SPA - PWA

Single Page Application

Progressive Web App

DigitalHouse > Coding School

O que veremos hoje:

- SPA
  - MPA x SPA
  - Como reconhecer uma SPA
  - Ferramentas que auxiliam no desenvolvimento de SPAs
  - Prós e contras

- PWA
  - Conceito
  - manifest.json
  - Service Workers
  - Armazenamento em Browser

Single Page Application

Tudo em uma página só, pode isso Arnaldo?

Conceito

A SPA inicialmente foi pensada para tornar a experiência na web mais próxima da experiência dinâmica vivenciada em aplicações desktop e mobile, reduzindo assim o gap de usabilidade e possibilitando a popularização de aplicações que rodam diretamente no browser.

Conceito

Uma single page application gera apenas um arquivo html conhecido como **app shell** que é baixado no momento que o usuário entra na página. Este arquivo por sua vez, possui toda a parte estática da aplicação e uma grande carga de javascript para gerenciar as mudanças de estado requisitadas pelo usuário (rotas, links e etc).

Conceito

Toda a parte dinâmica da página, como consulta a bancos de dados, fica a cargo do **AJAX** que faz requisições assíncronas ao servidor que por sua vez, retorna apenas dados, geralmente em formato **JSON**.

Este modelo torna mais simples o trabalho do servidor já que este apenas ficará responsável por prover dados “crus” e diminui o tráfego necessário, o que torna a aplicação mais fluida**

MPA x SPA

requisição: https://www.uol.com.br/

resposta: página html, css, javascript, imagens, fontes, etc

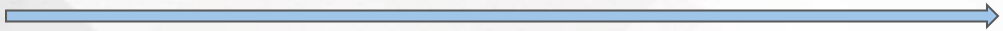
requisição: https://noticias.uol.com.br/cotidiano/

resposta: página html, css, javascript, imagens, fontes, etc

DigitalHouse > Coding School

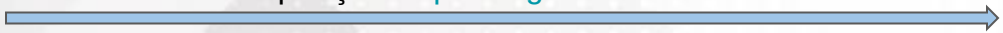
MPA x SPA

requisição: https://angular.io/



resposta: página html, css, javascript, imagens, fontes, etc

requisição: https://angular.io/features



resposta: talvez um JSON

DigitalHouse > Coding School

Vamos demonstrar

DigitalHouse > Coding School

Lazy Loading...

Em alguns momentos pode parecer que a sua aplicação SPA está se comportando como MPA e carregando conteúdo estático no momento do carregamento da página. Isto se deve a dois fatores:

- O conteúdo não é realmente estático
- O sistema marcou aquele arquivo para sofrer **lazy loading**

Lazy Loading...

- É utilizado para aumentar a performance e evitar que arquivos grandes sejam carregados desnecessariamente.

- Não afeta diretamente a reatividade do sistema, uma vez que mesmo esse conteúdo será carregado de forma assíncrona via AJAX.

Principais tecnologias

Angular

React

Vue

Principais tecnologias

Walmart

Paypal

Freelancer

Netflix

Facebook

Twitter

Gitlab

Grammarly

Adobe Portfolio

Prós de uma SPA

- Aplicações altamente reativas
- Total desacoplamento do backend
- Maior facilidade na realização de testes

