USCS
UNIVERSIDADE DE SÃO CAETANO DO SUL

Linguagem de Programação

Prof. Renato Carioca Duarte

USCS
UNIVERSIDADE DE SÃO CAETANO DO SUL

Unidade 08

Aplicações Web

Aplicações Web

- As aplicações web são sistemas que funcionam através de navegadores e utilizam a web como meio de transmissão de dados.
- Elas são construídas usando tecnologias web padrão e podem ser acessadas por qualquer pessoa com um navegador e uma conexão com a internet.

Cliente e Servidor

A arquitetura típica de uma aplicação web envolve pelo menos dois componentes principais: o cliente e o servidor.

Cliente:

- Geralmente refere-se ao navegador web, como Chrome, Firefox ou Safari.
- O cliente é responsável por solicitar recursos (como páginas web) e exibi-los para o usuário.

Servidor:

- É uma máquina ou sistema onde a aplicação web está hospedada.
- Ele escuta e processa as solicitações recebidas dos clientes e envia as respostas apropriadas.

Requisição e Resposta

1. Quando um usuário acessa uma aplicação web, o navegador faz uma requisição ao servidor.
2. Esta requisição pode ser para obter uma página web, enviar dados de um formulário ou buscar informações atualizadas.
3. O servidor processa essa requisição, interage com bancos de dados ou outros serviços se necessário, e envia uma resposta de volta ao navegador, que então renderiza a informação para o usuário.

Frontend

- Frontend refere-se à parte da aplicação que os usuários veem e interagem diretamente.
- Em desenvolvimento web e móvel, o frontend é tudo o que é renderizado nos navegadores ou nas telas dos dispositivos dos usuários.
- Basicamente, é a interface gráfica da aplicação.

Tecnologias Frontend

As tecnologias frontend são usadas para criar a interface do usuário em aplicações web. Isso inclui:

- **HTML** (Linguagem de Marcação de Hipertexto) para estruturar o conteúdo.
- **CSS** (Folhas de Estilo em Cascata) para estilizar e formatar a aparência.
- **JavaScript** para adicionar interatividade e dinamismo.

- Frameworks e bibliotecas como React, Angular, Vue.js, Bootstrap, entre outros, são frequentemente utilizados para facilitar o desenvolvimento frontend e tornar as interfaces mais interativas e responsivas.

Backend

- Backend refere-se à parte da aplicação que opera "por trás dos bastidores", invisível para os usuários finais.
- Enquanto o frontend lida com a interação direta com o usuário e a apresentação dos dados, o backend cuida da lógica, do processamento de dados, e da comunicação com bancos de dados e outras infraestruturas.

Tecnologias Backend

As tecnologias backend são responsáveis por processar as requisições, gerenciar a lógica de negócios, interagir com bancos de dados e fornecer dados ao frontend.

Exemplos de linguagens e frameworks backend incluem Node.js, Python (Django, Flask), Ruby on Rails, Java (Spring), .NET Core, entre outros.

Banco de Dados

- Muitas aplicações web armazenam dados persistentes, como informações de usuários, postagens de blog, produtos, etc.
- Isso é feito em bancos de dados como MySQL, PostgreSQL, MongoDB, etc.

APIs e Serviços Web

- Algumas aplicações web interagem com APIs (Interfaces de Programação de Aplicações) para buscar ou enviar dados.
- Por exemplo, uma aplicação pode usar uma API de terceiros para pagamentos, mapas, integração com redes sociais ou para buscar informações meteorológicas.

Sessão e Autenticação

- Para manter os usuários logados e rastrear suas atividades, as aplicações web frequentemente usam mecanismos de sessão.
- Cookies, tokens e técnicas relacionadas são usados para identificar e autenticar usuários.

Hosting e Implantação

- As aplicações web precisam ser hospedadas em servidores para que os usuários possam acessá-las.
- Existem muitos provedores de hospedagem e plataformas em nuvem, como AWS, Azure, Google Cloud, Heroku, entre outros.

Interação Frontend e Backend

- O frontend muitas vezes se comunica com o backend (ou servidor) para solicitar ou enviar dados.
- Esta comunicação é geralmente feita usando APIs (Application Programming Interfaces), frequentemente no formato REST ou GraphQL.

Comunicação

- A comunicação entre cliente e servidor geralmente ocorre usando o **protocolo HTTP, através de requisições e respostas,** ou sua versão segura, HTTPS.
- Para comunicações em tempo real, tecnologias como WebSockets ou Server-Sent Events podem ser usadas.

Requisições HTTP

- Requisições HTTP, também conhecidas como solicitações HTTP, são ações iniciadas por um cliente (geralmente um navegador da web ou um aplicativo) com o objetivo de obter informações de um servidor da web ou enviar informações para ele.
- Essas requisições são parte fundamental da comunicação na internet e são regidas pelo protocolo HTTP (Hypertext Transfer Protocol).

