

Arquitetura e Projeto de Sistemas

Objetivos da Disciplina

A disciplina de **Arquitetura e Projeto de Sistemas** tem como finalidade preparar o aluno para:

- Compreender a estrutura e funcionamento de sistemas de informação;
- Projetar soluções tecnológicas baseadas em requisitos reais;
- Aplicar modelos arquiteturais adequados a diferentes cenários;
- Desenvolver diagramas e artefatos de projeto como UML, DFDs e ERs;
- Avaliar decisões arquiteturais quanto a desempenho, segurança e escalabilidade.

Exemplos

1 – Sistema de e-commerce

- **Arquitetura:** Microsserviços com APIs REST, autenticação OAuth2, banco NoSQL (MongoDB), frontend em React.
- **Projeto:** Diagrama de caso de uso e de classes com entidades como Produto, Cliente, Pedido. Fluxograma de checkout com etapas de pagamento e confirmação.

2 – Sistema Acadêmico

- **Arquitetura:** Camadas (apresentação, lógica e dados) com MVC.
- **Projeto:** Diagrama de caso de uso, classes, sequência e componentes.

Profissão após conclusão da Disciplina

Ao concluir esta disciplina, o(a) estudante(a) estará apto(a) a exercer funções como:

| Cargo | Atividades Relacionadas |
|--|--|
| Analista de Sistemas | Levantamento de requisitos, modelagem, especificação de sistemas |
| Arquiteto de Software Júnior | Definição de padrões, estruturas e tecnologias de sistema |
| Desenvolvedor de Sistemas | Implementação com base nos modelos arquiteturais e de projeto |
| Analista de Requisitos/Negócios | Comunicação entre áreas técnicas e de negócio, com foco em projeto |

Arquitetura de Sistemas

Conceito

Arquitetura de sistemas é a **estrutura organizacional** de um sistema de informação. Ela define os componentes do sistema, suas funções, interações, interfaces, além das restrições impostas por requisitos funcionais e não funcionais.

Definições

"A arquitetura de um sistema é a organização fundamental de um sistema, incorporando seus componentes, seus relacionamentos e os princípios que orientam seu projeto e evolução." (ISO/IEC/IEEE 42010:2011)

"Arquitetura de sistemas envolve decisões estruturais e tecnológicas que definem como os componentes do sistema são organizados e como se comunicam." (SOMMERVILLE, 2019)

Objetivos

- Proporcionar visão macro do sistema;
- Apoiar decisões sobre tecnologia e infraestrutura;
- Identificar pontos de acoplamento, segurança e escalabilidade;
- Alinhar o desenvolvimento com os requisitos de negócio.

Exemplos

1 – Sistema de votação eletrônica

Componentes: urnas, banco de dados central, aplicativo de apuração, painel de resultados.

Estilo arquitetural: Cliente-servidor e Camadas.

2 – Aplicativo de Delivery

Componentes: app do usuário, app do entregador, backend, banco de dados, gateway de pagamento.

Estilo arquitetural: Microserviços e API REST.

3 – Sistema Acadêmico Universitário

Componentes: portal web, módulos de matrícula, financeiro, biblioteca.
Estilo arquitetural: Arquitetura modular com camadas.

Estilos Arquiteturais

| Estilo | Características | Exemplo |
|---------------------------|--|-------------------------------|
| Monolítico | Única aplicação | Sistema legado |
| Camadas | Separação lógica (UI, lógica, dados) | Sistemas web MVC |
| Cliente-servidor | Comunicação entre cliente e servidor central | Aplicações desktop conectadas |
| Microserviços | Componentes independentes com APIs | Netflix, Uber |
| Orientada a Eventos (EDA) | Comunicação por eventos assíncronos | Sistemas IoT e pagamentos |

Sistema legado é um sistema de informação desenvolvido com tecnologias, arquiteturas ou linguagens antigas, que continua em uso porque ainda atende a processos críticos de negócio, apesar de ser difícil de manter ou integrar com novas tecnologias, (SOMMERVILLE, 2019)

A UI (Interface do Usuário) é a camada responsável pela interação entre o usuário e o sistema. Em uma arquitetura lógica em camadas, a UI não contém regras de negócio nem acesso a dados, apenas apresenta informações e coleta comandos do usuário.

Projeto de Sistemas

Conceito

Projeto de sistemas consiste na especificação detalhada de **como** um sistema de informação será construído, a partir de seus requisitos definidos na fase de análise de sistemas. Envolve a definição da estrutura, das tecnologias e dos componentes que permitirão a construção, integração e operação do sistema.

Definições

“Projeto de sistemas é o processo de definir a arquitetura, os componentes, as interfaces e outras características de um sistema.” (PRESSMAN, 2016)

“Projeto é uma atividade de engenharia que se concentra em uma solução de software que atenda aos requisitos estabelecidos durante a análise.” (SOMMERVILLE, 2019)

Objetivos

- Transformar os requisitos em soluções técnicas concretas;

- Definir a estrutura lógica e física do sistema;
- Permitir que programadores implementem a solução com base nas especificações.

