**Introdução ao 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 banco 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 do servidor. Envolve linguagens como Python, Java e etc.

**Como a Web Funciona:**

* Internet x 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 os 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.

**APIs:**

API ou Interface de Programação de Aplicações, é um conjunto de regras e definições que permite que diferentes aplicações de software 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:**

No contexto da 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.

**Tipos de APIs:**

1. **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.

1. **API SOAP**

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

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.

1. **API GraphQL**

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

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:**

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 (inserção), PUT/PATCH para atualização e DELETE para remoção. Essas convenções são fundamentais para o design de uma API RESTful bem projetada.