

## NÍVEL AVANÇADO



### PROJETO 16

(CONTEÚDO DISPONÍVEL) {  
**TRIP;**  
**ADVISOR;**  
(end);  
})();

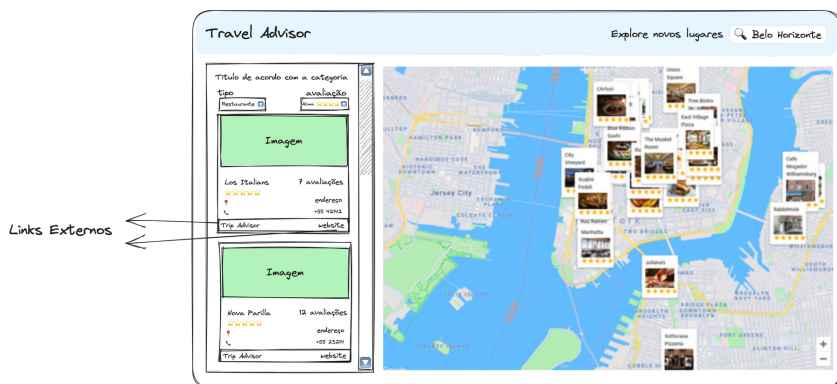
#PORTFÓLIOBOOSTPROGRAM

### CONHECIMENTOS REQUIRIDOS:



**FRONT-END**

# WIREFRAME



## Interação com API's

travel-advisor/Rapid API => Fetch dos lugares (restaurantes, pontos turísticos e hotéis)

open-weather-map/Rapid API => Fetch do Clima

Google Maps API => Mapa e navegação no MAPA

# TRIP ADVISOR

Vamos criar um app que irá ser seu guia para viagens!

## TECH STACK

- NextJs
- Typescript
- TailwindCSS
- React



## LIBRARIES

- [@REACT-GOOGLE-MAPS/API](#)



## BRIEFING

Vamos desenvolver ele utilizando boa parte das tecnologias que já citamos aqui muitas vezes. App excelente para treinamento de **React Hooks** e **manipulação de API**. Após o desenvolvimento desse app você se tornará um mestre no consumo de **API's** para o **front-end**.



## NÍVEL 1

Implementar a solução utilizando **NextJS**.

## NÍVEL EXTRA

Integrar com a **API**:

**API**

Para disponibilizar o **tempo** para cada cidade que for buscada.

## REQUISITOS DETALHADOS



Vamos desenvolver utilizando o **create-next-app**:

**CREATE-NEXT-APP**

Com **typescript**.



Instale a **library react-google-maps/api**:

**REACT-GOOGLE-MAPS/API**



- ➔ Leia **detalhadamente** a documentação pois iremos precisar de algumas **props** dessa **lib** como **onChange** ao mudar a **posição** do mapa.
- ➔ Você deverá criar uma **API KEY** dentro do **Google Maps Platform** e **habilitar em APIS** "Maps Javascript API".
- ➔ Recomendo **tailwindCSS** para os estilos.
- ➔ No **header** teremos uma **barra de navegação** que irá interagir com a **API** a cada evento **onKeyDown** dentro do nosso **input** de destinos, use um componente de **autocomplete** no input.
- ➔ Vamos **dividir** o app em **2 grandes components**, na coluna a esquerda será renderizada a lista de **ocorrências** e do lado direito o **mapa** (veja wireframe para maiores informações).
- ➔ Crie os components o mais **modularizado** possível, utilize o conceito **DRY(Don't Repeat Yourself)** para criar componentes **atômicos e reutilizáveis**.
- ➔ Separe a **logica** de **estados** no que diz respeito a **useEffects** em uma página separada somente para seus **hooks**. Dois hooks que serão necessários são:
- ➔ Um hook que será carregado **somente uma vez (sem dependency array)** com a sua **geolocalização** (**navigator.geolocation.getCurrentPosition**) mais detalhes em:

[DEVELOPER.MOZILLA.ORG](https://developer.mozilla.org)



- ➡ Um **hook** que será para receber os **resultados do fetch** de sua coordenada de busca.
- ➡ Faça **cadastro** no:

## RAPIDAPI



- ➡ Que é um **agregador de API's**. Esse serviço te poupará tempo em buscar onde **consumir os dados** que precisamos.
- ➡ A **API** que iremos **integrar** para nos **gerar as listas de restaurantes, atrações e hotéis** está disponível em:

## TRAVEL-ADVISOR



- ➡ Uma **opção secundária** seria a:

## FOODKIT.IO

