Diagrama de Flujo de Navegación y Datos

Este diagrama muestra cómo fluyen la navegación y los datos entre los componentes principales y la API.

EstructuraDeCostos.tsx

Componente principal que carga y muestra la lista de estructuras de costos.

Utiliza hooks personalizados para configuración, usuario, descarga y errores.

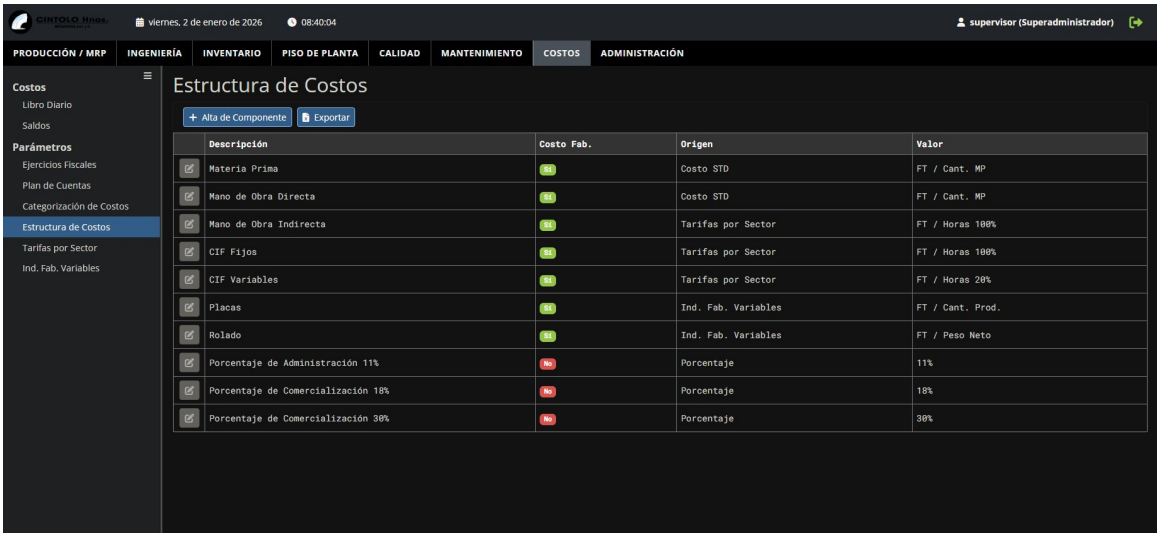

Figura 1 – Pantalla principal de Estructura de Costos con listado y acciones de alta y exportación.

- **Hooks usados**: `useState`, `useEffect`, `useTraerConfiguracion`, `useTraerUsuario`, `useDescargarExcel`, `useMostrarError`.
- **Componentes hijos**: `EstructuraDeCostosTabla`, `CrearCategoria` (modal).
- **Flujo**: monta → fetch datos → render tabla → acciones de UI.

tsx

```
useEffect(() => {
  hacerPeticionApi().then(respuesta => {
    config.setCargando(false);
    const cv = respuesta.datos?.estructurasDeCostos as typeof estructurasCostos;
    if (!cv) {
      mostrarError(respuesta);
      return;
    }
    setEstructurasCostos(cv);
  });
```

```
}, []);
```

Endpoints utilizados

- **GET** `/estructuras-costos` (obtención de lista).

GET /estructuras-costos

⚠ Fallas en la lógica

- No refresca la lista tras crear una nueva categoría.
- El botón **Editar** está siempre deshabilitado y sin handler.
- Se pasan 3 títulos a `generarEstilosDeCabezaDeTabla`, pero la tabla tiene 5 columnas.
- No maneja errores de red antes de ocultar el spinner.
- Falta cancelar la petición en el desmontaje del componente.

CrearCategoria.tsx

Modal funcional para crear una nueva categoría.

Controla el estado del campo **descripcion** y llama a la API al enviar el formulario.

