

Módulo Ingeniería

Este componente orquesta la navegación secundaria del módulo de Ingeniería.

Obtiene los datos del usuario y enruta según sus permisos.

Diagrama de flujo de navegación y datos

Este diagrama muestra cómo fluye la información entre el componente, el hook y el router secundario.

Endpoints utilizados

Este componente no realiza llamadas directas.

Sin embargo, depende del hook **useTraerUsuario**, que invoca:

- **GET /api/usuario**: Recupera datos del usuario actual, incluyendo rutas disponibles.

GET /api/usuario

Descripción de la implementación

Este componente sigue tres pasos clave:

- Invocar el hook `useTraerUsuario` para obtener el objeto `usuario`.
- Buscar en `usuario.rutas` la ruta con `path === rutasIngenieria.path`.
- Mostrar `EnrutadorSecundario` solo si la ruta existe.

Código principal

tsx

```
import { rutasIngenieria } from "../../enrutadores/rutas/rutasIngenieria";
import { useTraerUsuario } from "../_custom-hooks";
import EnrutadorSecundario from "../../enrutadores/EnrutadorSecundario";

const ModuloIngenieria = () => {
  const usuario = useTraerUsuario();
  const ruta = usuario.rutas.find(
    r => r.path === rutasIngenieria.path
  );

  return (
    <>
      {ruta && <EnrutadorSecundario ruta={ruta} />}
    </>
  );
};

export default ModuloIngenieria;
```

Propiedades clave

Elemento	Tipo	Descripción	
usuario	Usuario	Objeto con datos y rutas del usuario.	
rutasIngenieria	{ path: string; ...}	Configuración de rutas del módulo Ingeniería.	
ruta	Ruta \	undefined	Ruta filtrada según el path.
EnrutadorSecundario	React.ComponentType	Componente que renderiza la ruta secundaria.	

⚠ Fallas en la lógica

- No maneja estado de **carga** (loading) antes de obtener usuario.
- No contempla **errores** al traer usuario (p. ej. token expirado).
- Si usuario o usuario.rutas es undefined, lanzaría excepción.
- No muestra **fallback** cuando no existe la ruta.

Recomendación

Agregar manejo de loading, errores y fallback UI.