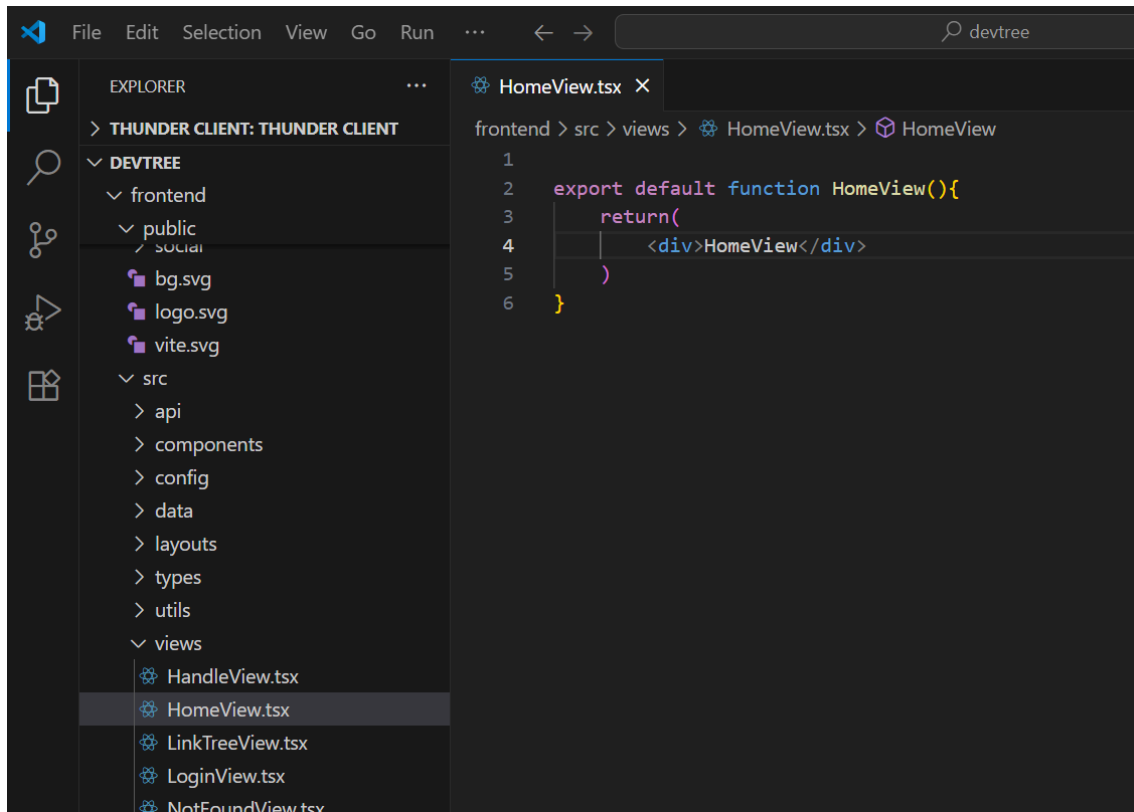


Finalizando el proyecto creando la página principal

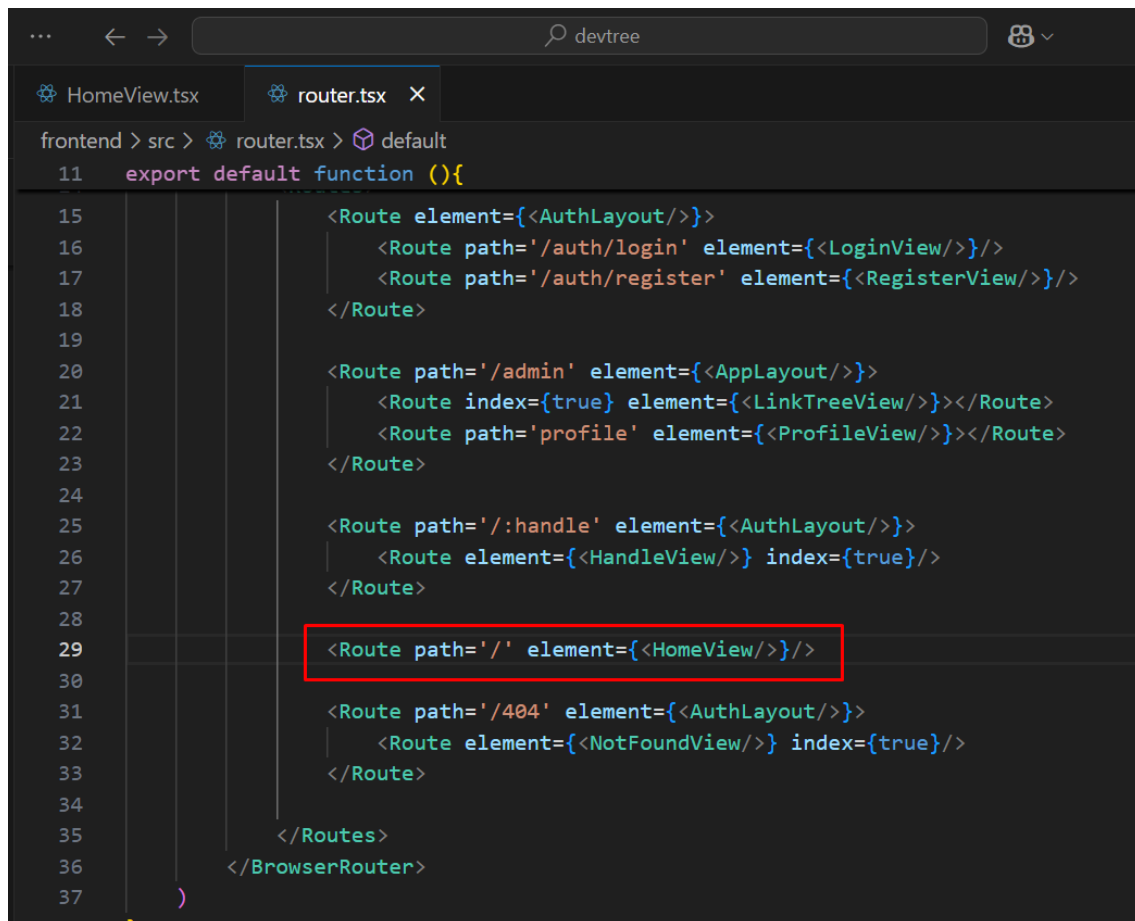
Vamos a crear una página principal para nuestro proyecto que va a explicar en qué consiste la aplicación.

Creemos HomeView



The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays the project structure. The 'src' directory is expanded, showing subdirectories like 'api', 'components', 'config', 'data', 'layouts', 'types', 'utils', and 'views'. The 'views' directory is further expanded, listing files: 'HandleView.tsx', 'HomeView.tsx' (which is selected and highlighted), 'LinkTreeView.tsx', 'LoginView.tsx', and 'NotFoundView.tsx'. The main editor area on the right shows the content of 'HomeView.tsx'. The breadcrumb navigation at the top of the editor indicates the path: 'frontend > src > views > HomeView.tsx > HomeView'. The code in the editor is as follows:

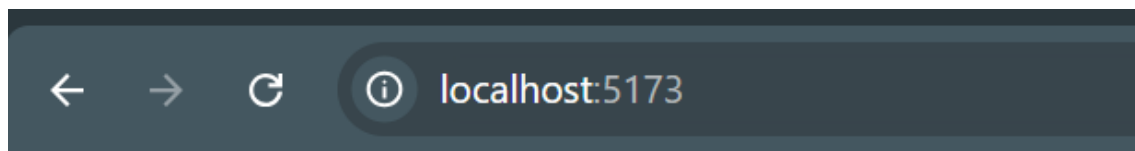
```
1
2 export default function HomeView(){
3   return(
4     <div>HomeView</div>
5   )
6 }
```



The screenshot shows a VS Code editor window with two tabs: 'HomeView.tsx' and 'router.tsx'. The 'router.tsx' tab is active, displaying the following code:

```
11 export default function () {  
15   <Route element={<AuthLayout/>}>  
16     <Route path='/auth/login' element={<LoginView/>} />  
17     <Route path='/auth/register' element={<RegisterView/>} />  
18   </Route>  
19  
20   <Route path='/admin' element={<AppLayout/>}>  
21     <Route index={true} element={<LinkTreeView/>} />  
22     <Route path='/profile' element={<ProfileView/>} />  
23   </Route>  
24  
25   <Route path='/:handle' element={<AuthLayout/>}>  
26     <Route element={<HandleView/>} index={true} />  
27   </Route>  
28  
29   <Route path='/' element={<HomeView/>} />  
30  
31   <Route path='/404' element={<AuthLayout/>}>  
32     <Route element={<NotFoundView/>} index={true} />  
33   </Route>  
34  
35 </Routes>  
36 </BrowserRouter>  
37 }  
38
```

The line `<Route path='/' element={<HomeView/>} />` at line 29 is highlighted with a red rectangle.



