**AJAX**

Acrônomo de **A**synchronous **J**avaScript **A**nd **X**ML;

**JavaScript** e **XML** como assíncronos;

**Técnica de programação**, utiliza Javascript e que possibilita a **comunicação** assíncrona entre **front-end e back-end** de aplicações Web;

Inicialmente **disponibilizado na versão 4.0** do Internet Explorer;

Potencializou a criação do conceito de web 2.0;

Atualmente o formato mais utilizado para comunicação entre front e back, é o JSON.

**Modelo conceitual de uma aplicação Web**

Cliente, front-end onde browser é executado -> back-end, servidor onde roda a aplicação que geralmente guarda regras de negócio e o banco de dados;

Entre eles, existe o meio físico de comunicação, a internet.

XMLHTTP Request – cliente realiza requisição para o servidor. Gerencia processo de envio da requisição e captura da resposta. Durante esse período, aplicação web não trava.

**Ajax – O que são requisições síncronas e assíncronas**

Aplicações web – inseridas: cliente/servidor;

**Síncrono** – quando efetuamos uma requisição, o browser responsável por ela ficará aguardando até que uma resposta seja enviada pelo servidor;

Cada requisição deve ser finalizada para possibilitar que uma nova requisição seja efetuada.

**Assíncrono** – browser é capaz de disparar diversas requisições, sem ser necessário um sincronismo entre elas;

É possível efetuar diversas requisições simultaneamente e cada uma delas será tratada de forma paralela.

**Requisições síncronas**

Cliente (**Browser**) ---**Request**---------🡪 **Servidor** Apache (htdocs)

Cliente (**Browser**) <----------**Response**-- **Servidor** Apache (htdocs)

Enquanto requisição está sendo feito, fica pendente para o usuário.

**Requisições assíncronas – 1**

**Efetuando requisições HTTP via XMLHttpRequest**

Ao invés de navegar entre páginas distintas, haverá apenas **uma página requisitando** **conteúdo dinâmico**;

Página deve sofrer **request a cada solicitação de dados**.

**function requisitarPagina(url) {**

**let ajax = new XMLHttpRequest();**

**ajax.open("GET", url);**

**//responsável por configurar url requisitada, podemos configurar também, método utilizado nesse processo**

**ajax.send();**

**Requisições assíncronas – 2**

**Estados de requisição**

ReadyState

- **0**: request not initialized (**requisição não iniciada**);

- **1**: server connection established (**conexão estabelecida com servidor**);

- **2**: request received (**requisição recebida**);

- **3**: processing request (**processando requisição**);

- **4**: request finished and response is ready (**requisição finalizada**).

**function requisitarPagina(url) {**

**let ajax = new XMLHttpRequest();**

**//requisição não iniciada, state = 0**

**console.log(ajax.readyState);**

**ajax.open("GET", url);**

**//responsável por configurar url requisitada, podemos configurar também, método utilizado nesse processo**

**//conexão estabelecida com servidor, state = 1**

**console.log(ajax.readyState);**

**//lógica que olhe progresso da requisição**

**ajax.onreadystatechange = () => {**

**console.log(ajax.readyState);**

**}; //executada pelo próprio ajax**

**ajax.send();**

**//na prática, geralmente só utilizamos o estado 4, pois ele indica que a requisição foi finalizada e com isso podemos recuperar a resposta fornecida pelo servidor e tomar alguma ação**

**}**