Requisições HTTP

- Uma requisição HTTP é um pacote de dados que é enviado de um cliente para um servidor.
- A requisição contém informações sobre o que o cliente está pedindo ao servidor, como o 1
    1. método HTTP,
    2. a URL,
    3. os cabeçalhos e o
    4. corpo da mensagem.

Requisições HTTP

- O método HTTP é um verbo que indica o que o cliente está pedindo ao servidor.
- Os métodos HTTP mais comuns são **GET**, **POST**, **PUT** e **DELETE**.

- O URL é o endereço do recurso que o cliente está pedindo ao servidor.
- O URL pode ser um caminho para um arquivo, um recurso de API ou um endereço de um serviço.

Requisições HTTP

- Os cabeçalhos são informações adicionais que podem ser enviadas com a requisição.
- Os cabeçalhos podem ser usados para fornecer informações sobre o cliente, como o tipo de navegador ou o sistema operacional.

- O corpo da mensagem é uma parte opcional da requisição.
- O corpo da mensagem pode ser usado para enviar dados para o servidor no formato texto, JSON ou XML.

Requisições HTTP

- Exemplo de uma requisição HTTP POST:

```
POST /api/users
Content-Type: application/json
{
  "name": "John Doe",
  "email": "johndoe@example.com"
}
```

JSON

- O JSON (JavaScript Object Notation) é um formato leve para troca de dados entre sistemas.
- Ele é fácil de ler e escrever para humanos e fácil de analisar e gerar para máquinas.
- JSON é frequentemente usado para transmitir dados em aplicações web entre um servidor e um cliente, mas também é útil em outras aplicações, incluindo configurações, armazenamento de dados e muito mais.

JSON

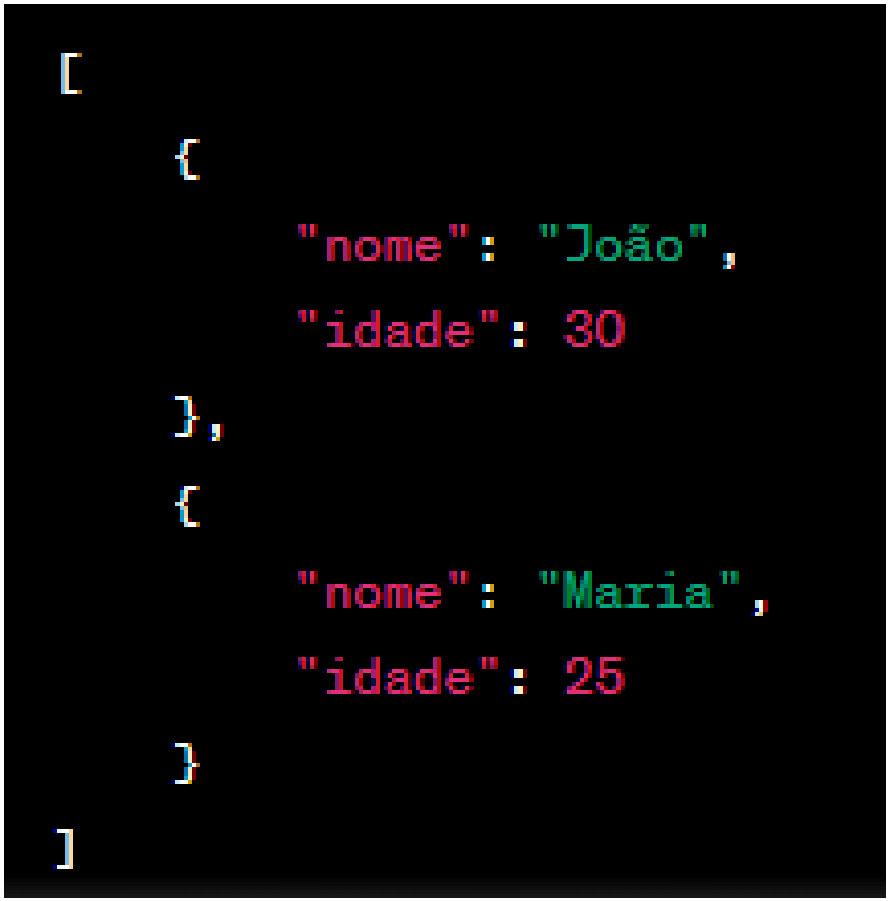
- O JSON é um subconjunto da notação literal de objeto do JavaScript, mas, apesar de seu nome, o JSON é independente da linguagem, com várias linguagens de programação possuindo métodos para analisar e gerar JSON.

Objeto Simples:

```
{
    "nome": "João",
    "idade": 30,
    "cidade": "São Paulo"
}
```

JSON

Array de Objetos:



Formato JSON

- Os nomes dos atributos (às vezes chamados de "chaves" ou "keys") são sempre strings e devem ser escritos entre aspas duplas.
- Os valores podem ser strings, números, objetos JSON, arrays, true, false, ou null.
- As strings em JSON também devem ser escritas entre aspas duplas.

Dúvidas

??????