HomeView

Creamos un nuevo componente

The Explorer sidebar shows the project structure for 'THUNDER CLIENT'. The 'components' folder is expanded, showing files like DevTree.tsx, DevTreeInput.tsx, DevTreeLink.tsx, ErrorMessage.tsx, HandleData.tsx, Header.tsx, and NavigationTabs.tsx. The 'Header.tsx' file is selected and its content is displayed in the editor:

```
1 export default function Header(){
2   return(
3     <div>Header</div>
4   )
5 }
```

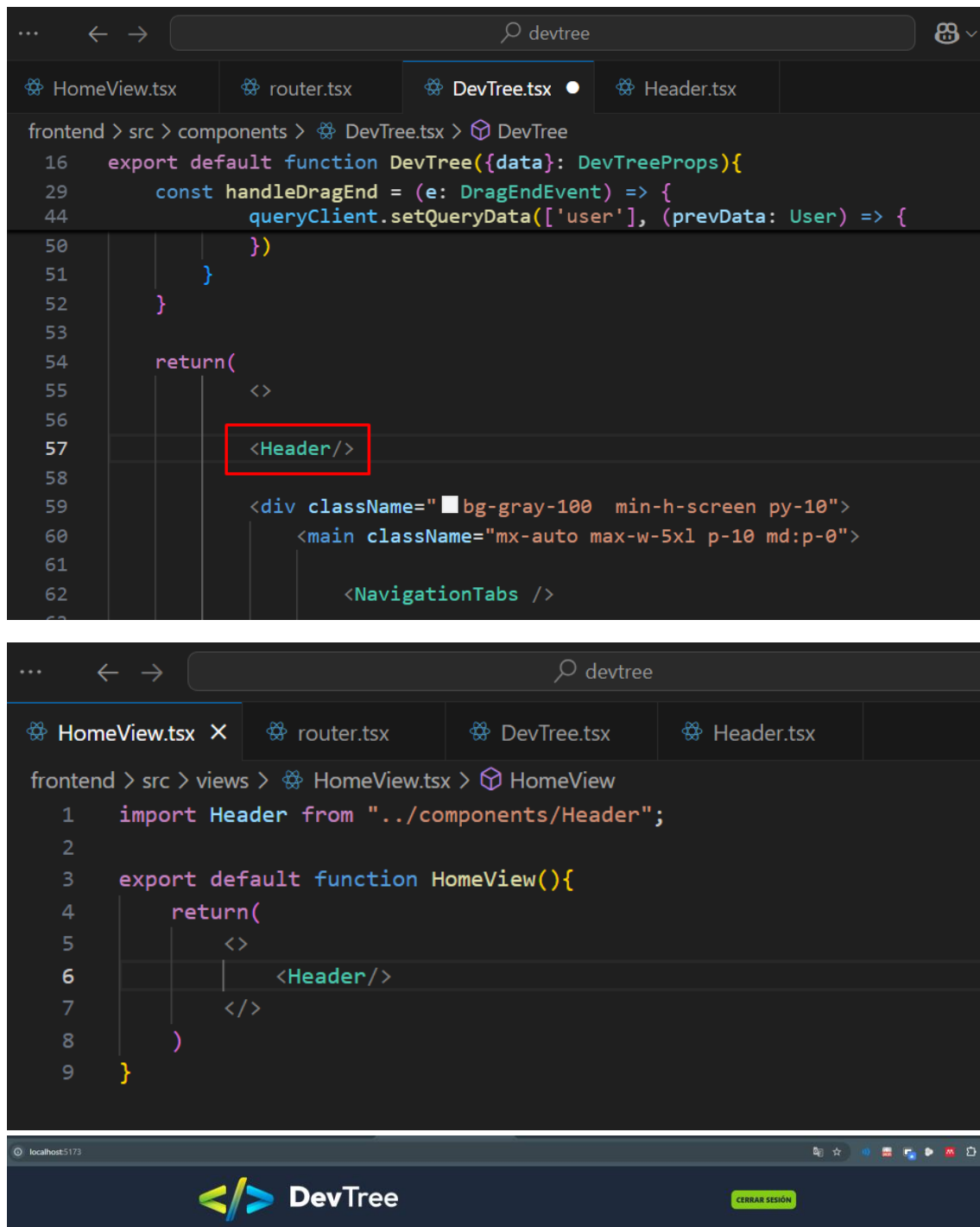
Del componente DevTree cortamos el header y lo pegamos en Header.tsx

The 'DevTree.tsx' file is selected in the Explorer. A red arrow points to the header section of the component's return statement. The code in the editor is:

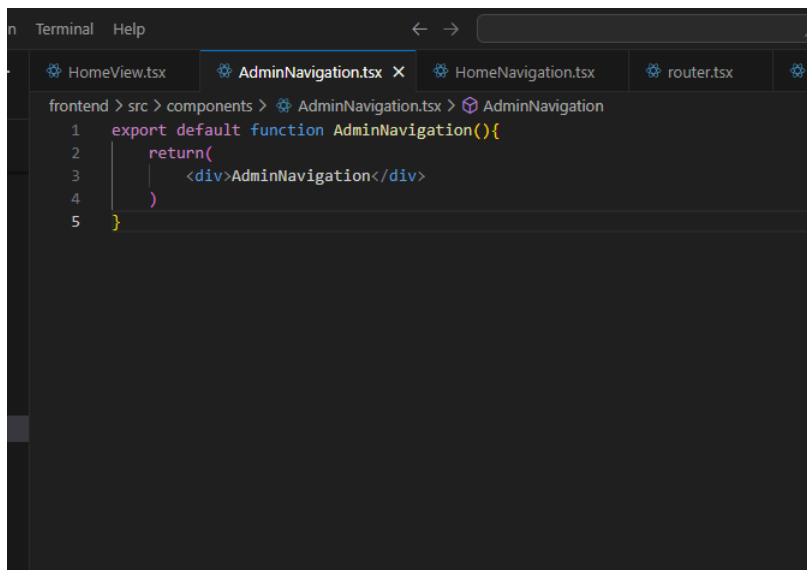
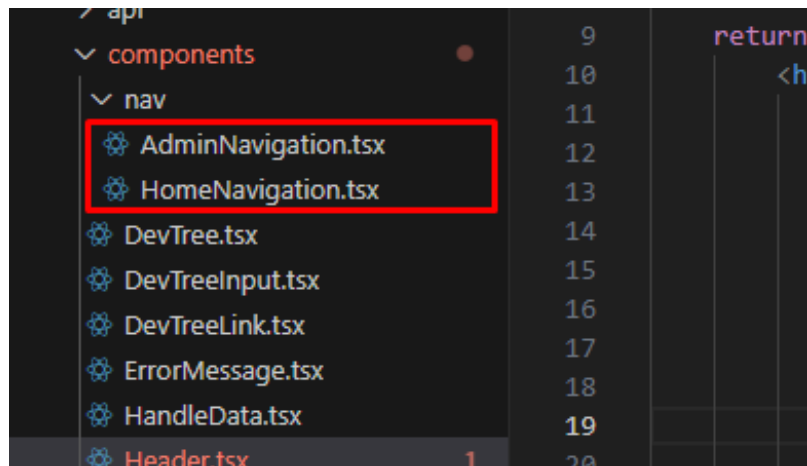
```
15 export default function DevTree((data): DevTreeProps){
16   const handleDragEnd = (e: DragEndEvent) => {
17     // ...
18   }
19   return(
20     <div className="mx-auto max-w-5xl flex flex-col md:flex-row items-center md:justify-between">
21       <div className="w-full p-5 lg:p-0 md:w-1/3">
22         
23       </div>
24       <div className="md:w-1/3 md:flex md:justify-end">
25         <button
26           className="bg-lime-500 p-2 text-slate-800 uppercase font-black text-xs rounded"
27           onClick={() => {}}
28         >
29           Cerrar Sesión
30         </button>
31       </div>
32     </div>
33   )
34 }
```

The 'Header.tsx' file is selected in the Explorer. The code in the editor now includes the header structure copied from DevTree.tsx:

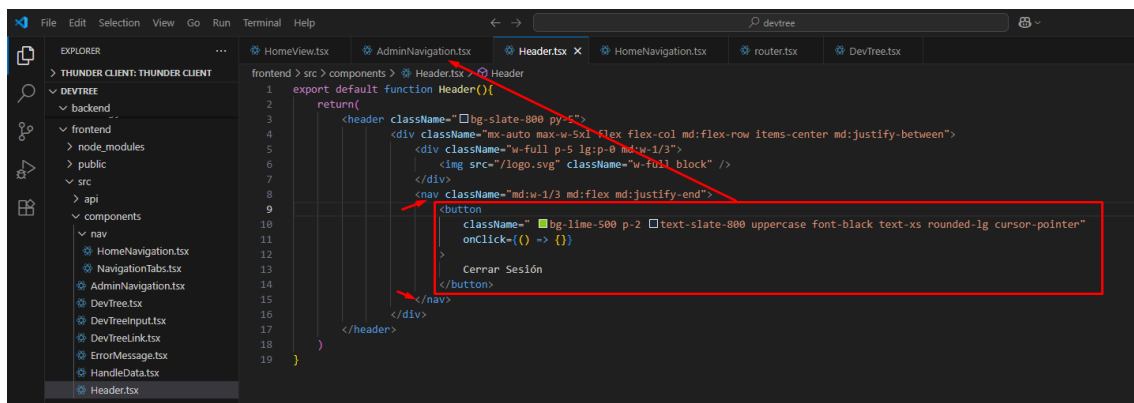
```
1 export default function Header(){
2   return(
3     <div className="mx-auto max-w-5xl flex flex-col md:flex-row items-center md:justify-between">
4       <div className="w-full p-5 lg:p-0 md:w-1/3">
5         
6       </div>
7       <div className="md:w-1/3 md:flex md:justify-end">
8         <button
9           className="bg-lime-500 p-2 text-slate-800 uppercase font-black text-xs rounded"
10          onClick={() => {}}
11        >
12          Cerrar Sesión
13        </button>
14      </div>
15    </div>
16  )
17 }
```



