

# Introdução ao Desenvolvimento Web

**Guilherme Arthur de Carvalho**

Analista de sistemas

**<https://linktr.ee/decarvalhogui>**

# Objetivo Geral

Introduzir os conceitos fundamentais do desenvolvimento web, compreendendo a estrutura básica da web, a arquitetura cliente-servidor e as principais tecnologias front-end e back-end.

# Pré-requisitos

✓ N/A

# Conteúdo

- ❑ **Introdução ao desenvolvimento web**
- ❑ **APIs – Conceitos fundamentais**

# Introdução ao desenvolvimento Web

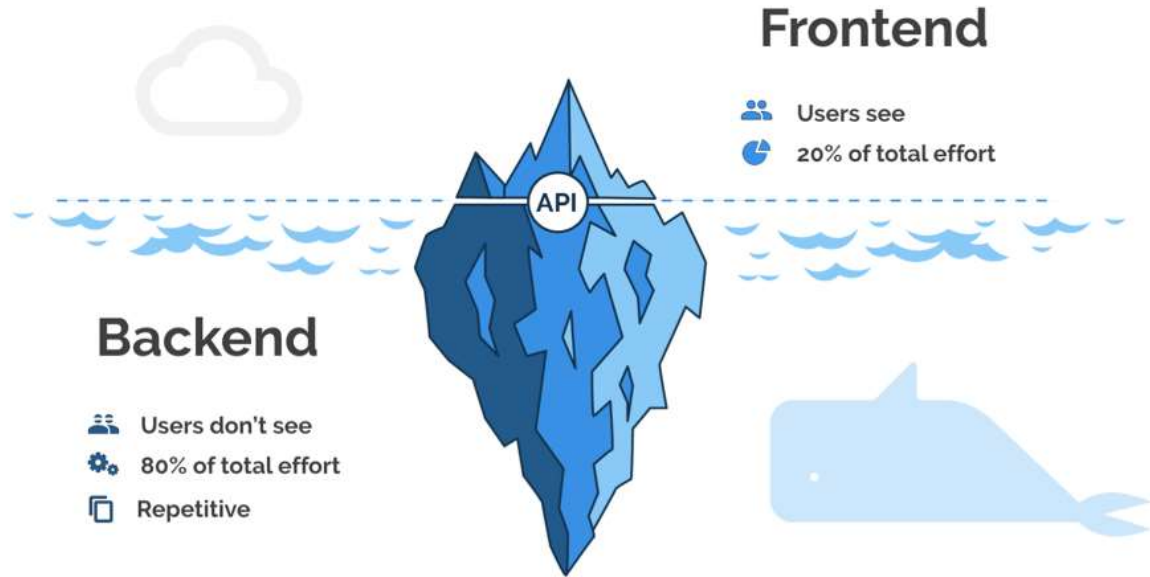
Introdução ao desenvolvimento Web

# O que é desenvolvimento Web?

Desenvolvimento web refere-se ao processo de criação de websites e aplicações para a internet ou uma intranet. Abrange uma variedade de tarefas, incluindo web design, programação web, gestão de bancos de dados e engenharia de servidores.

# Componentes principais

- Frontend: A parte do website que os usuários interagem diretamente. Envolve a criação de interfaces de usuário e experiências, usando tecnologias como HTML, CSS e JavaScript.
- Backend: O 'bastidor' de um website, onde ocorrem o processamento de dados, gerenciamento de banco de dados e controle de servidor. Envolve linguagens como Python, Java, Go, etc.



