

Estruturação de Software

Fichamento da OT 08 – Trilha Back End



por Murilo Ferreira





Como estruturar um software?

Planejamento e Definição Definição de Requisitos Requisitos

Estabelecer os objetivos do software, as funcionalidades desejadas e o público-alvo.

Design de Software

Traduzir os requisitos em um design abrangente. Inclui a arquitetura geral do software, a organização dos componentes, a interface do usuário e os fluxos de dados.

Implementação e Testes Testes

A produção do código-fonte de acordo com o design definido utilizando-se as práticas de programação adequadas

Documentação

Criação da documentação do software, incluindo descrições detalhadas das funcionalidades, instruções de uso e arquitetura do sistema

Componentes de Software

Definição

Componentes de software são unidades modulares e reutilizáveis com interfaces bem bem definidas.

Características

Modularidade, reutilização, interface e encapsulamento são características-chave.

Benefícios

Componentes de software proporcionam maior maior eficiência, produtividade e manutenção manutenção facilitada.

Exemplos

Bibliotecas, frameworks, serviços web e microsserviços são exemplos de componentes de software.

Tipos de Componentes de Software



Componentes de Interface do Usuário (UI)

Responsáveis pela interação com o usuário.



Componentes de Lógica de Negócio

Implementam as regras, regras e funcionalidades do negócio.



Componentes de Acesso a Dados

Responsáveis pela comunicação com bancos de dados.



Componentes de Comunicação

Gerenciam a comunicação entre diferentes partes do sistema.



Tipos de Arquitetura de Software

1

Layers (Camadas)

Este é um dos tipos mais utilizados, onde cada camada possui funções específicas no software, proporcionando maior flexibilidade. Facilita o desenvolvimento e a execução de testes, mas pode comprometer a escalabilidade quando o número de camadas aumenta.

2

Client-Server (cliente-servidor):

O processamento é dividido em módulos e processos separados, combinando dados do cliente e do servidor. Um módulo gerencia a informação, enquanto o outro lida com a obtenção de dados.

3

Model-View-Controller (MVC)

Divide o software em três camadas independentes: modelo (lógica de dados), visão (interface do usuário) e controlador (fluxo da aplicação). Essa separação facilita a manutenção e reutilização do código.



Tipos de Arquitetura de Software

4

Microservices (microserviços):

Utiliza múltiplos serviços e componentes para criar uma estrutura modular, permitindo escalabilidade e independência dos módulos, que podem ser escritos em diferentes linguagens.

5

Pipes-and-Filters (PF):

Baseado em uma arquitetura linear, onde componentes computacionais funcionam como filtros. Estes recebem uma entrada, transformam-na através de um ou mais algoritmos e produzem uma saída para um canal de comunicação.

6

Peer-to-Peer (P2P):

Na arquitetura Peer-to-Peer, todos os pares atuam como clientes e servidores. Cada computador é um provedor de serviços sem depender de um servidor central. O uso de torrents para baixar arquivos é um exemplo dessa arquitetura.



Tipos de Arquitetura de Software

7

Service-Oriented Architecture (SOA):

Organiza funcionalidades em unidades independentes chamadas serviços, que interagem entre si e com outras aplicações através de uma rede. Se baseia em princípios como modularidade, reusabilidade, interoperabilidade, escalabilidade e flexibilidade

8

Publish-Subscribe (Pub/Sub):

Conecta publicadores (publishers) e assinantes (subscribers). Publishers enviam mensagens aos subscribers, que são notificados sempre que novo conteúdo é disponibilizado. Redes sociais como Instagram e plataformas como Spotify usam este padrão arquitetural.

Camadas de Software e Padrões de Arquitetura Web

Camada de Apresentação

Interação com o usuário

Camada de Lógica de Negócios

Implementação das regras de negócio

Camada de Acesso a Dados

Comunicação com bancos de dados

Plano de Projeto de Software

Um plano de projeto de software é um documento que define o escopo, o cronograma, os recursos e as estratégias para o desenvolvimento de um software.

Elementos Essenciais:

Descrição do Projeto

Define os objetivos, funcionalidades e requisitos do software.

Gerenciamento de Escopo

Determina o que será incluído e excluído do projeto.

Cronograma

Estabelece prazos realistas para as etapas do desenvolvimento.

Estimativa de Recursos

Define os recursos humanos, materiais e financeiros necessários.

Plano de Projeto de Software

Estratégia de Desenvolvimento

Define a metodologia e as práticas a serem utilizadas.

Plano de Gestão de Riscos

Identifica e estabelece medidas para lidar com potenciais riscos.

Plano de Qualidade

Define os padrões de qualidade e os métodos de teste.

Plano de Comunicação

Estabelece canais e estratégias para a comunicação entre as partes interessadas.

Os 3 Principais Padrões de Arquitetura Web



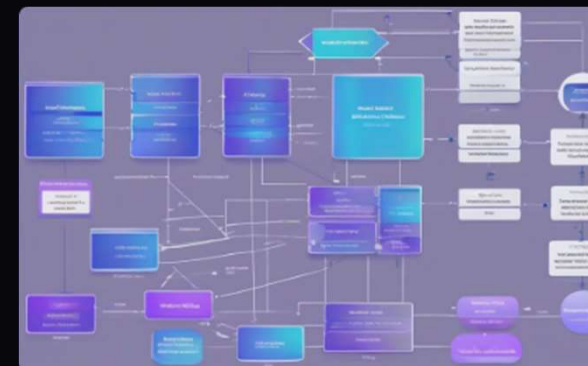
Cliente-Servidor

Divide a aplicação em cliente e servidor, com o cliente exibindo a interface e o servidor processando as requisições.



Camadas de Software

Organiza o software em camadas com responsabilidades específicas, facilitando modularidade, reutilização e manutenção.



Modelo MVC

Separa a lógica de negócio (Model), interface do usuário (View) e controle (Controller) em componentes distintos.

Estes padrões oferecem benefícios como escalabilidade, flexibilidade, segurança, modularidade e testabilidade.

FIM



Referências

PRESSMAN, Roger S. **Engenharia de software: uma abordagem abrangente**. 8. ed. Porto Alegre: Bookman, 2010.

SOMMERVILLE, Ian. **Engenharia de software**. 9. ed. Rio de Janeiro: Editora LTC, 2011.

MYERS, Glenn. **Teste de software: conceitos e técnicas**. 2. ed. Porto Alegre: Bookman, 2012.

ZUSER, Paul. **Documentação de software: um guia prático**. 2. ed. São Paulo: Editora Novatec, 2011.

TRUECHANGE. **Você conhece quais são os padrões e tipos de arquiteturas de software?** 2024. Disponível em: <https://truechange.com.br/blog/tipos-de-arquiteturas-de-software/>. Acesso em: 03 jun. 2024.

ROCK CONTENT. **Rock Content**. Acesso em: 03 jun. 2024.

TECNOBLOG. **Software e Apps**. <https://tecnoblog.net/tema/software-apps/>. Acesso em: 03 jun. 2024.