TRIPPUR

DOCUMENTAÇÃO

ANDRE DARGAINS

8 DE ABRIL DE 2018

Funcionalidades

Busca de locais que contenham no mínimo um hotel para busca de hotéis ou um aeroporto para busca de voos.

Filtros e ordenação de resultados.

Redireccionamento para a página do provider.

Subscrição em newsletter.

Multilingue.

Responsivo.

Full Progressive Web App (PWA).

Tecnologias

Markup: HTML;

Styling: CSS com pre-processador SASS;

Scripting: Javascript com framework React;

Bibliotecas de suporte ao scripting: axios, concurrently, insta-scraper, moment, node-sass, query-string, react-infinite-calendar, react-input-range, react-mailchimp-subscribe, react-paginate, react-router-dom, react-toastify;

Tasker: Gulp;

Bundler: Webpack;

Package manager: NPM;

Version Control: github;

Repositório Github: https://github.com/dargains/Trippur.git

Instalação e início

É preciso primeiro fazer um clone ou fork do repositório de github.

```
git clone https://github.com/dargains/Trippur.git
```

Após isto, entrar na pasta *sources* e instalar os pacotes via npm:

```
cd sources && npm install
```

Ao correr a solução, o browser arranca com duas abas: a primeira é a parte da programação e a segunda a parte do markup e styling.

```
npm start
```

Aba localhost:3000 – programação.

Aba localhost:3001 – markup e styling.

Organização lógica da solução

Existem três páginas distintas no website: Homepage, Resultados e Páginas internas. Cada um deles é um container com componentes, encontrado em */sources/src/containers*. Todos os containers são declarados como routes em App.js. São eles:

- **Home:** página inicial com o componente de busca, destinos sugeridos e inscrição na newsletter.
- **Results:** página de resultados tanto de hotéis quanto voos. Onde ocorre todo o mecanismo de tratamento de resultados, filtragem, ordenação e paginação.
- **Pages:** páginas internas. São elas: Sobre, Termos e Condições, Ajuda, Cookies, Privacidade e Parcerias.

Os demais componentes devem ficar em */source/src/components*. Todos importam o módulo *React* e a variável global de língua *lang* como um *prop*, bem como outros módulos e componentes necessários.

Ordem de produção de componentes

A solução foi partida em diversos módulos independentes, de forma que cada módulo pode ser adicionado a qualquer parte do website sem interferir com os demais. Os módulos são criados na pasta */source/modules*, cada um com seu ficheiro de markup e styling. Após o término do desenvolvimento, seu ficheiro de SASS é concatenado com os ficheiros SASS dos demais módulos (bundling) em um único ficheiro de CSS, para otimizar a performance do website via caching. O markup é transformado em um componente React que deverá ser criado na pasta */source/src/components*.

Multilingue

O website suporta multilíngue através da variável de estado *lang*, declarada e mantida unicamente no componente mestre *App.js*. Todo o texto do site deve estar no ficheiro *lang.js*, organizado em formato JSON. Para incluir uma nova língua é preciso criar um novo objeto com as iniciais da nova língua em *lang.js* e a nova opção no componente *languageMenu*.

Progressive Web App

A solução está configurada para ser full PWA, ou seja, atender todos os requisitos para ser considerada assim. São eles os requisitos:

- Site é veiculado por HTTPS;
- As páginas são responsivas em tablets e dispositivos móveis;
- Todos os URLs são carregados enquanto estão off-line;
- Metadados fornecidos para adicionar a webapp à tela inicial em dispositivos móveis;
- Primeiro arranque rápido mesmo em 3G;
- O site funciona em todos os navegadores mais populares;
- As transições de página não parecem ser bloqueadas na rede;
- Cada página tem um URL;