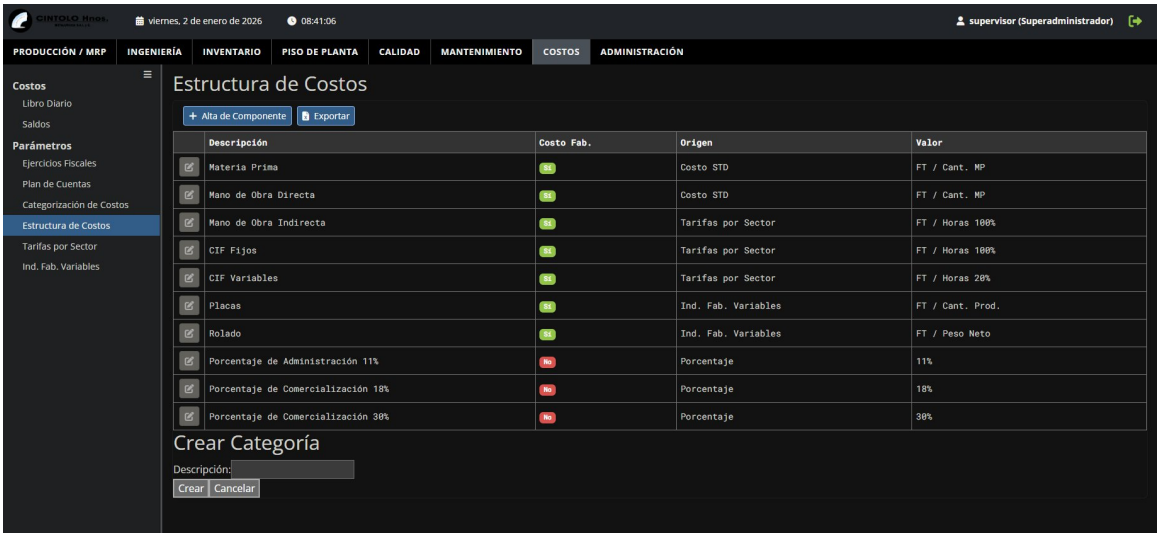

Figura 2 – Formulario de alta de una nueva Estructura de Costos con carga de Descripción.

- **Props**:

Prop	Tipo	Descripción
onClose	`() => void`	Cierra el modal de creación

- **Estado interno**: `descripcion: string`.
- **Hook usado**: `useMostrarError` para notificar fallos.

tsx

```
const handleSubmit = async (e: FormEvent<HTMLFormElement>) => {
  e.preventDefault();
  const response = await hacerPeticionApi({
    method: 'POST',
    url: '/categorias',
    body: { descripcion, estructuras: [] }
  });
  if (response.error) {
    mostrarError(response);
    return;
  }
  onClose();
}
```

Endpoints utilizados

- **POST** `/categorias` (crea una nueva categoría).

POST /categorias

⚠ Fallas en la lógica

- No muestra un **loading** al enviar el formulario.
- No valida duplicados o longitud mínima de descripción.
- No comunica al padre que recargue la lista tras creación.

EstructuraDeCostosTabla.tsx

Presenta los datos de `estructurasCostos` en una tabla.

Permite mostrar un badge según `esFabril` y un botón de edición según permisos.

- **Props**:

	Prop	Tipo	Descripción
estructurasCostos	`estructuraCostoType[] \	null`	Datos a mostrar en la tabla

- **Hooks usados**: `useTraerUsuario`.
- **Servicios**: `tienePermiso`, `generarEstilosDeCabezaDeTabla`.
- **Componente fallback**: `NoHayElementos` si no hay datos.

tsx

```
{!!estructurasCostos?.length
  ? estructurasCostos.map(e => (
      <tr key={e.id}>
        <td className='text-center-d'>
          {tienePermiso(usuario, e) && (
            <button className='btn btn-sm btn-secondary' disabled>
              <i className='fas fa-edit' />
            </button>
          )}
        </td>
        <td>{e.descripcion}</td>
        <td>
          {e.esFabril ? (
```

```
            <span className='badge bg-success'>Sí</span>
          ) : (
            <span className='badge bg-danger'>No</span>
          )}
        </td>
        <td>{e.origen}</td>
        <td>{e.valor}</td>
      </tr>
    ))
  : <NoHayElementos />}
```

Endpoints utilizados

Este componente no realiza llamadas directas a la API.

⚠ Fallas en la lógica

- **titulosDeTabla** (3 elementos) no coincide con las 5 columnas reales.
- El botón de **editar** siempre aparece deshabilitado; falta implementación.
- No ofrece funcionalidades de ordenación, filtrado o paginación.