

NÍVEL BÁSICO



PROJETO 08

(CONTEÚDO DISPONÍVEL) {
MAPA;
ESTATÍSTICO;
DASHBOARD;
(end);
})();

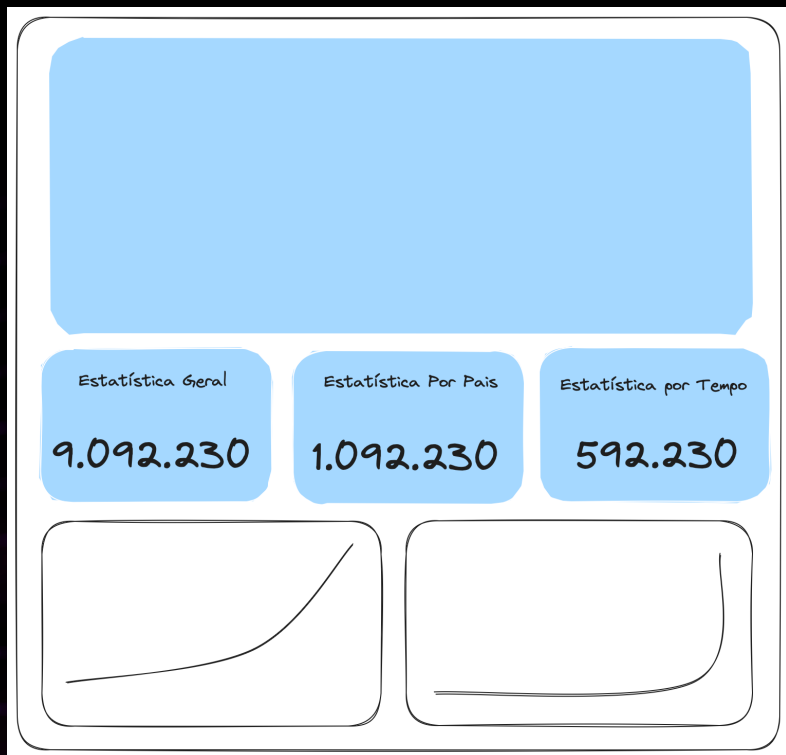
#PORTFÓLIOBOOSTPROGRAM

CONHECIMENTOS REQUIRIDOS:



FULL-STACK

WIREFRAME



NEWSLETTER

Criar um **mapa dashboard** que mostra as **estatísticas e informação geográfica** sobre o **COVID-19**.

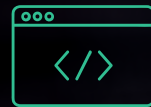
TECH STACK

- ➔ React
- ➔ NodeJS
- ➔ NextJS (opcional)



LIBRARIES

- ➔ React Leaflet
- ➔ Mapbox GL
- ➔ COVID-19 Data APIs
- ➔ React-Router



BRIEFING

A idéia é ter um mapa com as estatísticas de cada país é uma maneira útil de determinar coisas como quais países foram mais afetados.

NÍVEL 1

A melhor maneira desse app ter uma boa experiência com o usuário é ele exibir um **mapa interativo** no qual voce consegue visualizar os dados ao clicar em um certo país no mapa.



NÍVEL 2

Enquanto ter um mapa interativo parece uma excelente idéia, para subirmos de nível, vamos **comparar as estatísticas** de cada país com os valores globais que serão recebidos da **API**.

NÍVEL 3

Por último para seu app estar completo, porque não usar os dados históricos para **prover contexto** e exibir um **gráfico de crescimento**?



REQUISITOS DETALHADOS

Back-end:

- ➔ Crie um novo aplicativo **Node.js**.
- ➔ Você pode usar **express**, ou até **nextJS** (vale lembrar que a solução fica mais complexa com next)
- ➔ Instale as **APIs** de dados COVID-19.

Possibilidades:

API NINJAS



ou

APIFY



- ➔ Obtenha **dados** sobre países e a situação global.
- ➔ **Armazene** os dados em um **banco de dados**.

Opcional para o nível 1

- ➔ Crie uma **API REST** para expor os dados ao **front-end**.

Front-end:

- ➡ Crie um novo aplicativo **React**.
- ➡ Use qualquer versão acima da **16.8.0** (primeira versão do hooks)
- ➡ Instale **React Leaflet** ou **Mapbox GL**.

REACT LEAFLET**REACT MAPBOX**

- ➡ Busque dados da **API REST**.
- ➡ Crie um **mapa** e adicione **marcadores** ao mapa.
- ➡ Adicione **estatísticas aos marcadores**.
- ➡ Crie um **painel de estatísticas**.
- ➡ Adicione **gráficos** ao mapa.
- ➡ **Vale lembrar que:**
- ➡ O **back-end** deve ser responsável por **buscar dados, armazenar dados e fornecer uma API REST**.

- ➔ O **front-end** deve ser responsável por **exibir dados, interagir com o usuário e fornecer uma interface de usuário**.
- ➔ As duas partes devem ser **dissociadas** para que possam ser **desenvolvidas de forma independente**.

O que isso quer dizer?

- ➔ Separe a lógica em **folders** que tenham nomes bem **semânticos e distintos** de acordo com a sua responsabilidade.
Exemplo: Web para Front e API para back.