Pero, en esta ventana de inicio tenemos “Cerrar sesión”, y esto no necesitamos en la página principal. En realidad, tendría que llevarlo a crear una cuenta o iniciar sesión. Entonces:



Cortamos el botón de Header y lo llevamos al AdminNavigation



```

Terminal Help
AdminNavigation.tsx X Header.tsx HomeNavigation.tsx router.tsx DevTree.tsx
frontend > src > components > AdminNavigation.tsx > AdminNavigation
1 export default function AdminNavigation() {
2   return (
3     <button
4       className="bg-lime-500 p-2 text-slate-800 uppercase font-black text-xs rounded-lg cursor-pointer"
5       onClick={() => { }}
6     >
7       Cerrar Sesión
8     </button>
9   )
10 }

```

```

HomeView.tsx AdminNavigation.tsx Header.tsx HomeNavigation.tsx X router.tsx DevTree.tsx
frontend > src > components > nav > HomeNavigation.tsx > HomeNavigation
1 import {Link} from 'react-router-dom'
2
3 export default function HomeNavigation(){
4   return(
5     <>
6       <Link
7         className="text-white p-2 uppercase font-black text-xs cursor-pointer"
8         to="/auth/login"
9       >
10        Iniciar Sesión
11      </Link>
12    </>
13  )
14 }

```

Creando un componente reutilizable para el Logo y las Navegaciones

```

HomeView.tsx Header.tsx X AdminNavigation.tsx HomeNavigation.tsx router.tsx DevTree.tsx
frontend > src > components > Header.tsx > Header
1 import { useLocation } from "react-router-dom";
2 import AdminNavigation from "../nav/AdminNavigation";
3 import HomeNavigation from "../nav/HomeNavigation";
4
5 export default function Header(){
6
7   const location = useLocation()
8
9   return(
10     <header className="bg-slate-800 py-5">
11       <div className="mx-auto max-w-5xl flex flex-col md:flex-row items-center md:justify-between">
12         <div className="w-full p-5 lg:p-0 md:w-1/3">
13           
14         </div>
15         <nav className="md:w-1/3 md:flex md:justify-end">
16           {location.pathname === '/' ? <HomeNavigation/> : <AdminNavigation/>}
17         </nav>
18       </div>
19     </header>
20   )
21 }
22
23

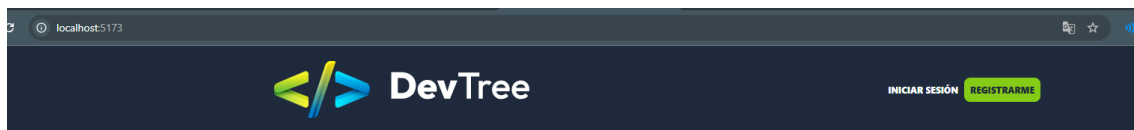
```

```

Terminal Help
HomeView.tsx Header.tsx AdminNavigation.tsx HomeNavigation.tsx router.tsx DevTree.tsx

frontend > src > components > nav > HomeNavigation.tsx > HomeNavigation
1 import {Link} from 'react-router-dom'
2
3 export default function HomeNavigation(){
4   return(
5     <>
6       <Link
7         className="text-white p-2 uppercase font-black text-xs cursor-pointer"
8         to="/auth/login"
9       >Iniciar Sesión</Link>
10
11       <Link
12         className="bg-lime-500 text-slate-800 p-2 uppercase font-black text-xs cursor-pointer rounded-lg"
13         to="/auth/register"
14       >Registrarme</Link>
15     </>
16   )
17 }
18
19

```



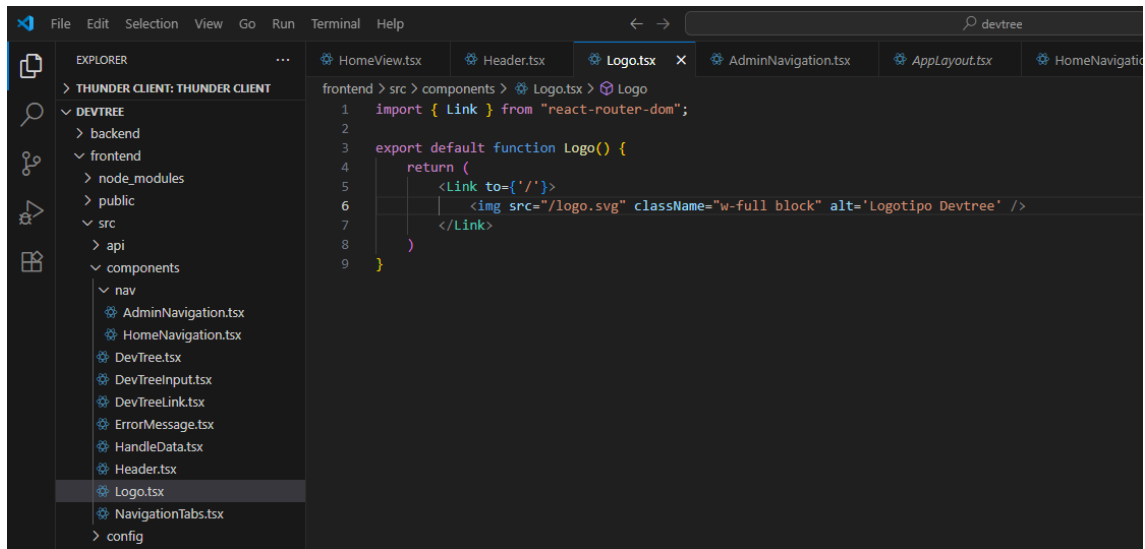
```

Terminal Help
HomeView.tsx Header.tsx AdminNavigation.tsx HomeNavigation.tsx router.tsx DevTree.tsx

frontend > src > components > Header.tsx > Header
1 import { Link, useLocation } from "react-router-dom";
2 import AdminNavigation from "../nav/AdminNavigation";
3 import HomeNavigation from "../nav/HomeNavigation";
4
5 export default function Header(){
6
7   const location = useLocation()
8
9   return(
10     <header className="bg-slate-800 py-5">
11       <div className="mx-auto max-w-5xl flex flex-col md:flex-row items-center md:justify-between">
12         <div className="w-full p-5 lg:p-0 md:w-1/3">
13
14           <Link to="/">
15             
16           </Link>
17
18         </div>
19         <nav className="md:w-1/3 md:flex md:justify-end">
20           {location.pathname === '/' ? <HomeNavigation/> : <AdminNavigation/>}
21         </nav>
22       </div>
23     </header>
24   )
25 }
26
27
28

```

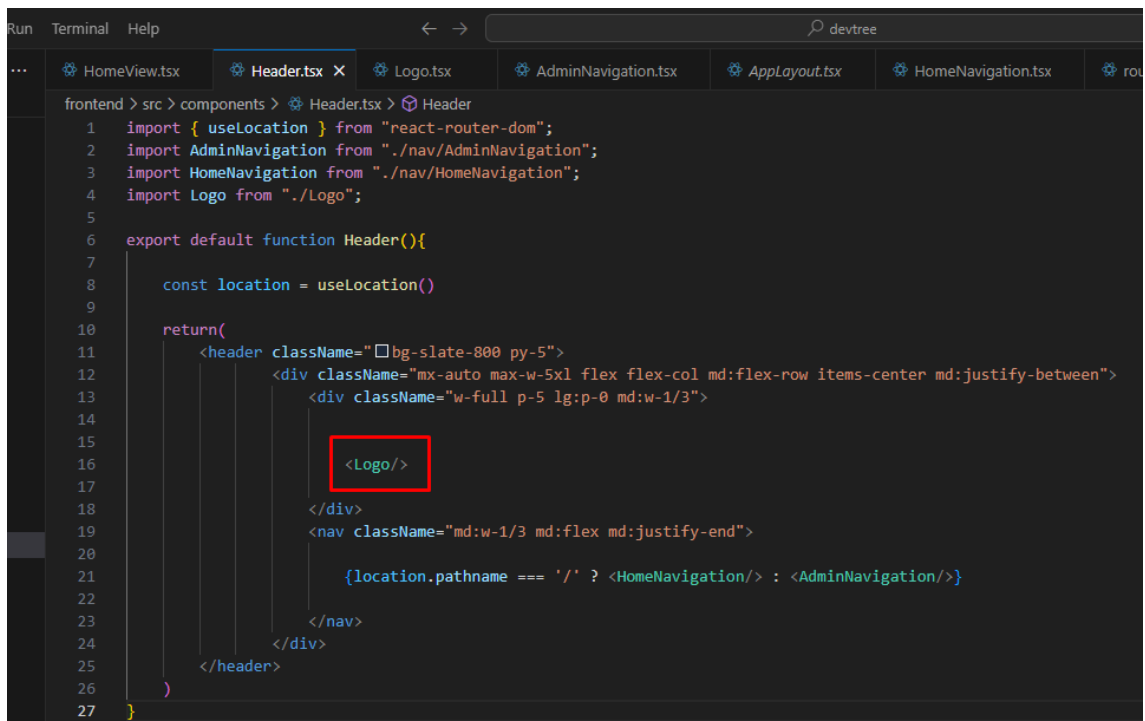
Pero, ese logo podría ser un componente:



```

1  import { Link } from "react-router-dom";
2
3  export default function Logo() {
4    return (
5      <Link to={'/'}>
6        
7      </Link>
8    )
9  }