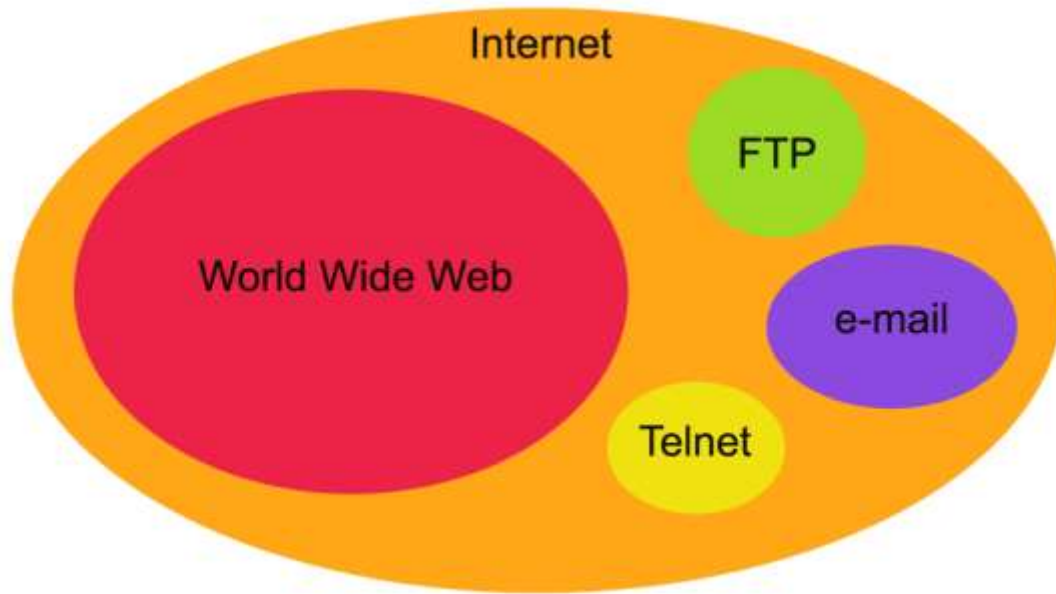


# Como a Web funciona?

Introdução ao desenvolvimento Web

# Internet vs. Web

A internet é uma rede global de computadores interconectados. A Web, ou World Wide Web, é um sistema de informação construído sobre a internet que utiliza o protocolo HTTP para transmitir dados.