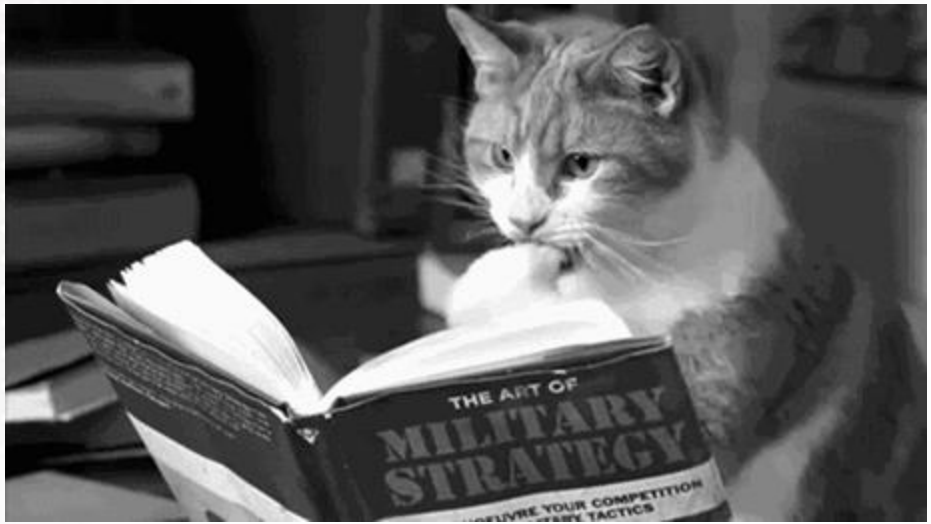
Contras de uma SPA

- Difícil implementação de SEO (Search Engine Optimization)
- Dependência total do JavaScript
- Maior tempo de carregamento
- Baixo suporte para browsers mais antigos

Roleplay

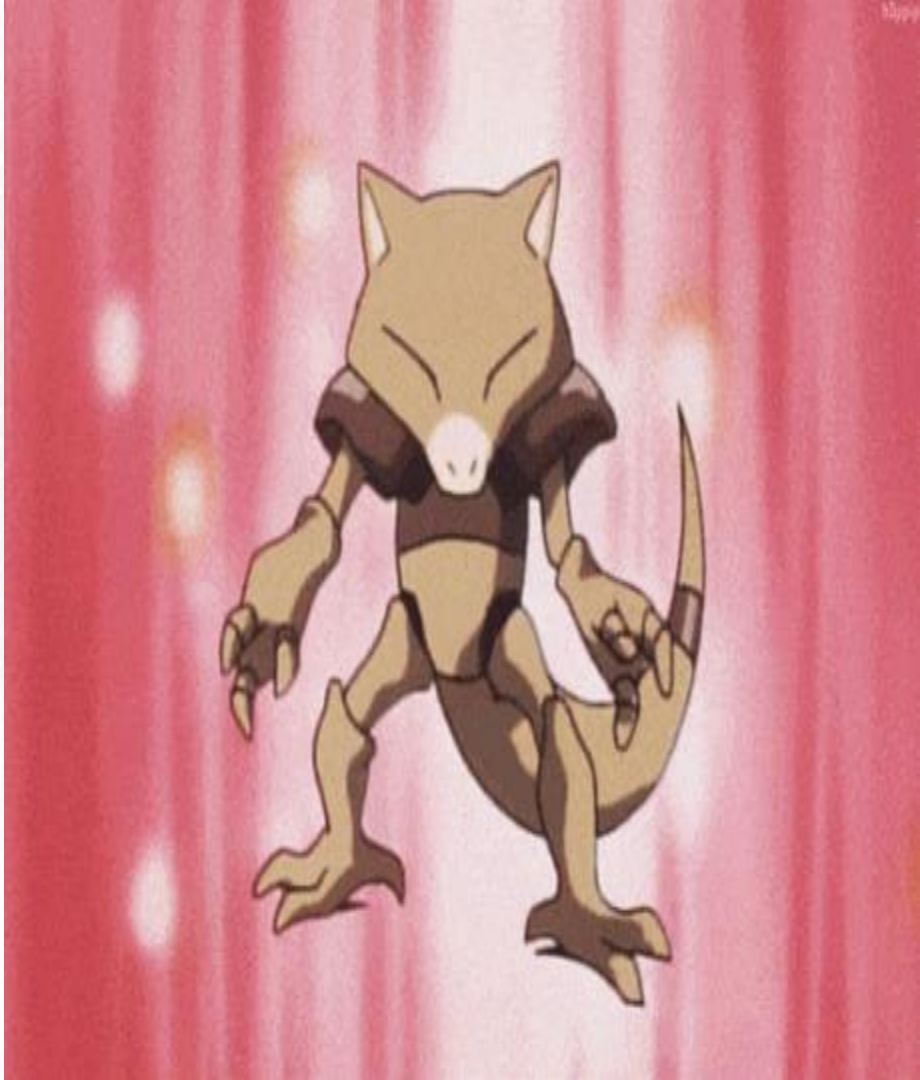
Você e sua equipe estão para lançar um novo sistema no mercado, as especificações, área e público alvo já foram definidos. Com base nisso será necessário fazer a escolha do tipo de frontend que será utilizado.

