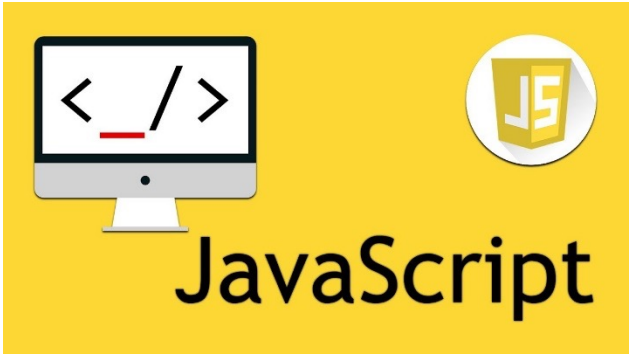
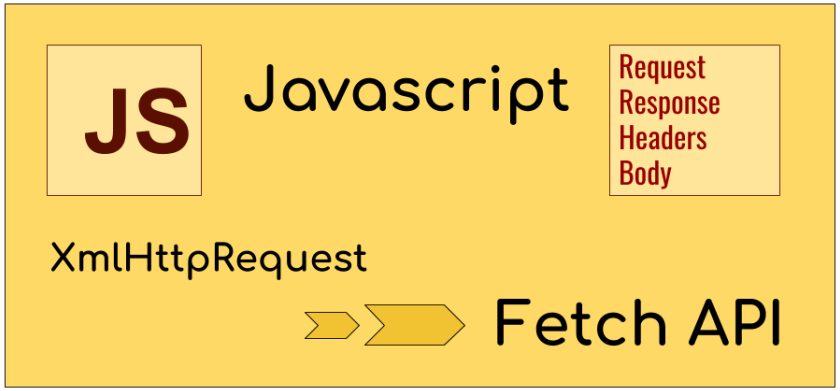
Programação Web



JS – JavaScript

AJAX and Fetch API



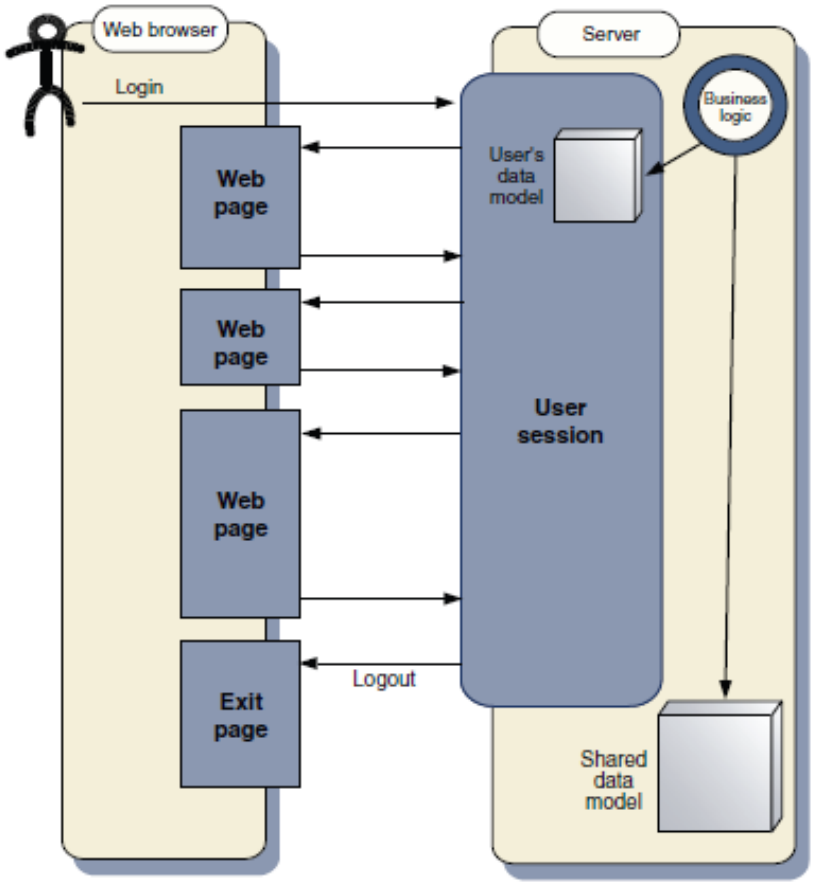
Web Clássica

- Na web clássica, a atualização das páginas web é feita sempre à custa de recarregar toda a página ou carregar novas páginas.