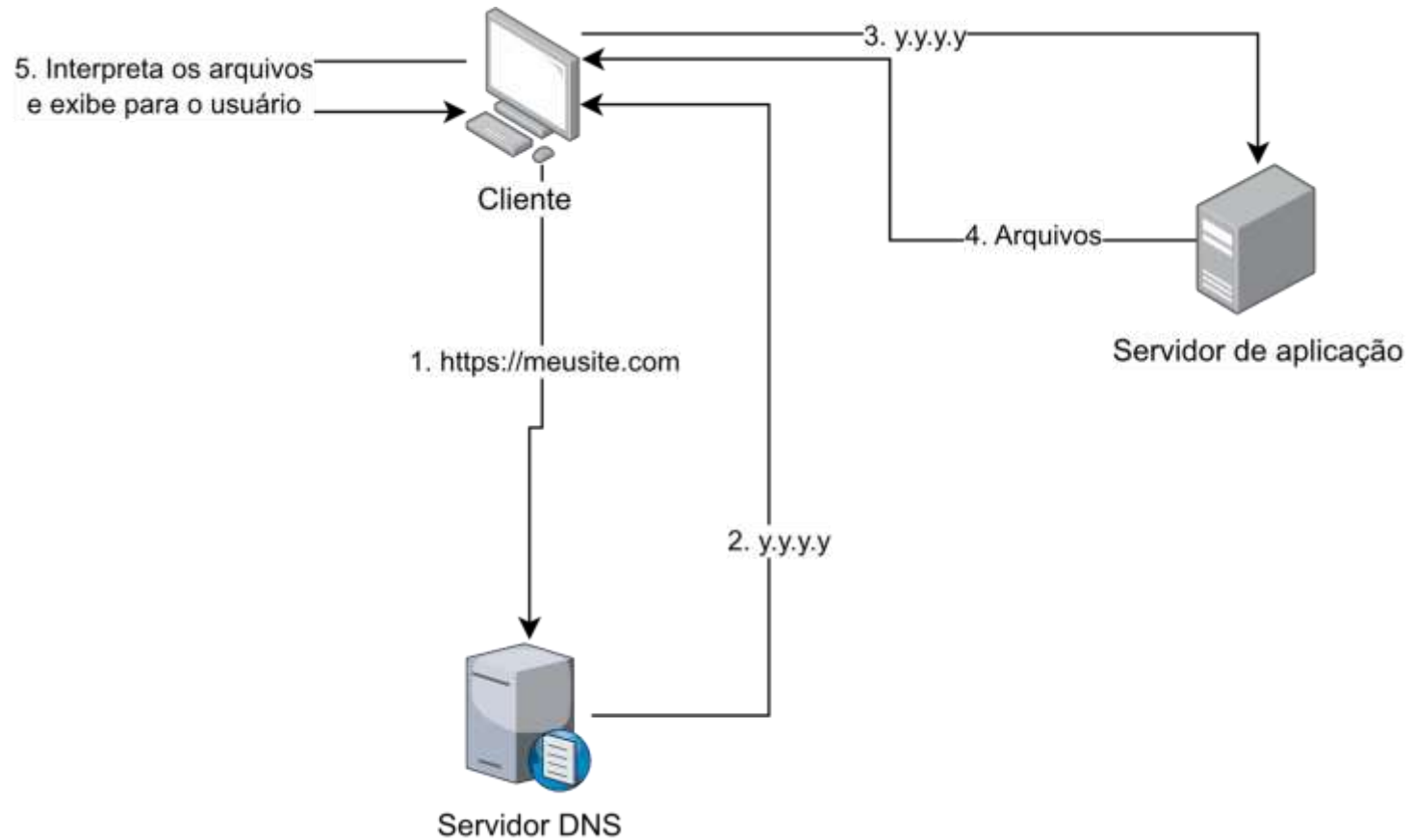


# Protocolo HTTP

HTTP (Hypertext Transfer Protocol) é o protocolo fundamental usado na Web para a transferência de dados. Quando um usuário acessa um site, o navegador envia uma solicitação HTTP para o servidor do site, que responde com os dados do site.

# Funcionamento de um Website

1. **Solicitação do usuário:** Tudo começa com o usuário inserindo um URL no navegador ou clicando em um link.
2. **Resolução de DNS:** O URL é traduzido em um endereço IP através de um sistema chamado DNS (Domain Name System).
3. **Conexão com o servidor:** O navegador utiliza o endereço IP para estabelecer uma conexão com o servidor que hospeda o site.
4. **Resposta do servidor:** O servidor processa a solicitação HTTP e envia de volta os arquivos do site, geralmente em HTML, CSS e JavaScript.
5. **Renderização no navegador:** O navegador interpreta esses arquivos e exibe o site ao usuário.



# Tecnologias envolvidas

Além de HTML, CSS e JavaScript, tecnologias como SSL/TLS para segurança, APIs para interatividade e bancos de dados para armazenamento de dados também desempenham um papel vital no funcionamento da Web.

# Tecnologias front-end e back-end

Introdução ao desenvolvimento Web



# Front-end: A interface do usuário

Front-end refere-se à parte do desenvolvimento web que lida com a interface do usuário. O objetivo é apresentar informações de forma interativa e acessível para o usuário final.

# Tecnologias chave

- HTML (Hypertext Markup Language): Estrutura o conteúdo da web.
- CSS (Cascading Style Sheets): Estiliza e apresenta o conteúdo HTML.
- JavaScript: Torna as páginas web interativas e dinâmicas.

# Back-end: A lógica por trás dos bastidores

Back-end é a parte do site que o usuário não vê. Inclui servidor, aplicação e banco de dados. É responsável por gerenciar e processar dados, garantindo que tudo no front-end funcione corretamente.

# Linguagens e tecnologias

- Linguagens: Python, Ruby, PHP, Java, JavaScript, entre outras.
- Banco de dados: PostgreSQL, MySQL, MongoDB, Oracle, etc.
- Frameworks: Django (Python), Express (JavaScript), Spring Boot (Java).

# Desenvolvimento Full Stack

Desenvolvedores Full Stack são profissionais que têm habilidades tanto em front-end quanto em back-end, sendo capazes de trabalhar em ambas as áreas do desenvolvimento web.

# APIs - Conceitos fundamentais

Introdução ao desenvolvimento Web

# O que é uma API?

API ou Interface de Programação de Aplicações, é um conjunto de regras e definições que permite que diferentes aplicações de software ou componentes se comuniquem entre si. Funciona como um intermediário, permitindo que pedidos sejam feitos e respostas sejam recebidas entre diferentes sistemas de software.



