

Roteiro de Estudos – Sistemas Web I

Bloco 1

Assuntos:

1. Arquitetura da Web.
2. Projeto de Serviços.
3. Web Service e suas aplicações.
4. Protocolo HTTP.

Calendário de aulas:

- 29/09/2022.
- 30/09/2022.
- 03/10/2022.
- 04/10/2022.
- 05/10/2022.
- 06/10/2022.
- 07/10/2022.

Introdução:

Já vimos que, para um site (conjunto de páginas de internet e hipermídias interligadas), a Interface é a parte que interage com o Usuário. O desenvolvimento Front-end se preocupa com o conteúdo (HTML) e a formatação (CSS) da página, além da interação (JavaScript) e a forma com que o usuário experiencia a navegação (UI/UX).

Por trás da Interface Web, há um sistema que faz com que, através da internet, todo usuário que deseja acessar e interagir com um conteúdo possa fazê-lo de forma simples e intuitiva. O desenvolvimento Backend é o que faz a ligação entre a interface e bancos de dados de forma simples e segura. Portanto, para que uma aplicação web funcione corretamente, é necessário um planejamento de acordo com a sua finalidade, seu tamanho (escala) e a forma de trabalho da empresa que a desenvolve. Essa etapa consiste na escolha da Arquitetura do Sistema.

Estilos arquiteturais não resolvem problemas, representando apenas maneiras recorrentes de arquitetura com um propósito geral a ser seguido.

Descrevemos informalmente que um padrão arquitetural detém seu foco em um problema específico, propondo a resolução de suas divergências somente na combinação do emprego de técnicas isoladas.

Alguns dos principais estilos arquiteturais são:

Client-Server

Layered

Service-Oriented Architecture (SOA)
REpresentational State Transfer (REST)
Microservices

Quanto aos padrões arquiteturais, os mais conhecidos são:
Model-View-Controller (MVC)
Three-tier

Padrões de Arquitetura de Sistemas:

Assim como para construir um edifício é necessário um projeto antes da construção, a escolha do tipo de arquitetura de sistemas define a forma com que a aplicação será desenvolvida, implementada e sustentada.

Cliente-Servidor:

O Cliente (um computador ou um programa) possui uma interface de interação com o Usuário; sua função é fazer perguntas ao Servidor e exibir as respostas de forma inteligível.

O Servidor (um computador, uma rede ou diversos serviços) recebe as perguntas do Cliente, processa os pedidos através de sua lógica de funcionamento (programas), acessa bancos de dados e gera respostas para o Cliente.

Exemplo: O browser ou navegador (cliente) exibe uma página (interface) que permite ao usuário fazer solicitações; o cliente envia perguntas para o servidor web através do protocolo HTTP; o servidor web processa a solicitação (através de programas e bancos de dados) e envia uma resposta HTTP com código HTML encapsulado ao cliente, que exibe de forma intuitiva ao usuário.

Camadas:

Separação das responsabilidades.

Hierarquia.

Exemplo de camadas:

Camada de Interação ou Apresentação: cuida da interface (interação com o cliente), responsabilidade da equipe de desenvolvimento frontend.

Camada da Lógica ou de Aplicação: aplica as regras de negócio de forma computacional (programas), responsabilidade da equipe de desenvolvimento backend.

Camada de Dados: persistência (gravação de dados), segurança.

Arquitetura Monolítica:

Todas as camadas encontram-se dentro de uma única aplicação (possuem um núcleo comum).

Fácil de implementar (deploy); estabilidade; robustez.

Difícil de expandir (escalar) e de fazer manutenção, requer uso de computadores potentes.

Exemplo: software de controle de tráfego aéreo.

Arquitetura Distribuída:

Aplicações utilizam recursos externos (serviços) em outro sistema ou rede.

Comunicação entre serviços: protocolos, criptografia.

Quanto maior o número de camadas, mais complexo (diversos protocolos).

Caso especial: microsserviços:

Serviços bem definidos e independentes entre si.

Permite o uso de diversas tecnologias.

Alta escalabilidade.

Difícil para implementar inicialmente (tempo, equipe).

Vocabulário:

Hipertexto	URL	Serviço
Hipermídia	URI	JavaScript
Página	HTTP	Framework
Site	FTP	UI/UX
WWW ou Web	SMTP/POP	Frontend
WEB1.0/2.0/3.0	TCP/IP	Backend
HTML	DNS	Desenvolvimento/Dev
CSS	SSL	Implementar/Deploy
Protocolo	Cliente	Escalabilidade
Endereço	Servidor	Criptografia
Domínio	Usuário	API

Vídeos:

Cliente-Servidor:

Fácil: <https://youtu.be/Xmu-EpxcROM>

Médio: <https://youtu.be/SEC-tbKa1KQ>

Difícil: <https://youtu.be/NMa-f2BtHxs>

1. Arquitetura da Web

Material de Apoio:

“Ferramentas de Desenvolvimento Web”, Daniel dos Santos Brandão.

Resumo:

A. Características e Componentes.

- Estrutura Cliente-Servidor
 - Cliente: interface.
 - Servidor: hospedagem dos sistemas web e armazenamento.
 - Cliente faz pedidos, Servidor envia respostas.
 - Padrão de comunicação = HTTP.
 - Veja os vídeos: <https://youtu.be/Xmu-EpxcROM>
<https://youtu.be/SEC-tbKa1KQ>

B. Padrões de Arquiteturas Web.

- 3 camadas:
 - Apresentação (Interface): front-end, UI/UX.
 - Lógica (Regras de Negócios): back-end, computacional.
 - Dados (Banco de Dados): armazenamento, segurança.
 - Cliente = vê um produto único.
 - Servidor = equipes de desenvolvimento.
 - Veja os vídeos: <https://youtu.be/b7y5emBO2ag>
<https://youtu.be/Em0R3csNMVE>
- Monolítica:
 - Diversas partes com núcleo em comum.
 - Aplicações de grande porte.
 - Servidor monolítico: mesma base de dados e mesmo código-fonte, usados em diversas aplicações.
 - Desvantagem: escalabilidade; sistemas complexos e fortemente acoplados, de manutenção difícil; necessidade de computadores potentes.
- Microserviços:
 - Uso de soluções bem definidas;
 - Serviços e equipes bem definidas e independentes.
 - Vantagem: escalabilidade das aplicações.

Veja os vídeos: Fácil: <https://youtu.be/cqnwYicxUYk>
Médio: https://youtu.be/KBmWI1e_voc
Difícil: <https://youtu.be/RNGpkA2TVCc>

C. Tecnologias e elementos básicos.

- Protocolos (padrões de comunicação entre cliente e servidor):
 - HTTP (HyperText Transfer Protocol): envio de hipermídia.
 - URL (Uniform Resource Location): identificação única de um recurso.
 - FTP (File Transfer Protocol): transferência de arquivos.
 - SMTP (Simple Mail Transfer Protocol): envio de e-mails.
 - POP (Post Office Protocol): recebimento de e-mails.
- Outros: DNS, SSL.