```



```

1  import { useLocation } from "react-router-dom";
2  import AdminNavigation from "./nav/AdminNavigation";
3  import HomeNavigation from "./nav/HomeNavigation";
4  import Logo from "./Logo";
5
6  export default function Header(){
7
8    const location = useLocation()
9
10   return(
11     <header className="bg-slate-800 py-5">
12       <div className="mx-auto max-w-5xl flex flex-col md:flex-row items-center md:justify-between">
13         <div className="w-full p-5 lg:p-0 md:w-1/3">
14           <Logo/>
15         </div>
16         <nav className="md:w-1/3 md:flex md:justify-end">
17           {location.pathname === '/' ? <HomeNavigation/> : <AdminNavigation/>}
18         </nav>
19       </div>
20     </header>
21   )
22 }

```



```

1  import {Outlet} from 'react-router-dom'
2  import {Toaster} from 'sonner'
3  import Logo from '../components/Logo'
4
5  export default function AuthLayout(){
6    return(
7      <>
8        <div className='bg-slate-800 min-h-screen'>
9          <div className='max-w-lg mx-auto pt-10 px-5'>
10             <Logo/>
11          </div>
12          <div className='py-10'>
13             <Outlet/>
14          </div>
15        </div>
16        <Toaster position='top-right' />
17      </>
18    )
19  }
20
21

```

Primeros pasos con la página principal

```

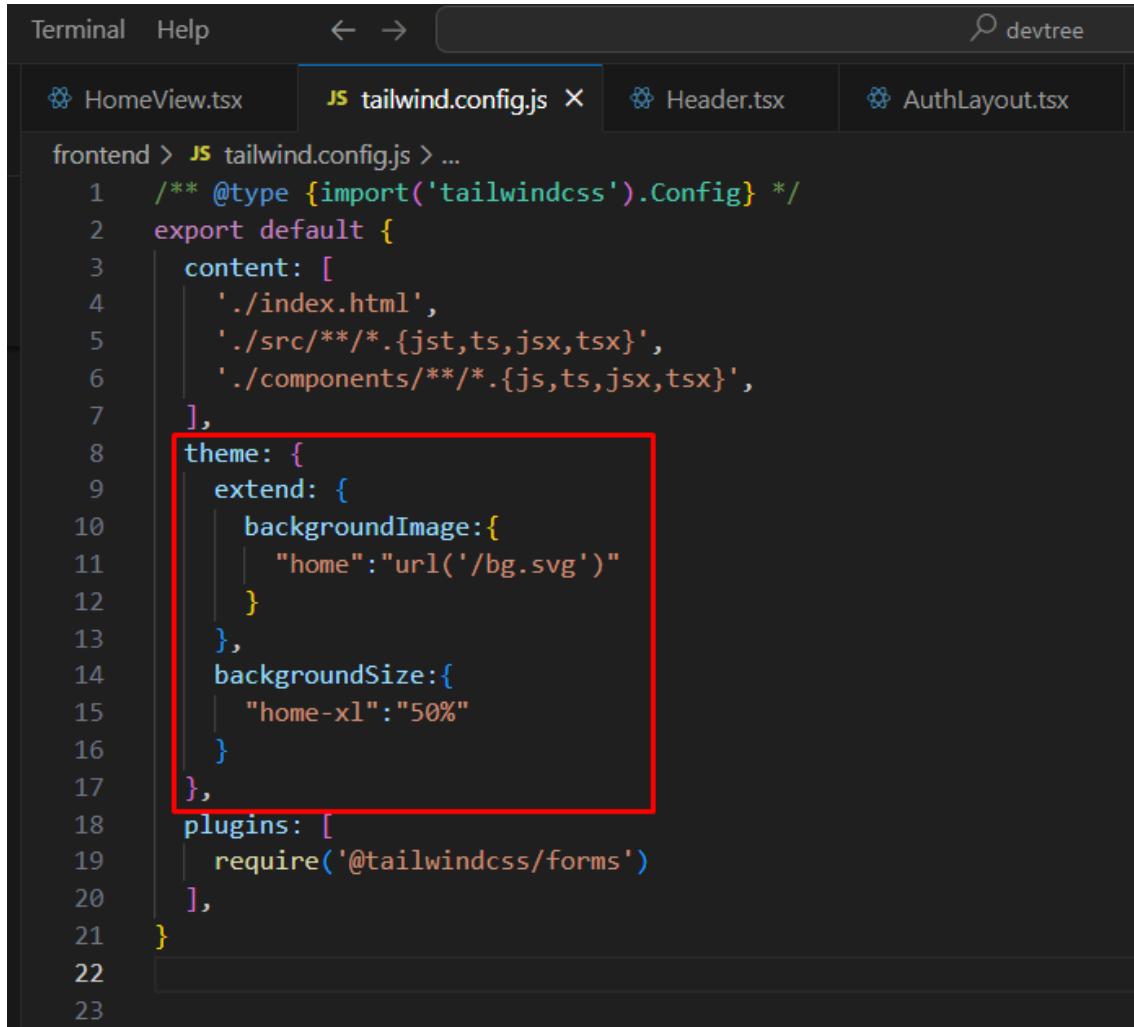
1  import Header from '../components/Header';
2
3  export default function HomeView(){
4    return(
5      <>
6        <Header/>
7
8        <main className='bg-gray-100 py-10 min-h-screen'>
9          <div className='max-w-5xl mx-auto mt-10'>
10            <div className='lg:w-1/2 px-10 lg:p-0 space-y-6'>
11              <h1 className='text-6xl font-black'>
12                Todas tus <span className='text-cyan-400'>Redes Sociales</span> en un enlace
13              </h1>
14              <p className='text-slate-800 text-xl'>Únete a más de 50 mil developers compartiendo sus redes sociales, comparte tu perfil de TikTok, Instagram, Github y más.</p>
15            </div>
16          </div>
17        </main>
18      </>
19    )
20  }
21
22
23
24

```



Pero, podríamos crear nuestras propias clases de Tailwind, eso se conoce como Theming y tú puedes crear tus propias clases.

Nosotros tenemos que agregar una imagen de fondo para nuestro componente de React y mostrarlo en la página que estamos creando.



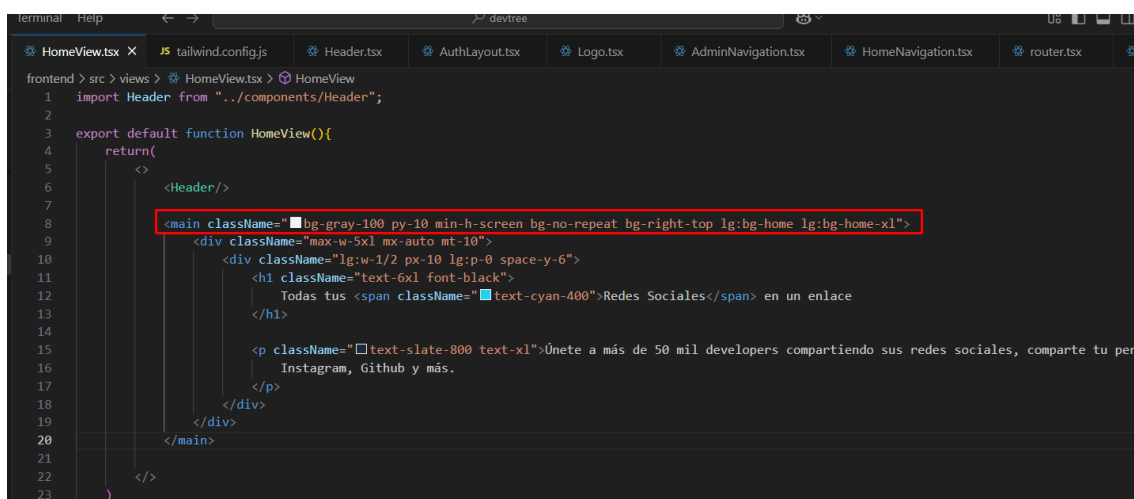
```

Terminal  Help  < ->  devtree

HomeView.tsx  JS tailwind.config.js  Header.tsx  AuthLayout.tsx

frontend > JS tailwind.config.js > ...
1  /** @type {import('tailwindcss').Config} */
2  export default {
3    content: [
4      './index.html',
5      './src/**/*.{jst,ts,jsx,tsx}',
6      './components/**/*.{js,ts,jsx,tsx}',
7    ],
8    theme: {
9      extend: {
10       backgroundImage:{
11         "home": "url('/bg.svg')"
12       },
13     },
14     backgroundSize:{
15       "home-xl": "50%"
16     }
17   },
18   plugins: [
19     require('@tailwindcss/forms')
20   ],
21 }
22
23

