

Introducción al desarrollo Web con Ant Design Header

git branch Navbar

Este es un ejemplo de cómo podemos implementar el header de la aplicación con un Navbar, incluyendo la opción del icono user para cerrar sesión o acceder a su perfil.

Lo primero será construir el componente Nav dentro de components, dentro de los imports, vamos a considerar **Link** de *react-router-dom* Layout y Menú de Antd y acceder a Header de Layout. Almacena una imagen que será el logo de tu aplicación.

```
src > components > Nav > index.jsx > ...
1  import React from 'react'
2  import { Link } from 'react-router-dom';
3  import { Layout, Menu } from 'antd';
4  const { Header } = Layout;
5  import logo from '../assets/logo.png'
6  import './Nav.css';
7
8
9  const Nav = () => {
10   return (
11     <div>
12       <h1>Nav</h1>
13     </div>
14   );
15 }
16
17 export default Nav;
```

Para comenzar, dentro del componente Nav crea una constante **tabNames** que va a almacenar los nombres de los “Tabs” o Links que van a componer el Navbar, así mismo crea la constante **items** que nos servirá para iterar cada uno de los tabs y Generar URLs basadas en el nombre del item.

```
const tabNames = ['', 'Productos', 'Servicios', 'Contacto'];
const items = tabNames.map((name, index) => ({
  key: index + 1,
  label: name,
  url: index === 0 ? "/" : `/${name.toLowerCase()}`,
}));
```

Dentro del return envolvemos el contenido en el componente Layout y Header de Antd, Se incluye un campo link que envuelve el Logo de la app para volver siempre al Home. Dentro incluir el Menú con las propiedades de configuración que más te convenga al diseño de tu aplicación, este menú lo que mostrará será item iterando a través de key.

```
<Header className='header-content'  
>  
  <Link to="/">  
    <img src={logo} alt="logo" />  
  </Link>  
  <Menu  
    theme="light"  
    mode="horizontal"  
    defaultSelectedKeys={['1']}  
    style={{  
      display: 'flex',  
      justifyContent: 'flex-end',  
      flex: 1,  
      minWidth: 0,  
      marginRight: '20px'  
    }}  
  >  
    {items.map(item => (  
      <Menu.Item key={item.key}>  
        <Link to={item.url}>{item.label}</Link>  
      </Menu.Item>  
    ))}  
  </Menu>
```

Dentro de el Navbar en el icono user abrirá un Drawer para cerrar sesión o acceder a su perfil.

De Antd accedemos al componente y de manera inicial, nos quedaría de esta manera, como viene en la documentación de Antd solo estamos añadiendo que sea el Avatar de User quien reciba la función de *showDrawer*.

src/components/Drawer

```
src > components > Drawer > index.jsx > ...  
1  import React, { useState } from 'react';  
2  import { Drawer, Avatar } from 'antd';  
3  import { UserOutlined } from '@ant-design/icons';  
4  
5  const DrawerComponent = () => {  
6    const [open, setOpen] = useState(false);  
7    const showDrawer = () => {  
8      setOpen(true);  
9    };  
10   const onClose = () => {  
11     setOpen(false);  
12   };  
13   return (  
14     <>  
15       <Avatar  
16         onClick={showDrawer}  
17         size={44}  
18         style={{ backgroundColor: '#87d068', cursor: 'pointer' }}  
19         icon={<UserOutlined />}  
20       />  
21       <Drawer title="Basic Drawer" onClose={onClose} open={open}>  
22         <p>Some contents...</p>  
23         <p>Some contents...</p>  
24         <p>Some contents...</p>  
25       </Drawer>  
26     </>  
27   );  
28 };  
29 export default DrawerComponent;
```

Una vez creado este componente, lo importamos en el componente Nav y antes de que se cierre la etiqueta Header los mandamos llamar.

Estos son los estilos para personalizar un poco más el componente Header

```
src > components > Nav > Nav.css > .header-content
1  .header-content{
2      display: flex;
3      justify-content: space-between;
4      align-items: center;
5      padding: 40px;
6      background-color: #fff;
7      height: 60px;
8  }
```

Ahora solo queda crear las rutas que se están creando a través de los Tabs en nuestro archivo de routes, crear las vistas en pages, y utilizar el componente Nav en cada uno de ellos.

Para terminar esta práctica, pasa estos cambios a development y main crea una rama llamada Drawer y personaliza el componente Drawer, accediendo a los datos del usuario y usando la función de log out para cerrar sesión.