais:** *arquitetura de software descomplicada.* Alura. Disponível em: <https://www.alura.com.br/artigos/padroes-arquiteturais-arquitetura-software-descomplicada?srsltid=AfmBOoogoo443GKEVbtQXsxbPFrNEfyFlGooaRSvtQj37dSq2iiTVqW8>. Acessado em: 13 ago. 2025.

ISO/IEC/IEEE 42010:2022 – Systems and software engineering – Architecture description.