- Este cenário implica muitas vezes o pedido ao servidor de informação que o cliente (browser) já possui e por conseguinte uma carga adicional no servidor e na rede.

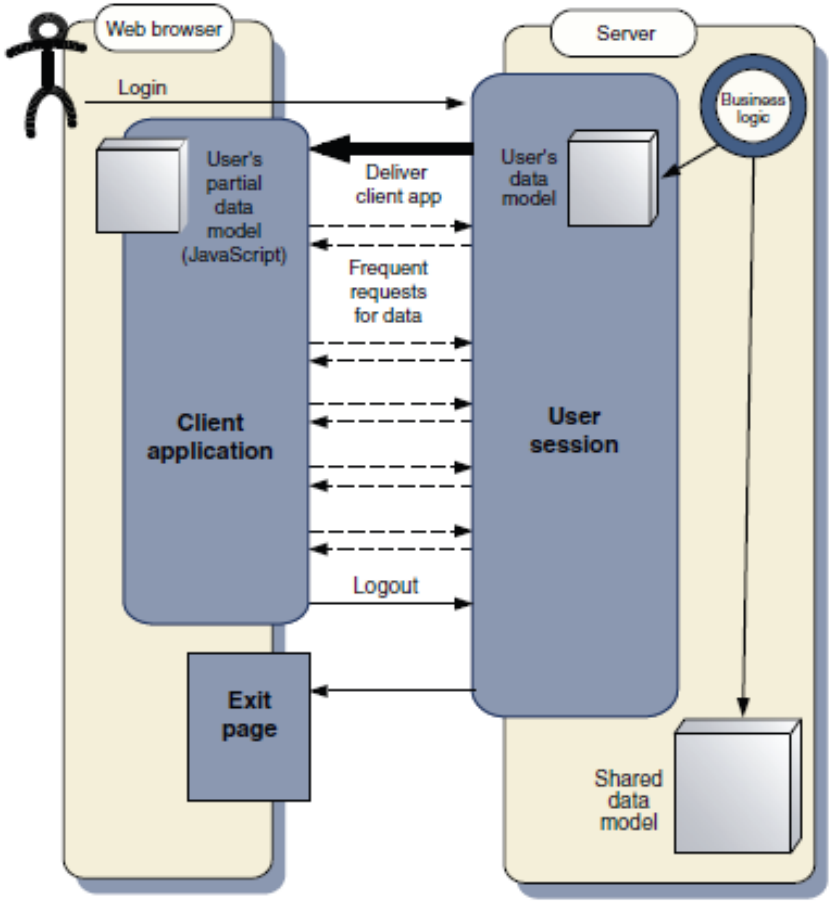
Web Clássica

- Ciclo de Vida



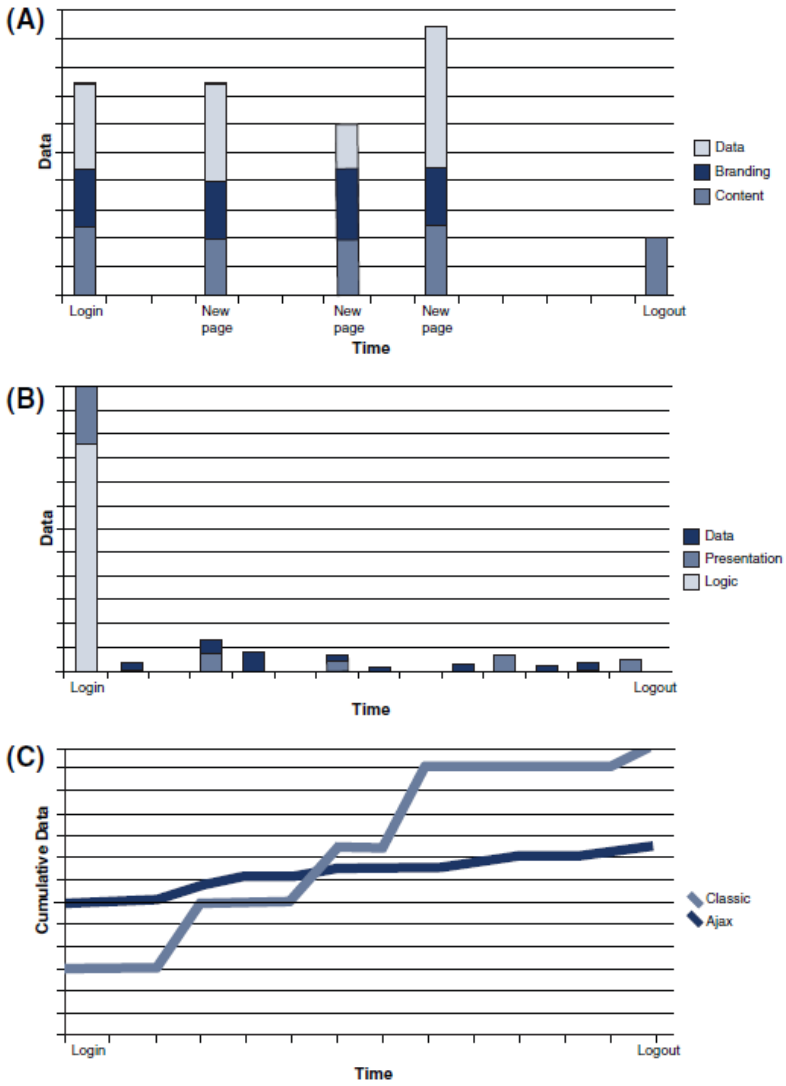
Web com AJAX ou Fetch API

- Ciclo de Vida



Clássica vs AJAX ou Fetch

- Comparativo
  - Dados
  - Representação
  - Processamento



(A) Clássica

(B) AJAX

Web com AJAX e Fetch API

- As tecnologias AJAX e Fetch API permitem a criação de uma web mais dinâmica e mais rápida.
- Permitem transferir dados do servidor para atualizar as páginas web, depois de estas já terem sido carregadas pelo browser.
- Esta atualização pode afetar apenas partes das páginas web sem necessidade de as recarregar de novo do servidor.
- Esta transferência de dados pode ser feita de forma assíncrona, de modo a evitar o bloqueio da interação do utilizador com as páginas.

