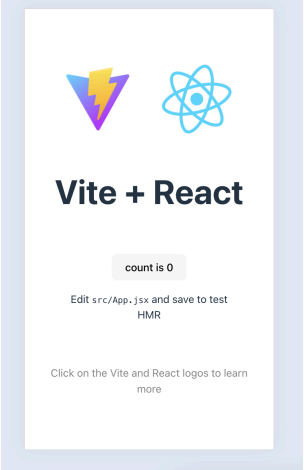
Desarrollo Web
Mayo- Agosto 2024
Ing. Ana Sol Arteaga

Introducción al desarrollo Web con Ant Design

Para este ejemplo, vamos a usar un proyecto usando el framework de Vite para desarrollo con React y Js, para crear el proyecto y de acuerdo a la documentación oficial de Vite desde terminal usaremos el siguiente comando ***npm create vite@latest*** seguido del nombre de la carpeta, y continuamos con las instrucciones que veamos en terminal hasta ver la aplicación corriendo.



Crear una rama llamada **development**

Crear la rama layout

Cambiarse a la rama layout y comenzar con la configuración.

Una vez que creamos el proyecto, nos dirigimos a la página oficial de **Ant Design** de acuerdo a la documentación, de manera inicial, tenemos que instalar esta dependencia en nuestro proyecto ***npm install antd --save***

Para comprobar que la librería se haya instalado correctamente y podamos usarla en nuestro proyecto, realiza el ejemplo que viene en la documentación oficial, importa el componente **DatePicker** desde antd y renderiza dentro del componente App, podrás observar una pantalla con el componente como se muestra a continuación.

Para modificar algunas de las configuraciones primarias de antd, es necesario envolver nuestra app con el provider de ant importa ***import { ConfigProvider } from 'antd';*** y envuelve tu aplicación con este provider pasandole las configuraciones necesarias para tu proyecto, por ejemplo para la propiedad *theme* modificamos el color primario.

```jsx
src > App.jsx > ...
1
2   import { DatePicker } from 'antd';
3   import { ConfigProvider } from 'antd';
4   import './App.css'
5
6   function App() {
7
8     return (
9       <ConfigProvider
10        theme={{
11          token: {
12            colorPrimary: '#7A83E1',
13          },
14        }}
15      >
16        <DatePicker />
17      </ConfigProvider>
18    )
19  }
20
21  export default App
```

Para no tener problemas con los márgenes, padding, entre otras propiedades a la hora de renderizar componentes en el componente principal App vamos a configurar los estilos de App.css

```css
1  html, body, #root {
2    margin: 0;
3    padding: 0;
4    height: 100%;
5    width: 100%;
6    overflow: hidden;
7  }
```

Ahora hablemos de layout, para este caso usaremos grid para posicionar los elementos teniendo en cuenta las propiedades responsive de los mismos componentes.

Vamos a crear una carpeta para componentes src/components y dentro la siguiente estructura ***Layout/LayoutComponent.jsx y .css*** de manera inicial este Layout tendrá dos espacios, uno a la derecha y otro a la izquierda, usando **Col y Row** de la librería de **Ant Design** para considerar el diseño responsivo usaremos las propiedades de acuerdo al tamaño de pantalla para cada columna, no te olvides de importar los componentes necesarios para este ejemplo.

src > components > Layout > LayoutComponent.jsx > ...

```jsx
import React from 'react';
import { Col, Row } from 'antd';
import './LayoutComponent.css';

const LayoutComponent = () => {
    return (
        <div className="layout-container">
            <Row>
                <Col xs={0} sm={0} md={4} lg={6} >
                    <div className="content-left">
                        <h1 className='title'>Contenido Izquierda</h1>
                    </div>
                </Col>
                <Col xs={24} sm={24} md={20} lg={18}>
                    <div className="content-right">
                        <h1 className='title'>Contenido Derecha</h1>
                    </div>
                </Col>
            </Row>
        </div>
    );
}

export default LayoutComponent;
```

Para lograr que todos los elementos estén centrados, es necesario aplicar estilos extras en las clases mencionadas en el componente, al final obtendrás una vista como esta usando este componente en el componente App

src > components > Layout > LayoutComponent.css > ...

```css
/* Contenedor principal del layout */
.layout-container {
    width: 100%;
    height: 100%;
    margin: 0;
    padding: 0;
}

.content-left{
    height: 100vh;
    background-color: #EBEFFF;
    display: flex;
    justify-content: center;
    align-items: center;
}
.content-right {
    height: 100vh;
    background-color: #AFB3FF;
    display: flex;
    justify-content: center;
    align-items: center;
}
.title{
    font-size: 20px;
    font-weight: 700;
    color: #656ED3;
}
```

Para hacer realmente el componente reutilizable, es necesario especificar por props las medidas de cada uno de los contenedores. Y pasarle los valores al componente donde lo vamos a utilizar.

src > components > Layout > LayoutComponent.jsx > ...

~/Documents/antd/antd-intro/src • Contiene elementos resaltados

```jsx
import { Col, Row } from 'antd';
import './LayoutComponent.css';

const LayoutComponent = ({ leftColSize, rightColSize, leftContent, rightContent }) => {
    return (
        <div className="layout-container">
            <Row>
                <Col xs={leftColSize.xs} sm={leftColSize.sm} md={leftColSize.md} lg={leftColSize.lg}>
                    <div className="content-left">
                        {leftContent}
                    </div>
                </Col>
                <Col xs={rightColSize.xs} sm={rightColSize.sm} md={rightColSize.md} lg={rightColSize.lg}>
                    <div className="content-right">
                        {rightContent}
                    </div>
                </Col>
            </Row>
        </div>
    );
}

export default LayoutComponent;
```

Crea dos componentes para renderizar en cada espacio del layout.

src > pages > Login > index.jsx > Login

```jsx
import React from 'react';
import LayoutComponent from '../../components/Layout/LayoutComponent';
import FormLogin from '../../components/FormLogin';
import ImageButton from '../../components/ImageButton';
const Login = () => {
    return (
        <LayoutComponent
            leftColSize={{ xs: 24, sm: 12, md: 8, lg: 6 }}
            rightColSize={{ xs: 24, sm: 12, md: 16, lg: 18 }}
            leftContent={<ImageButton />}
            rightContent={< FormLogin />}
        />
    );
}

export default Login;
```