```



```

Terminal  Help  < ->  devtree

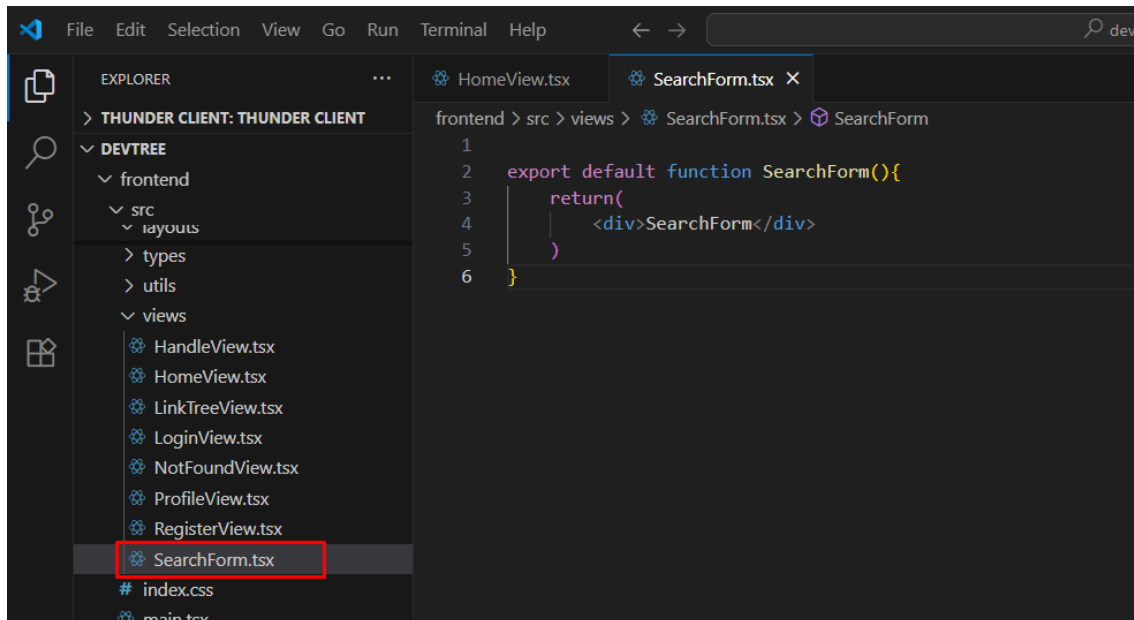
HomeView.tsx  JS tailwind.config.js  Header.tsx  AuthLayout.tsx  Logo.tsx  AdminNavigation.tsx  HomeNavigation.tsx  router.tsx

frontend > src > views > HomeView.tsx > HomeView
1  import Header from "../components/Header";
2
3  export default function HomeView(){
4    return(
5      <>
6        <Header/>
7
8        <main className="bg-gray-100 py-10 min-h-screen bg-no-repeat bg-right-top lg:bg-home lg:bg-home-xl">
9          <div className="max-w-5xl mx-auto mt-10">
10             <div className="lg:w-1/2 lg:p-0 space-y-6">
11               <h1 className="text-6xl font-black">
12                 Todas tus <span className="text-cyan-400">Redes Sociales</span> en un enlace
13             </h1>
14
15             <p className="text-slate-800 text-xl">Únete a más de 50 mil developers compartiendo sus redes sociales, comparte tu perfil
16               Instagram, Github y más.
17             </p>
18           </div>
19         </div>
20       </main>
21     </>
22   )
23

```



Añadir buscador de handles



```

Terminal  Help  ←  →  devtree
HomeView.tsx  SearchForm.tsx
frontend > src > views > HomeView.tsx > HomeView
1  import Header from "../components/Header";
2  import SearchForm from "../SearchForm";
3
4  export default function HomeView(){
5    return(
6      <>
7        <Header/>
8
9        <main className="bg-gray-100 py-10 min-h-screen bg-no-repeat bg-right-top lg:bg-home lg:bg-home-xl">
10         <div className="max-w-5xl mx-auto mt-10">
11           <div className="lg:w-1/2 px-10 lg:p-0 space-y-6">
12             <h1 className="text-6xl font-black">
13               Todas tus <span className="text-cyan-400">Redes Sociales</span> en un enlace
14             </h1>
15
16             <p className="text-slate-800 text-xl">Únete a más de 50 mil developers compartiendo sus
17               Instagram, Github y más.
18             </p>
19
20             <SearchForm/>
21           </div>
22         </div>
23       </main>
24     </>
25   )
26 }
27

```

En SearchForm, copiamos:

```

<form
  onSubmit={() => {}}
  className="space-y-5">
  <div className="relative flex items-center bg-white px-2">
    <label
      htmlFor="handle"
    >devtree.com/</label>
    <input
      type="text"
      id="handle"
      className="border-none bg-transparent p-2 focus:ring-0 flex-1"
      placeholder="elonmusk, zuck, jeffbezos"
      {...register("handle", {
        required: "Un Nombre de Usuario es obligatorio",
      })}
    />
  </div>
  {errors.handle && (
    <ErrorMessage>{errors.handle.message}</ErrorMessage>
  )}
  <div className="mt-10">

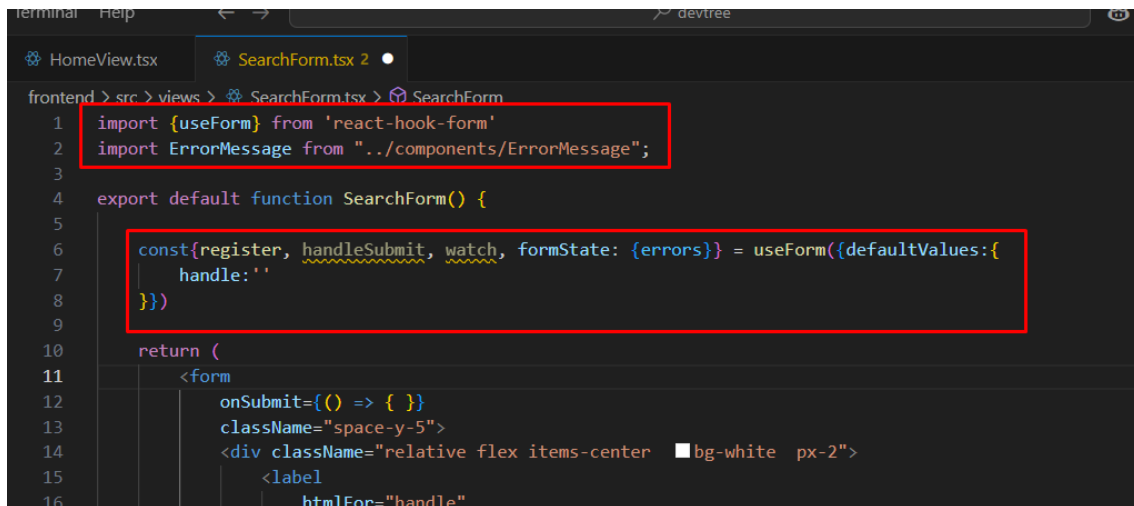
</div>

```

```

<input
  type="submit"
  className="bg-cyan-400 p-3 text-lg w-full uppercase text-slate-600 rounded-lg
  font-bold cursor-pointer"
  value='Obtener mi DevTree'
/>
</form>

```



```

1  import {useForm} from 'react-hook-form'
2  import ErrorMessage from "../components/ErrorMessage";
3
4  export default function SearchForm() {
5
6      const{register, handleSubmit, watch, formState: {errors}} = useForm({defaultValues:{
7          handle:''
8      }})
9
10     return (
11         <form
12             onSubmit={() => { }}
13             className="space-y-5">
14             <div className="relative flex items-center bg-white px-2">
15                 <label
16                     htmlFor="handle"

```



DevTree

Todas tus Redes Sociales en un enlace

Únete a más de 50 mil developers compartiendo sus redes sociales, comparte tu perfil de TikTok, Instagram, Github y más.