AJAX e Fetch API

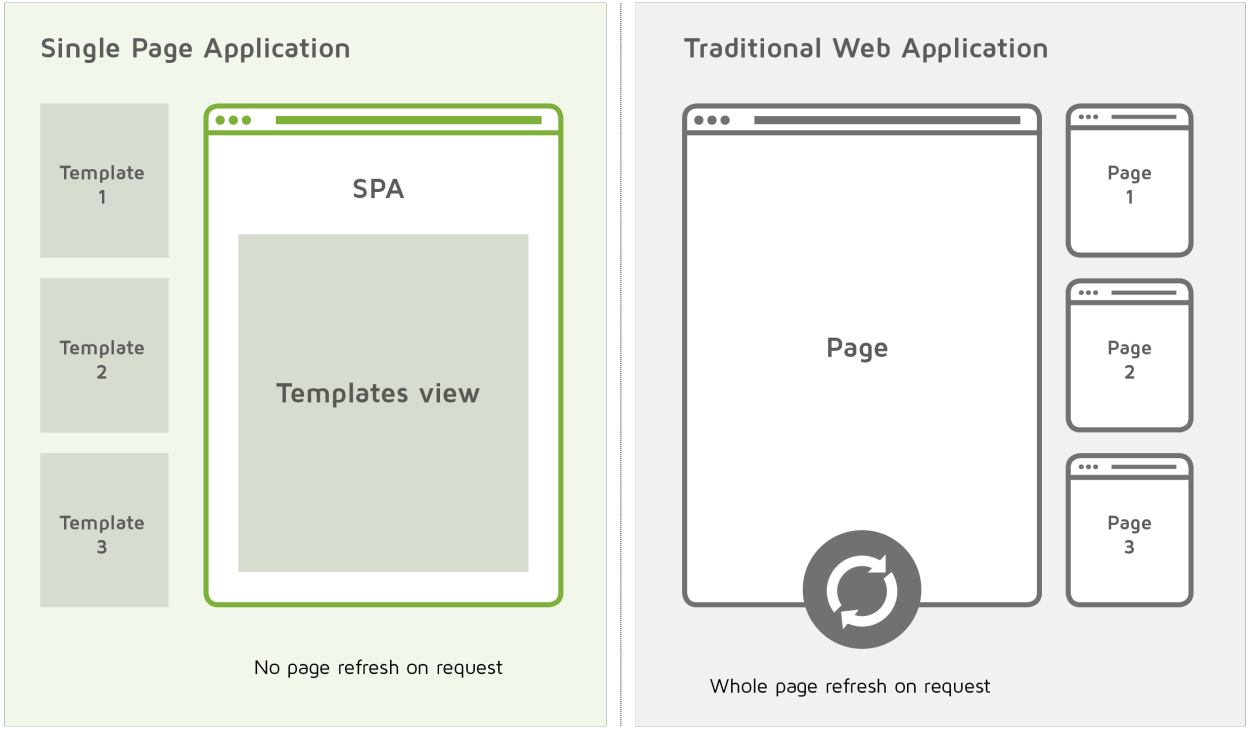
- AJAX – *Asynchronous Javascript And XML*
  - Técnica de desenvolvimento web surgida em 1999 para carregar dados do servidor, no formato XML, de forma assíncrona, permitindo a atualização das páginas web. Atualmente, é mais usado o formato JSON para os dados.
  - Esta técnica tem por base o objeto Javascript XMLHttpRequest que possui todo o seu comportamento padronizado pelo W3C (*World Wide Web Consortium*) e a sua última especificação data de 2016.
- Fetch API
  - Técnica similar ao AJAX, na funcionalidade oferecida, mas com um conjunto mais poderoso e flexível de caraterísticas e maior facilidade de uso.
  - No centro da sua implementação encontram-se as interfaces Request, Response e Headers e o objeto Promise que faz toda a gestão dos acessos assíncronos a recursos.
  - Consiste num *Living Standard*, com data de atualização de 19 de Dezembro de 2021 (https://fetch.spec.whatwg.org/).

SPA – Single Page Application

- Uma SPA consiste num website, ou aplicação web, com uma única página web. À medida que interage com o utilizador e conforme a necessidade, vai reescrevendo dinamicamente partes dessa mesma página.
- Para este efeito, vai carregando do servidor, a cada momento, apenas as partes da página que necessita alterar ou a informação a atualizar e não páginas inteiras, conforme o método tradicional.
- Pelo que, numa SPA, depois do carregamento da principal e única página web, nunca mais se dá um recarregamento total dessa ou doutra página.
- O objetivo é conseguir um website com transições mais rápidas fazendo-o parecer mais com uma aplicação nativa.

SPA – Single Page Application

- A principal técnica a adotar no desenvolvimento de uma SPA é então criar uma página web principal e um conjunto de páginas parciais (*templates*) que serão carregadas e embutidas na página principal, quando necessárias.



Fetch API

- Duas formas de usar a Fetch API
  - com objetos Promise encadeados:

```
fetch(file)
.then(x => x.text())
.then(y => myDisplay(y));
```

  - ou com as palavras async e await que gerem os objetos Promise:

```
async function getText(file) {
  let x = await fetch(file);
  let y = await x.text();
  myDisplay(y);
}
```

Fetch API – Carregar HTML

- Exemplo de uma função para carregar HTML num *container* a partir de um ficheiro.

```
// Load HTML container from Template file
// @dest - id from container where to load HTML
// @file - template filename
async function LoadTemplate(dest, file) {
  let url = "templates/" + file;
  let obj = await fetch(url);
  let txt = await obj.text();
  document.getElementById(dest).innerHTML = txt;
}
```