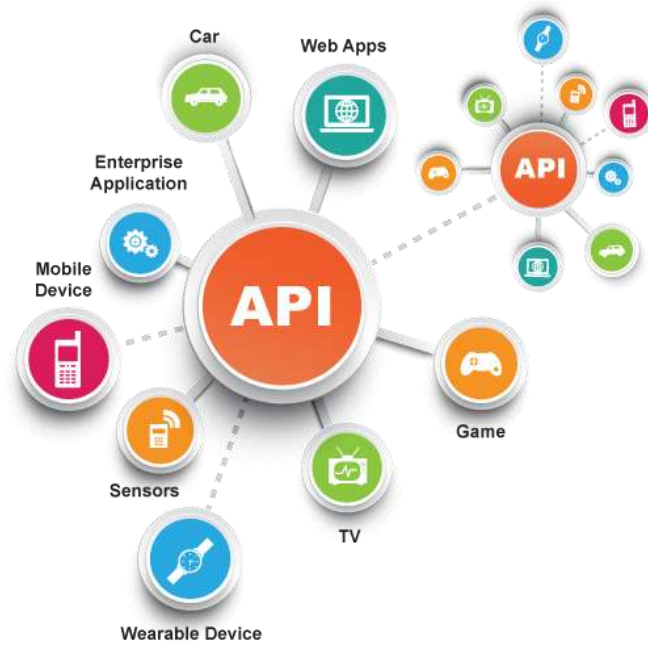


# APIs no contexto da Web

Na web, as APIs são usadas para permitir a interação entre diferentes serviços e aplicações, como enviar dados de um usuário de um aplicativo para um servidor ou solicitar dados de um serviço externo (por exemplo, redes sociais, mapas, previsão do tempo).

# Importância das APIs

As APIs são cruciais para a construção de aplicações modernas e escaláveis. Elas permitem a flexibilidade para integrar e expandir funcionalidades sem reinventar a roda.



# Exemplos práticos

- APIs de pagamento: Facilita transações de comércio eletrônico através de diferentes plataformas de pagamentos.

<https://vindi.github.io/api-docs/dist/#/>

# Tipos de APIs: RESTful, SOAP e GraphQL

Introdução ao desenvolvimento Web

# API RESTful

RESTful refere-se a APIs que seguem os princípios do REST (Representational State Transfer). São baseadas em padrões HTTP e utilizadas para interações web.

# Características de APIs RESTful

- Uso dos métodos HTTP (GET, POST, PUT, DELETE) para operações CRUD.
- Curva de aprendizado menor.
- Fácil de entender e implementar.

# API SOAP

SOAP (Simple Object Access Protocol) é um protocolo que define um padrão para a troca de mensagens baseadas em XML.

[https://www.w3schools.com/xml/xml\\_soap.asp](https://www.w3schools.com/xml/xml_soap.asp)



# Características de APIs SOAP

- Protocolo baseado em XML para troca de informações.
- Independente de linguagem e plataforma de transporte.
- Suporte para operações complexas e segurança avançada.

# API GraphQL

Uma linguagem de consulta para sua API, e um servidor capaz de executar essas consultas, retornando apenas os dados especificados.

<https://studio.apollographql.com/public/SpaceX-pxxbxen/variant/current/explorer>

# Características de APIs

## GraphQL

- Permite que os clientes especifiquem exatamente quais dados querem.
- Eficiente na redução de solicitações e no tamanho dos dados transferidos.
- Flexível e fortemente tipada, facilitando a evolução das APIs.

# Escolhendo o tipo certo de API

A escolha depende das necessidades específicas do projeto, dos recursos disponíveis e da expertise da equipe.

RESTful é popular pela simplicidade, SOAP é preferido para segurança e transações complexas, enquanto GraphQL é ideal para aplicações que requerem dados dinâmicos e personalizados.

# Verbos HTTP: GET, POST, PATCH, PUT e DELETE

Introdução ao desenvolvimento Web

# Verbos HTTP em APIs RESTful

Em APIs RESTful, os verbos HTTP têm papéis específicos que se alinham com as operações CRUD.

Esta abordagem padronizada permite que as APIs sejam intuitivas e previsíveis, facilitando a interação entre diferentes sistemas e aplicações.

# Convenções RESTful

GET para leitura, POST para criação, PUT/PATCH para atualização e DELETE para remoção.

Essas convenções são fundamentais para o design de uma API RESTful bem projetada.

# Dúvidas?

> Fórum/Artigos - <https://web.dio.me/articles>



# Exemplo de código:

Insira sua imagem dentro deste espaço  
(retire o retângulo azul, ele deverá ser utilizado  
somente para referência)