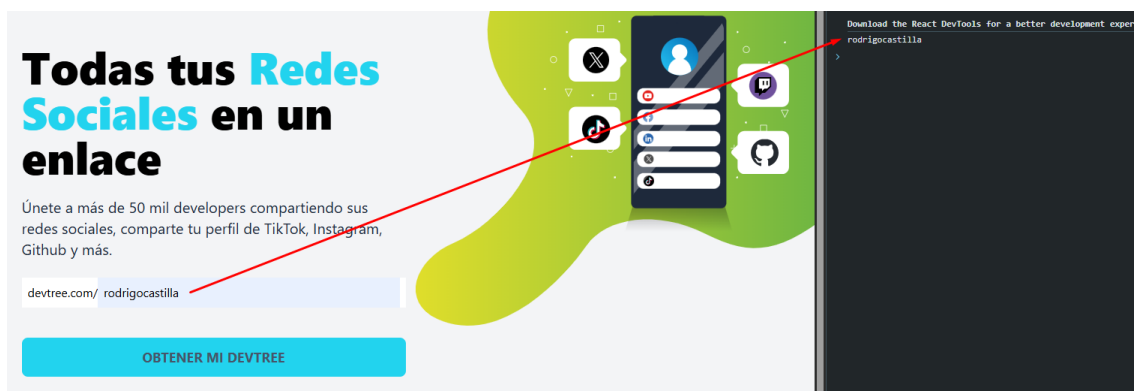
devtree.com/ elonmusk, zuck, jeffbezoz

OBTENER MI DEVTREE

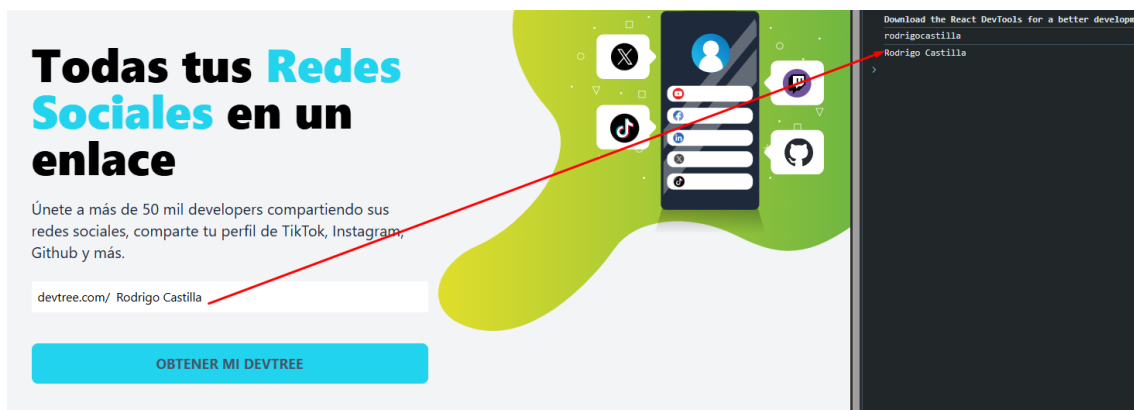
```

1 import {useForm} from 'react-hook-form'
2 import ErrorMessage from "../components/ErrorMessage";
3
4 export default function SearchForm() {
5
6     const {register, handleSubmit, watch, formState: {errors}} = useForm({defaultValues:{
7         handle:''
8     }})
9
10    const handle = watch('handle')
11
12    const handleSearch = () => {
13        console.log(handle)
14    }
15
16    return (
17        <form
18            onSubmit={handleSubmit(handleSearch)}
19            className="space-y-5">

```



Si colocamos con espacios, igual lo muestra:



Entonces, debemos de convertir esta respuesta en un slug.

```

• PS C:\Users\rodri\devtree> cd frontend
• PS C:\Users\rodri\devtree\frontend> npm i react-slugify

added 2 packages, and audited 322 packages in 3s

87 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
• PS C:\Users\rodri\devtree\frontend> 

```

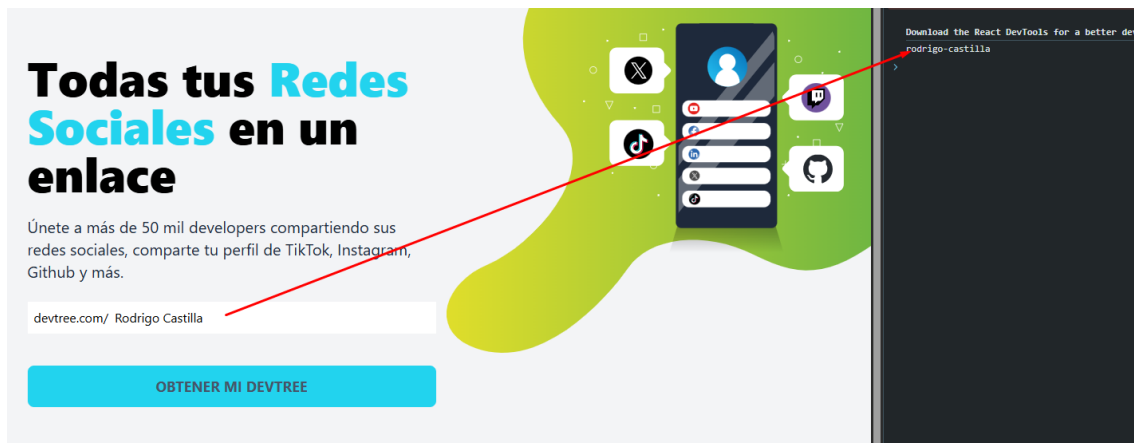
```

Terminal Help  ← →  devtree

HomeView.tsx  SearchForm.tsx •

frontend > src > views > SearchForm.tsx > SearchForm
1  import {useForm} from 'react-hook-form'
2  import slugify from 'react-slugify'
3  import ErrorMessage from '../components/ErrorMessage';
4
5  export default function SearchForm() {
6
7      const {register, handleSubmit, watch, formState: {errors}} = useForm({defaultValues:{
8          handle:''
9      }})
10
11     const handle = watch('handle')
12
13     const handleSearch = () => {
14         const slug = slugify(handle)
15         console.log(slug)
16     }
17
18     return (

```



Creando la ruta para revisar si el handle está tomado o disponible

En el back:

```

45     handleInputErrors,
46     authenticate,
47     updateProfile)
48
49     router.post('/user/image', authenticate, uploadImage)
50
51     //routing dinámico
52     router.get('/:handle', getUserByHandle)
53
54     router.post('/search',
55       body('handle').notEmpty().withMessage('El handle no puede ir vacío'),
56       handleInputErrors,
57       searchByHandle
58     )
59
60     export default router
61
62

```

```

138 export const getUserByHandle = async (req:Request, res:Response)=>{
139   return res.status(500).json({error: error.message})
140 }
141
142
143
144
145 export const searchByHandle = async (req:Request, res:Response)=>{
146   try{
147     const {handle} = req.body
148     const userExists = await User.findOne({handle})
149     if(userExists){
150       const error = new Error(`${handle} ya está registrado`)
151       return res.statud(409).json({error: error.message})
152     }
153     res.send(`${handle} está disponible`)
154   }catch(e){
155     const error = new Error("Hubo un error")
156     return res.status(500).json({error: error.message})
157   }
158 }
159
160

```

Añadiendo funcionalidad para ir al registro

Movemos SearchForm a componentes (Me confundí).


```

TS index.ts  TS DevTreeAPI.ts X  SearchForm.tsx  TS router.ts
frontend > src > api > TS DevTreeAPI.ts > searchByHandle
49 export async function getUserByHandle(handle: string){
50   }
51   throw new Error("No se pudo obtener el usuario")
52 }
53
54
55
56
57
58
59 export async function searchByHandle(handle: string){
60   try{
61     const {data} = await api.post('/search', {handle})
62     return data
63   }catch(error){
64     if (isAxiosError(error) && error.response) {
65       throw new Error(error.response.data.error)
66     }
67     throw new Error("No se pudo obtener el usuario")
68   }
69 }
70
71
72
73
74
75
76
77

```

```

terminal  Help  < - ->  devtree
TS index.ts  TS DevTreeAPI.ts  SearchForm.tsx X  TS router.ts
frontend > src > components > SearchForm.tsx > SearchForm
1  import {useForm} from 'react-hook-form'
2  import slugify from 'react-slugify'
3  import { useMutation } from '@tanstack/react-query'
4  import ErrorMessage from "../components/ErrorMessage";
5  import { searchByHandle } from '../api/DevTreeAPI';
6
7  export default function SearchForm() {
8
9    const{register, handleSubmit, watch, formState: {errors}} = useForm({defaultValues:{
10      handle:''
11    }})
12
13    const mutation = useMutation({
14      mutationFn: searchByHandle
15    })
16
17    const handle = watch('handle')
18
19    const handleSearch = () => {
20      const slug = slugify(handle)
21      mutation.mutate(slug)
22    }
23
24    return (
25      <form
26        onSubmit={handleSubmit(handleSearch)}
27        className="space-y-5">
28        <div className="relative flex items-center  bg-white px-2">
29          <label
30            htmlFor="handle"