- E-commerce voltado para produtos tecnológicos
- Sistema clínica médica



DigitalHouse > Coding School

PWA - Um passo além na reatividade



O que é PWA?

PWA é uma sigla para Progressive Web App. O progressive significa que, a depender dos recursos disponíveis no navegador do usuário a sua aplicação apresentará um conjunto de recursos diferentes de forma a atender as necessidades e limitações de quem estiver utilizando.

O Web App reflete bem o que o seu sistema tentará entregar. A experiência de uma aplicação Desktop ou mobile diretamente no browser, e isso quer dizer inclusive, que a sua aplicação poderá ser “instalada” no computador ou smartphone do usuário e deverá ter funcionalidades básicas acessíveis mesmo offline.

PWA exclusivo para SPA?

FALSO! O conceito de PWA pode ser implementado tanto em aplicações MPA quanto em aplicações SPA e isso é o que o torna uma opção tão atraente. Basta que o seu app siga alguns padrões que, independente do tipo de abordagem adotada, ele ainda será uma PWA.

manifest.json

Onde a magia começa. Neste simples arquivo json o desenvolvedor, no caso você, poderá especificar as primeiras configurações sobre o visual da app que será desenvolvida.

```json
{
  "name": "Aplicação de testes - Digital House",
  "short_name": "DH Teste",
  "icons": [
    {
      "src": "/src/images/icons/app-icon-48x48.png",
      "type": "image/png",
      "sizes": "48x48"
    },
    {
      "src": "/src/images/icons/app-icon-96x96.png",
      "type": "image/png",
      "sizes": "96x96"
    },
    {
      "src": "/src/images/icons/app-icon-144x144.png",
      "type": "image/png",
      "sizes": "144x144"
    },
    {
      "src": "/src/images/icons/app-icon-192x192.png",
      "type": "image/png",
      "sizes": "192x192"
    }
  ],
  "start_url": "/index.html",
  "scope": ".",
  "display": "standalone",
  "orientation": "portrait-primary",
  "background_color": "#fff",
  "theme_color": "#187ad6",
  "description": "Uma aplicação muito top que estamos construindo",
  "dir": "ltr",
  "lang": "pt-br"
}
```

Service Worker

Um service worker é um arquivo javascript que funciona como um interceptor de requisições internas do browser rodando em background de forma paralela ao javascript tradicional da sua página.

Ele escuta eventos e dispara ações baseadas no evento que foi disparado, entretanto estes eventos tendem a ser bastante diferentes dos que são usualmente escutados na DOM, uma vez que o service worker não conta com interface gráfica.

Service Worker

- Install: É o evento que dispara a instalação do service worker na aplicação e cria todo o cache estático. Geralmente ocorre na primeira vez que o usuário entra na página.

- Activate: Este evento é disparado após a instalação e é o responsável por fazer a atualização tanto do cache dinâmico quanto do cache estático

Service Worker

- Fetch: Escuta por solicitação de conteúdo e implementa a lógica que deverá servir o conteúdo em cache (se disponível)

- Sync: Faz a sincronização de conteúdo novo no cache

Armazenamento em Browser

- Cache: Para armazenamento de arquivos, documentos e etc.

- IndexedDB: Próprio para armazenamento indexado, pode guardar grande quantidade de dados sendo ideal para espelhar o conteúdo de tabelas, collections, etc.

- Cookies: Servem principalmente para guardar pequenas informações (id do usuário, id da sessão, personalizações feitas pelo usuário, etc).

Hora de fuçar!

Debate!

A PWA é um modelo viável para todos os modelos de negócio?

Debate!

Seria viável aplicar a PWA em um e-commerce?

Dúvidas?

Até o Módulo 2!

DigitalHouse > Coding School