Componentes do projeto de sistemas

| Componente | Descrição |
|----------------|--|
| Projeto lógico | Modelagem da lógica do sistema (como fluxogramas, DFDs, diagramas UML) |
| Projeto físico | Escolha de plataformas, bancos de dados, infraestrutura e tecnologias |
| Interfaces | Especificação das entradas, saídas e comunicação entre os módulos |
| Segurança | Regras de controle de acesso, criptografia, integridade |
| Desempenho | Especificações para garantir tempo de resposta, disponibilidade |

Exemplos

1 – Sistema de Agendamento Médico

- Projeto lógico: diagrama de casos de uso e de classes (paciente, consulta, médico).
- Projeto físico: uso de banco de dados MySQL, API em Node.js, frontend em React.

2 – Aplicativo de Gestão de Estoque

- Projeto lógico: diagrama de casos de uso, classes, sequência e de atividades.
- Projeto físico: arquitetura em três camadas com Java e PostgreSQL.

Conceitos Arquiteturais em Sistemas de Software

Conceitos

A arquitetura de software é o conjunto de estruturas necessárias para raciocinar sobre o sistema, que compreende elementos de software, as relações entre eles e as propriedades de ambos.

Definição

"A arquitetura de software é um conjunto de estruturas de um sistema, composta por elementos de software, as propriedades externamente visíveis desses elementos e os relacionamentos entre eles." (Bass, Clements e Kazman, 2012)

Importância da Arquitetura

- Facilita a manutenção e evolução do sistema;
- Suporta decisões de desempenho, escalabilidade, segurança e reutilização;
- Permite a comunicação entre stakeholders (desenvolvedores, arquitetos, analistas, clientes).

Estilos Arquiteturais

1. Arquitetura em Camadas (Layered)

Divide o sistema em camadas horizontais, como:

- Apresentação (UI)
- Lógica de Negócio
- Acesso a Dados

Vantagens: separação de responsabilidades, fácil manutenção.

Exemplo: sistemas Web MVC (Model-View-Controller).

2. Cliente-Servidor

Componentes divididos em cliente (requisita serviços) e servidor (responde aos serviços).

Exemplo: Aplicações web tradicionais.

Vantagens: centralização do controle, escalabilidade.

3. Microsserviços

O sistema é dividido em pequenos serviços independentes, que se comunicam por APIs.

Exemplo: Netflix, Amazon.

Vantagens: alta escalabilidade, manutenção independente de módulos.

Desvantagem: complexidade de orquestração.

4. Event-Driven Architecture (EDA)

Baseia-se em eventos para comunicação entre componentes.

Exemplo: sistemas de pagamento, sensores IoT.

Vantagens: desacoplamento, reação em tempo real.

5. Arquitetura Monolítica

Aplicação única onde todas as funcionalidades estão agrupadas.

Vantagens: simplicidade inicial.

Desvantagens: difícil manutenção e escalabilidade.

Relação com Requisitos Não Funcionais (RNF)

| Requisito não funcional | Impacto na arquitetura |
|-------------------------|--|
| Escalabilidade | Microserviços, EDA |
| Manutenibilidade | Camadas, Microserviços |
| Desempenho | Cliente-servidor, EDA |
| Segurança | Camadas, autenticação em microserviços |

Exemplo

Sistema de e-commerce:

- Camada de apresentação: React (Frontend)
- Camada de negócio: API em Node.js
- Camada de persistência: MongoDB
- Estilo: Arquitetura em camadas e microserviços (pagamento, estoque, usuários)

Referências

ABNT. NBR ISO/IEC/IEEE 42010:2013 – **Engenharia de Software** – Arquitetura de Sistemas e Software – Documentação de Arquitetura. Rio de Janeiro: ABNT, 2013.

ABNT. NBR ISO/IEC/IEEE 12207:2017 – **Engenharia de Software** – Processos do ciclo de vida de software. Rio de Janeiro: ABNT, 2017.

BASS, Len; CLEMENTS, Paul; KAZMAN, Rick. **Software Architecture in Practice**. 3. ed. Boston: Addison-Wesley, 2012.

ISO. ISO/IEC/IEEE 42010:2011 – **Engenharia de sistemas e software** — Descrição de arquitetura. Genebra: International Organization for Standardization, 2011.

LARMAN, Craig. **Utilizando UML e Padrões**: uma introdução à análise e ao projeto orientado a objetos. 3. ed. Porto Alegre: Bookman, 2007.

NBR 6023:2018. **Informação e documentação** – Referências – Elaboração. Rio de Janeiro: ABNT, 2018.

PRESSMAN, Roger S. **Engenharia de Software**: uma abordagem profissional. 8. ed. São Paulo: McGraw-Hill, 2016.

RICHARDS, Mark; FORD, Neal. **Fundamentals of Software Architecture**: An Engineering Approach. Boston: O'Reilly Media, 2020.

SOMMERVILLE, Ian. **Engenharia de Software**. 10. ed. São Paulo: Pearson, 2019.