

Entendido, quieres mantener los estilos separados para cada componente (`App.js` y `ModuloEstudiantes.jsx`) sin crear un archivo CSS global. Las irregularidades en la interfaz probablemente se deben a conflictos entre las variables CSS definidas en `App.css` y `ModuloEstudiantes.css`, o a la manipulación del elemento `body` en `ModuloEstudiantes.jsx`. Para resolver esto manteniendo los archivos CSS independientes, podemos aislar los estilos de `ModuloEstudiantes` usando **CSS Modules** o ajustando el código para evitar conflictos. La solución más efectiva será convertir `ModuloEstudiantes.css` en un módulo CSS y corregir la manipulación del DOM para que no afecte globalmente. A continuación, te detallo los pasos y proporciono el código actualizado.

Solución

1. Convertir `ModuloEstudiantes.css` en un módulo CSS

Usar CSS Modules asegura que los estilos de `ModuloEstudiantes` sean locales y no entren en conflicto con los de `App.css`. Esto evita que las variables CSS o clases como `.header` o `.main-content` se sobrescriban.

- Renombra `ModuloEstudiantes.css` a `ModuloEstudiantes.module.css`.
- Actualiza las reglas CSS para eliminar la dependencia de `body` y usar un contenedor local para el estado de colapso del sidebar.
- Modifica `ModuloEstudiantes.jsx` para importar los estilos como un objeto.

2. Corregir la manipulación del DOM

En lugar de agregar/remover la clase `sidebar-collapsed` en el elemento `body`, aplicaremos una clase local al contenedor del componente para manejar el colapso del sidebar.

3. Verificar importaciones

Asegúrate de que la importación de `locoAvatar` y la estructura de carpetas sean correctas para evitar errores de renderizado.

4. Mantener `App.css` sin cambios

`App.css` permanecerá intacto, ya que se aplica principalmente a `PaginaPrincipal`. Los estilos de `ModuloEstudiantes` estarán completamente aislados.

Código actualizado

A continuación, se presentan los archivos modificados para `ModuloEstudiantes.jsx` y `ModuloEstudiantes.module.css`. No se realizan cambios en `App.js` ni `App.css`, ya que deseas mantenerlos separados.



`ModuloEstudiantes.jsx`
javascript

Mostrar esquemas alineados



`ModuloEstudiantes.module.css`
css

Mostrar esquemas alineados

Explicación de los cambios

1. CSS Modules:

- `ModuloEstudiantes.css` se renombró a `ModuloEstudiantes.module.css` y se importó como `styles` en `ModuloEstudiantes.jsx`.
- Las clases se aplican usando `styles.nombreClase` (por ejemplo, `styles.container`, `styles.sidebar`). Esto asegura que los estilos sean locales y no entren en conflicto con `App.css`.
- Las variables CSS (`--primary-color`, `--secondary-color`, etc.) se eliminaron de `ModuloEstudiantes.module.css`, y se usaron valores directos (por ejemplo, `#00BCD4` en lugar de `var(--secondary-color)`) para evitar depender de las variables de `App.css`.

2 Manipulación del DOM:

- Se eliminó el `useEffect` que modificaba el elemento `body` para agregar/remover `sidebar-collapsed`.
- En su lugar, se aplica la clase `collapsed` al contenedor principal (`<div className={`${styles.container} ${sidebarCollapsed ? styles.collapsed : ''}`}>`).
- Los estilos de colapso del sidebar ahora se aplican usando el selector `.collapsed.sidebar` en lugar de `body.sidebar-collapsed`.

3 Nombres de clases:

- Algunas clases se renombraron (por ejemplo, `sidebar-header` a `sidebarHeader`, `sidebar-nav` a `sidebarNav`) para seguir la convención de camelCase en CSS Modules y evitar conflictos con guiones.

4 Sin cambios en `App.js` ni `App.css`:

- `App.js` y `App.css` permanecen intactos, ya que solo afectan a `PaginaPrincipal`. Los estilos de `ModuloEstudiantes` ahora están completamente aislados.

5 Verificación de la imagen:

- La importación de `locoAvatar` se mantuvo, pero asegúrate de que la ruta `.../../image/logo.png` sea correcta. Si la imagen está en la carpeta `public`, puedes usar `` para evitar problemas con la resolución de rutas.

⁶ Font Awesome:

- Asegúrate de que Font Awesome esté configurado en el proyecto. Por ejemplo, agrega en `public/index.html`:

```
html
```

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/
```

O instala e importa la librería en `index.js`:

```
javascript
```

```
import '@fortawesome/fontawesome-free/css/all.min.css';
```

Pasos adicionales

1 Verificar la estructura de carpetas:

- Confirma que `moduloEstudiantes.jsx` está en `src/pages/layout/`.
- Asegúrate de que `ModuloEstudiantes.module.css` esté en `src/styles/`.
- Verifica que `.../.../image/logo.png` exista. Si está en `public/image/`, usa la ruta `/image/logo.png` en el JSX.

2 Probar la ruta `/moduloEstudiantes`:

- Navega a `/moduloEstudiantes` en el navegador y verifica que el componente se renderice correctamente.
- Usa las herramientas de desarrollador del navegador para inspeccionar los estilos y confirmar que no hay conflictos con `App.css`.

3 Depurar errores:

- Revisa la consola del navegador para detectar errores relacionados con importaciones o renderizado.
- Si los íconos de Font Awesome no se muestran, verifica que la librería esté correctamente incluida.

4 Probar responsividad:

- Prueba en diferentes tamaños de pantalla para asegurarte de que el sidebar colapsado y los elementos de la tabla se vean correctamente.

5 Probar funcionalidad:

- Verifica que las acciones de la tabla (nuevo, modificar, eliminar) funcionen sin errores.
- Confirma que el colapso del sidebar no afecta otros elementos de la interfaz.

Notas finales

- Al usar CSS Modules, los estilos de `ModuloEstudiantes` están completamente aislados de `App.css`, eliminando conflictos de clases o variables CSS.
- Si observas irregularidades adicionales (por ejemplo, problemas específicos de diseño o comportamiento), proporciona más detalles (como capturas de pantalla o mensajes de error) para un análisis más profundo.
- Mantener los archivos CSS separados es ahora posible sin conflictos, ya que `ModuloEstudiantes.module.css` es específico para el componente `ModuloEstudiantes`.

Si necesitas ajustes adicionales o tienes más detalles sobre las irregularidades, házmelo saber.