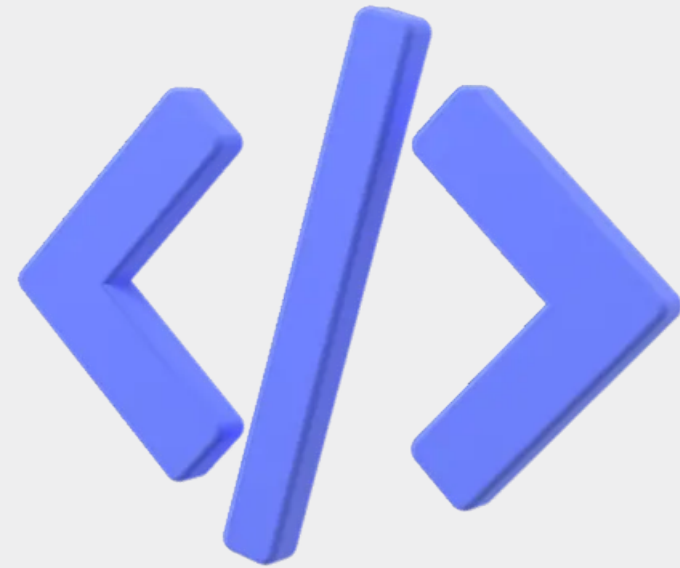


REVISÃO

**CÓDIGOS DE ALTA
PERFORMANCE - WEB**



Protocolos de Internet: HTTP e HTTPS

O HTTP (Hypertext Transfer Protocol) é um protocolo de comunicação utilizado na web para a transferência de dados entre cliente e servidor. Ele segue um modelo de requisição-resposta, onde o cliente (normalmente um navegador) envia uma requisição para um servidor, que retorna a resposta com os dados solicitados.

O HTTPS (Hypertext Transfer Protocol Secure) é uma versão segura do HTTP, que utiliza criptografia SSL/TLS para proteger a troca de informações, garantindo segurança contra ataques como man-in-the-middle.

Principais Métodos HTTP

GET: Solicita a representação de um recurso. Não deve modificar dados no servidor.

POST: Envia dados ao servidor para criação de um novo recurso.

PUT: Atualiza um recurso existente ou cria um novo caso ele não exista.

DELETE: Remove um recurso do servidor.

PATCH: Aplica modificações parciais a um recurso.

TRACE: Retorna a requisição enviada ao servidor para fins de depuração.

HEAD: Semelhante ao GET, mas sem retornar o corpo da resposta, úteis para verificar cabeçalhos.

OPTIONS: Retorna os métodos HTTP suportados por um recurso.

Protocolos SMTP e POP3

SMTP (Simple Mail Transfer Protocol): Utilizado para envio de e-mails, garantindo a transmissão de mensagens entre servidores de e-mail.

POP3 (Post Office Protocol 3): Utilizado para o recebimento de e-mails, permitindo que o cliente baixe as mensagens do servidor para leitura offline.

Arquitetura de Aplicações Web

A arquitetura de aplicações web define a estrutura e organização de um sistema online, considerando aspectos como desempenho, escalabilidade, segurança e manutenção.

Importância da Arquitetura de Aplicação Web

- Facilidade de desenvolvimento: Estruturas bem definidas ajudam no desenvolvimento organizado e colaborativo.
- Escalabilidade: Permite que a aplicação suporte um aumento no número de usuários sem comprometer o desempenho.

Arquitetura de Aplicações Web

- **Segurança:** Proteção contra ataques, falhas e vulnerabilidades.
- **Gerenciamento eficiente:** Facilita a manutenção e a implementação de novas funcionalidades sem comprometer a estrutura existente.
- **Adaptação:** Permite que a aplicação evolua conforme as necessidades dos usuários e as tendências tecnológicas.

Principais Tipos de Arquitetura de Aplicações Web

MVC (Model-View-Controller)

Uma arquitetura que divide a aplicação em três camadas principais:

- Model: Responsável pela lógica de negócio, armazenamento e manipulação dos dados.
- View: Interface visual exibida ao usuário, garantindo uma boa experiência de navegação.
- Controller: Intermediário que recebe requisições, processa as informações e envia a resposta adequada ao usuário.

Principais Tipos de Arquitetura de Aplicações Web

REST (Representational State Transfer)

Arquitetura baseada em recursos acessíveis via URLs e interação com métodos HTTP, seguindo princípios como:

- Baseado em Recursos: Cada endpoint representa um recurso (exemplo: /usuarios, /produtos).
- Uso de Métodos HTTP: GET → Buscar dados, POST → Criar novos dados, PUT → Atualizar dados, DELETE → Remover dados.
- Formato de Dados: Normalmente JSON, mas pode ser XML.
- Stateless: O servidor não mantém estado entre requisições.

UI e UX Designer

UI (User Interface) Designer

Foca no design da interface visual de uma aplicação, garantindo que os elementos sejam intuitivos e agradáveis. Principais aspectos:

- **Tipografia, cores e espaçamento:** Definição visual para tornar a interface harmoniosa e legível.
- **Layout responsivo:** Garante boa usabilidade em diferentes dispositivos.
- **Design de botões e componentes interativos:** Proporciona uma navegação eficiente e acessível.

UI e UX Designer

UX (User Experience) Designer

Foca na experiência do usuário, garantindo que a navegação seja eficiente, intuitiva e satisfatória. Principais aspectos:

- **Pesquisa de usuários e testes de usabilidade:** Avalia necessidades e dificuldades dos usuários para melhorar a experiência.
- **Jornada do usuário (user journey):** Cria fluxos lógicos e naturais para o uso da aplicação.
- **Facilidade de uso e acessibilidade:** Considera públicos diversos, incluindo pessoas com deficiências, garantindo acessibilidade.

Com uma boa combinação entre UI e UX, a aplicação se torna mais atrativa, intuitiva e funcional para os usuários.