```

```

Terminal  Help  < ->  devtree
TS index.ts  TS DevTreeAPLts  SearchForm.tsx  TS router.ts

frontend > src > components > SearchForm.tsx > SearchForm
8  export default function SearchForm() {
35      id= 'handle'
36      className="border-none bg-transparent p-2 focus:ring-0 flex-1"
37      placeholder="elonmusk, zuck, jeffbezos"
38      {...register("handle", {
39          required: "Un Nombre de Usuario es obligatorio",
40      })}
41  }
42  />
43
44  </div>
45  {errors.handle && (
46      <ErrorMessage>{errors.handle.message}</ErrorMessage>
47  )}
48
49  <div className="mt-10">
50      {mutation.isPending && <p className='text-center'>Cargando...</p>}
51      {mutation.error && <p className='text-center text-red-600 font-black'>{mutation.error.message}</p>}
52      {mutation.data && <p className='text-center text-cyan-500 font-black'>{mutation.data}
53      ir a <Link to={'/auth/register'}>Registro</Link></p>}
54  </div>
55
56  <input
57      type="submit"
58      className="bg-cyan-400 p-3 text-lg w-full uppercase text-slate-600 rounded-lg font-bold cursor-pointer"
59      value='Obtener mi DevTree'
60  />
61  </form>
62  }

```



Todas tus Redes Sociales en un enlace

Únete a más de 50 mil developers compartiendo sus redes sociales, comparte tu perfil de TikTok, Instagram, Github y más.

devtree.com/ rodrigocastilla

rodrigocastilla ya está registrado

OBTENER MI DEVTREE

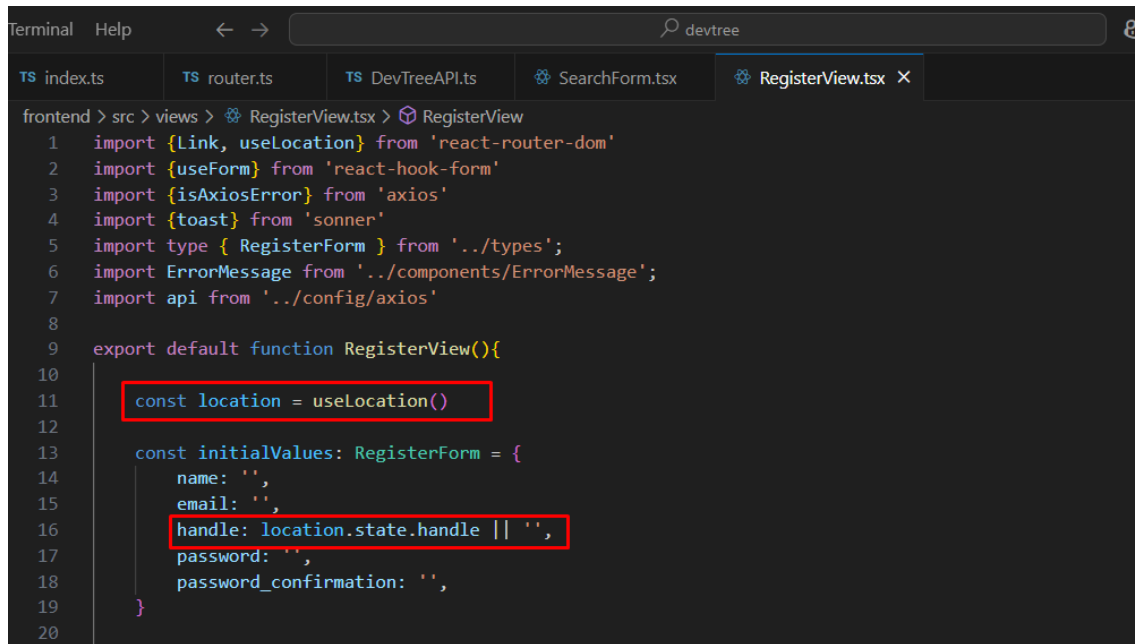


Llenar el handle desde el home si está disponible

```

Terminal Help  ← →  devtree
TS index.ts  TS router.ts  TS DevTreeAPI.ts  SearchForm.tsx
frontend > src > components > SearchForm.tsx > SearchForm
8  export default function SearchForm() {
40      }
41  }
42  />
43  </div>
44  {errors.handle && (
45    <ErrorMessage>{errors.handle.message}</ErrorMessage>
46  )}
47
48  <div className="mt-10">
49    {mutation.isPending && <p className='text-center'>Cargando...</p>}
50    {mutation.error && <p className='text-center text-red-600 font-black'>{mutation.error.message}</p>}
51    {mutation.data && <p className='text-center text-cyan-500 font-black'>{mutation.data}
52    ir a <Link to={'/auth/register'} state={{handle: slugify(handle)}}>registro</Link></p>}
53  </div>
54
55  <input
56    type="submit"
57    className="bg-cyan-400 p-3 text-lg w-full uppercase text-slate-600 rounded-lg font-bold cursor-pointer"
58    value="Obtener mi DevTree"
59  />
60  </form>
61  )
62  }

```



```
Terminal  Help  ← →  devtree  TS index.ts  TS router.ts  TS DevTreeAPI.ts  SearchForm.tsx  RegisterView.tsx X

frontend > src > views > RegisterView.tsx > RegisterView
1  import {Link, useLocation} from 'react-router-dom'
2  import {useForm} from 'react-hook-form'
3  import {isAxiosError} from 'axios'
4  import {toast} from 'sonner'
5  import type { RegisterForm } from '../types';
6  import ErrorMessage from '../components/ErrorMessage';
7  import api from '../config/axios'
8
9  export default function RegisterView(){
10
11      const location = useLocation()
12
13      const initialValues: RegisterForm = {
14          name: '',
15          email: '',
16          handle: location.state.handle || '',
17          password: '',
18          password_confirmation: '',
19      }
20
```

Todas tus Redes Sociales en un enlace

Únete a más de 50 mil developers compartiendo sus redes sociales, comparte tu perfil de TikTok, Instagram, Github y más.

devtree.com/ edgarcastilla

edgarcastilla está disponible ir a Registro

OBTENER MI DEVTREE

Crear cuenta

Nombre

Tu Nombre

E-mail

Email de Registro

Handle

edgarcastilla

Password

Password de Registro

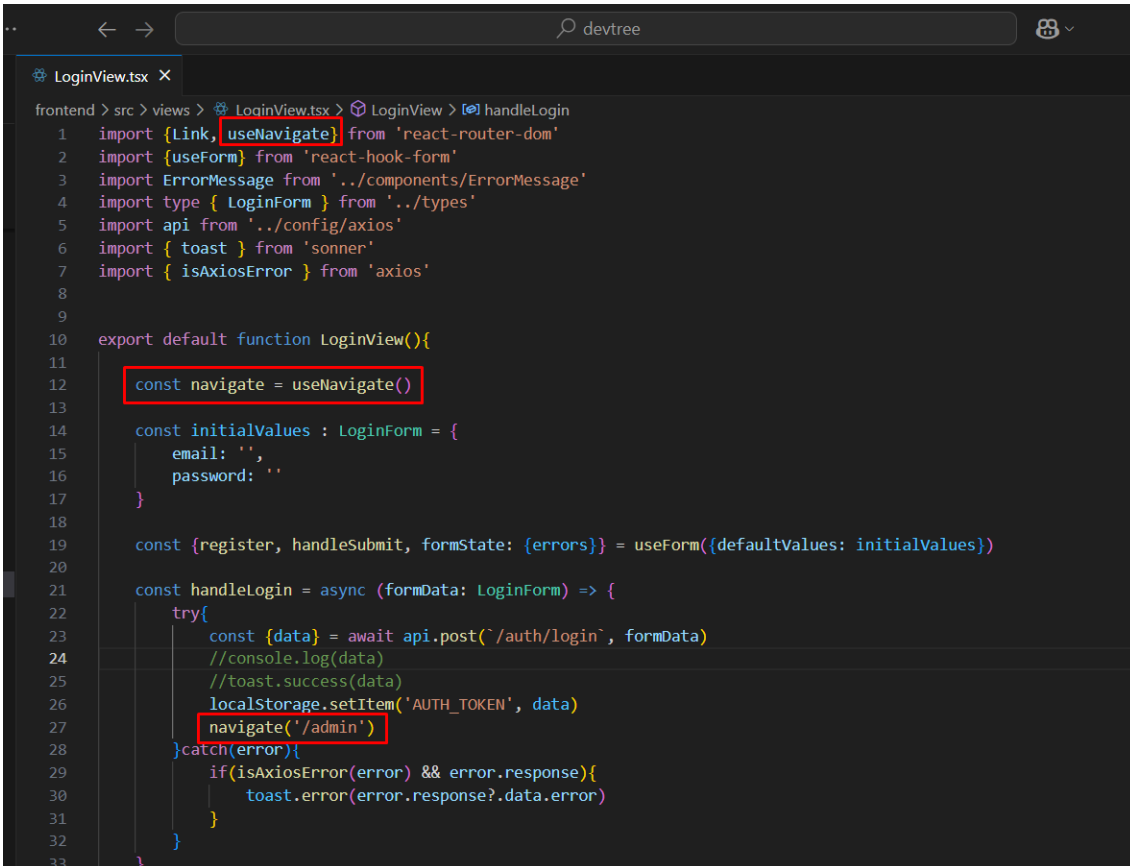
Cerrando sesión del usuario

Hay dos cosas que son necesarias para cerrar la sesión, uno es limpiar el token y lo otro, en AppLayout tenemos un usuario que hay que invalidarlo.

```
AdminNavigation.tsx
1 import {useQueryClient} from '@tanstack/react-query'
2
3 export default function AdminNavigation() {
4
5   const queryClient = useQueryClient()
6
7   const logout = () => {
8     localStorage.removeItem('AUTH_TOKEN')
9     queryClient.invalidateQueries({queryKey: ['user']})
10  }
11
12  return (
13    <button
14      className="bg-lime-500 p-2 text-slate-800 uppercase font-black text-xs rounded-lg cursor-p
15      onClick={logout}
16    >
17      Cerrar Sesión
18    </button>
19  )
20 }
```

