**REACTGRAM**

***FRONT-END***

**npm i react-icons –** pegar ícones de ações necessárias;

**react-router-dom –** para rotear as páginas;

**@reduxjs/toolkit –** serve para gerenciar os estados como a contextapi.

**Implementando o Redux**

Instalação do redux de fato:

***npm i react-redux***

**Implementação: *index.js***

- Import Provider

// Redux

import { Provider } from 'react-redux'//similar a contextAPI

- entrega o que está no redux (store manage: guarda dados e compartilha entre a aplicação)

- Onde dados são guardados

//arquivo único que importa os contexts mais variados, só é trabalho um arquivo desse

import { store } from './store'

**arquivo store.js**

***src > store.js***

import { configureStore } from '@reduxjs/toolkit'//importação necessária

export const store = configureStore({//onde contexts estão sendo salvos(deve ser feita a execução da função importada acima 'configureStore')

    reducer:{//onde fica os contexts

    }

})

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(

  <React.StrictMode>

    <Provider store={store}>{/\*o que será entregue aos componentes, tudo oque está no redux\*/}

      <App />

    </Provider>

  </React.StrictMode>

)

A partir desse envolvimento tudo o que estiver no redux poderá ser trabalhado no app

**- Criação de mais duas pastas**

**services:** lida com o backend, onde requisições com o Fetch serão executadas; É como uma “ponte” a API.

**slices:** ações a serem executadas baseadas nas requisições que tem em serviço. Ex.:

Ação de registro que dispara os seus diversos estados (loading, success..); podem ser feitas manipulações entre esses estados.