Últimos ajustes

Redireccionar al usuario cuando inicie sesión



```
1 import {Link, useNavigate} from 'react-router-dom'
2 import {useForm} from 'react-hook-form'
3 import ErrorMessage from '../components/ErrorMessage'
4 import type { LoginForm } from '../types'
5 import api from '../config/axios'
6 import { toast } from 'sonner'
7 import { isAxiosError } from 'axios'
8
9
10 export default function LoginView(){
11   const navigate = useNavigate()
12
13   const initialValues : LoginForm = {
14     email: '',
15     password: ''
16   }
17
18   const {register, handleSubmit, formState: {errors}} = useForm({defaultValues: initialValues})
19
20   const handleLogin = async (formData: LoginForm) => {
21     try{
22       const {data} = await api.post('/auth/login', formData)
23       //console.log(data)
24       //toast.success(data)
25       localStorage.setItem('AUTH_TOKEN', data)
26       navigate('/admin')
27     }catch(error){
28       if(isAxiosError(error) && error.response){
29         toast.error(error.response?.data.error)
30       }
31     }
32   }
33 }
```

Ahora, algunas veces, este paso de iniciar sesión en lo que genera el token y lo envía, no lo hace uno tras otro, no espera a que genere el token y después trate de iniciar sesión. En esos casos lo puedes solucionar incrementando tiempo/intentos y que ese JWT se almacene en LocalStorage para que entonces lo mande.

```

1
2  import { useQuery } from '@tanstack/react-query'
3  import { Navigate } from 'react-router-dom'
4
5  import { getUser } from '../api/DevTreeAPI';
6  import DevTree from '../components/DevTree';
7
8  export default function AppLayout() {
9
10     const {data, isLoading, isError} = useQuery({
11         //pasarle algunas configuraciones
12         queryFn: getUser, //qué función va a hacer la consulta
13         queryKey: ['user'], //reactquery identifica el query
14         retry: 2, //cuantas veces queremos que react intente
15         refetchOnWindowFocus: false //no queremos que se haga
16     })
17

```

Ahora, tenemos un error cuando damos clic en registrar al usuario.

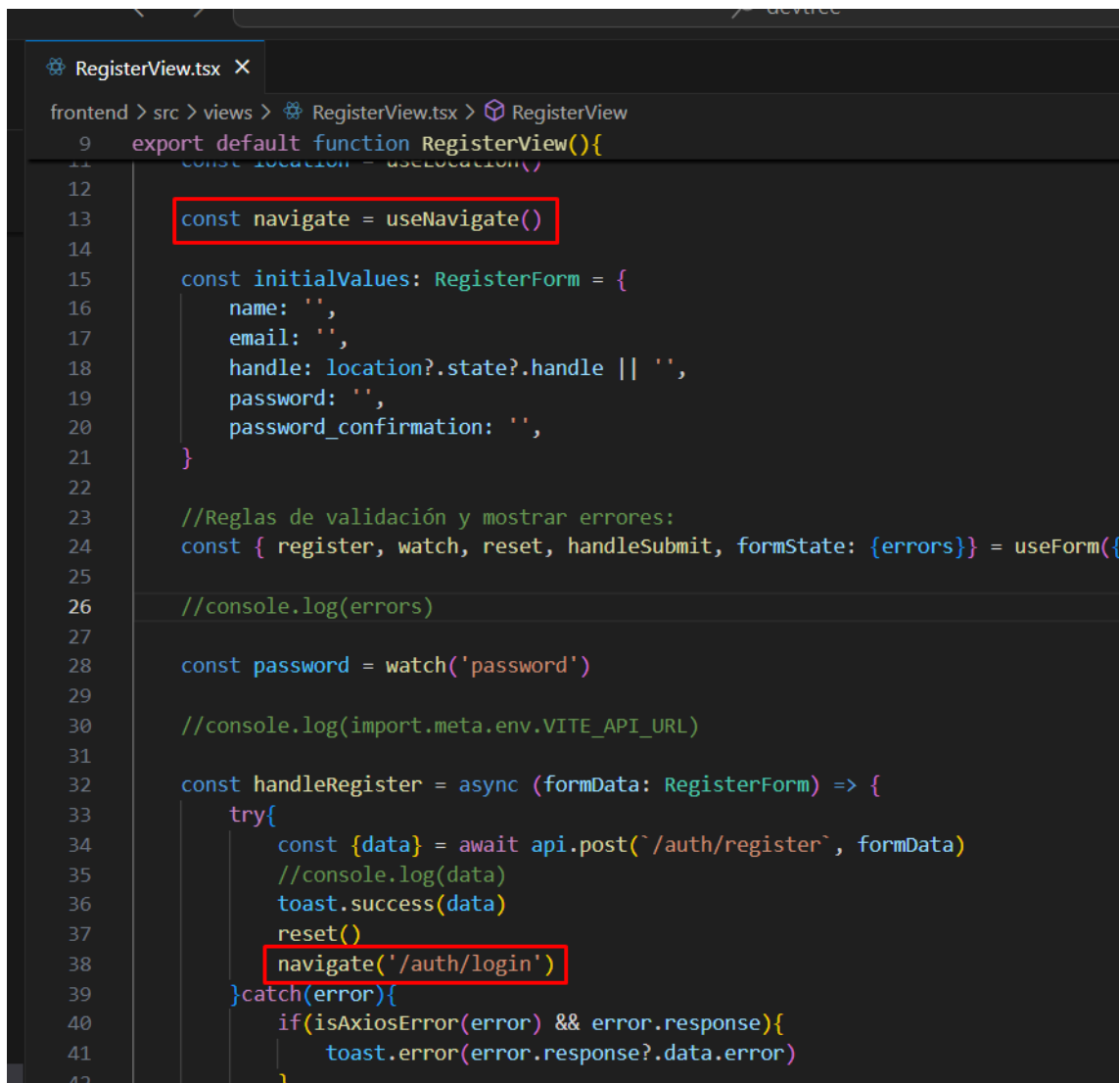
Vamos a RegisterView y utilizaremos el OptionalChaining para ir entrando a cada uno de los valores en caso de que existan únicamente. Probamos y ya no tenemos el error:

```

1  import {Link, useLocation} from 'react-router-dom'
2  import {useForm} from 'react-hook-form'
3  import {isAxiosError} from 'axios'
4  import {toast} from 'sonner'
5  import type { RegisterForm } from '../types';
6  import ErrorMessage from '../components/ErrorMessage';
7  import api from '../config/axios'
8
9  export default function RegisterView(){
10
11     const location = useLocation()
12
13     const initialValues: RegisterForm = {
14         name: '',
15         email: '',
16         handle: location?.state?.handle || '',
17         password: '',
18         password_confirmation: '',
19     }
20
21     //Reglas de validación y mostrar errores:
22     const { register, watch, reset, handleSubmit, formState: {errors}}

```

Al registrarnos debe llevarnos directamente al Login:



```
9 export default function RegisterView(){
10   const location = useLocation()
11
12
13   const navigate = useNavigate()
14
15   const initialValues: RegisterForm = {
16     name: '',
17     email: '',
18     handle: location?.state?.handle || '',
19     password: '',
20     password_confirmation: '',
21   }
22
23   //Reglas de validación y mostrar errores:
24   const { register, watch, reset, handleSubmit, formState: {errors}} = useForm({
25
26     //console.log(errors)
27
28     const password = watch('password')
29
30     //console.log(import.meta.env.VITE_API_URL)
31
32     const handleRegister = async (formData: RegisterForm) => {
33       try{
34         const {data} = await api.post(`/auth/register`, formData)
35         //console.log(data)
36         toast.success(data)
37         reset()
38         navigate('/auth/login')
39       }catch(error){
40         if(isAxiosError(error) && error.response){
41           toast.error(error.response?.data.error)
42         }
43       }
44     }
45   })
46 }
```