

Diagrama de flujo de navegación y datos

Este diagrama muestra la navegación entre componentes y el flujo de datos desde la petición API hasta la representación en tabla.

IndirectosVariables.tsx

Este componente principal carga y muestra los costos indirectos variables. Utiliza hooks personalizados para controlar la configuración, la exportación y el acceso al historial.

The screenshot shows a dark-themed application interface. At the top, there's a navigation bar with tabs: PRODUCCIÓN / MRP, INGENIERÍA, INVENTARIO, PISO DE PLANTA, CALIDAD, MANTENIMIENTO, COSTOS (which is currently selected), and ADMINISTRACIÓN. On the far right, it says 'supervisor (Superadministrador)' with a gear icon. Below the navigation, on the left, is a sidebar with sections: Costos (Libro Diario, Saldos), Parámetros (Ejercicios Fiscales, Plan de Cuentas, Categorización de Costos, Estructura de Costos, Tarifas por Sector), and Ind. Fab. Variables (selected). The main content area is titled 'Costos Indirectos de Fabricación Variables'. It contains several tables: 'Rolado' (with rows for Virolos hasta 14 pulg., Virolos más de 14 pulg., Conos concéntricos, Conos excéntricos), 'Radiografías' (with rows for Valor por Placa), 'Partículas Magnéticas' (with rows for Valor por Kg Bruto, Valor por Hora), and 'Pesos Reales' (with rows for Rolado, Placas, Partículas Magnéticas). At the bottom, there's a section for 'Valores por Hora Impositivo'. Two buttons are visible: 'Exportar' and 'Historial'.

Figura 1 – Pantalla principal de Costos Indirectos de Fabricación Variables con listado, y acciones de exportación.

• Hooks utilizados

- useTraerConfiguracion: controla el estado de carga y error global.
- useMostrarError: despliega alertas de error según respuesta API.
- useDescargarExcel: gestiona la descarga en formato Excel.
- useAcceder: verifica permisos y navega al historial.

• Lógica principal

js

```
useEffect(() => {
  hacerPeticionApi().then(respuesta => {
    config.setCargando(false);
    const dr = respuesta.datos?.datosCostosIndirectos as costosIndFabrVarType;
    if (!dr) {
      mostrarError(respuesta);
      return;
    }
    setCifv(dr);
  });
}, []);
```

1. Realiza petición a API al montar.

2. Desactiva el spinner (`config.setCargando(false)`).

3. Valida si hay datos y actualiza el estado.

- **Endpoints utilizados**

- GET `window.location.pathname`

- Retorna `datos.datosCostosIndirectos`.

- GET `${window.location.pathname}/historial`

- Generado por `useAcceder`, para navegar al historial.

- **⚠ Posibles fallas en la lógica**

- Lanza error aun si simplemente no hay datos.

- El botón **Guardar Cambios** está siempre deshabilitado.

- No permite reintentos al no cambiar dependencias en `useEffect`.

- No cancela la petición al desmontar el componente.

IndirectosVariablesTabla.tsx

Este componente recibe un objeto con costos y los muestra en una tabla. Cada celda es un input de solo lectura con formato monetario.

- **Props**

	Prop	Tipo	Descripción
datosRolado	<code>`costosIndFabrVarType`</code>	<code>null`</code>	Datos de costos para el periodo fiscal

- **Lógica de renderizado**

- Encabezados muestran el periodo formateado.

- Filas fijas para procesos de rolado, radiografías y partículas.

- Inputs con clase `claseCtrlDer` para alinear texto a la derecha.

- **Servicios utilizados**

- `formatearPeriodoFiscal()`: convierte el periodo numérico en texto legible.

- **⚠ Posibles fallas en la lógica**

- Se esperaba un array (comentario de `.map`), pero `datosRolado` es un objeto.

- No hay manejo si `datosRolado` es `null`.

- Hardcode de filas complica futuros cambios en estructura de datos.

HistorialIndVar.tsx

Este componente muestra el historial de cambios de costos variables. Trae una lista de entradas y la pasa a su tabla.

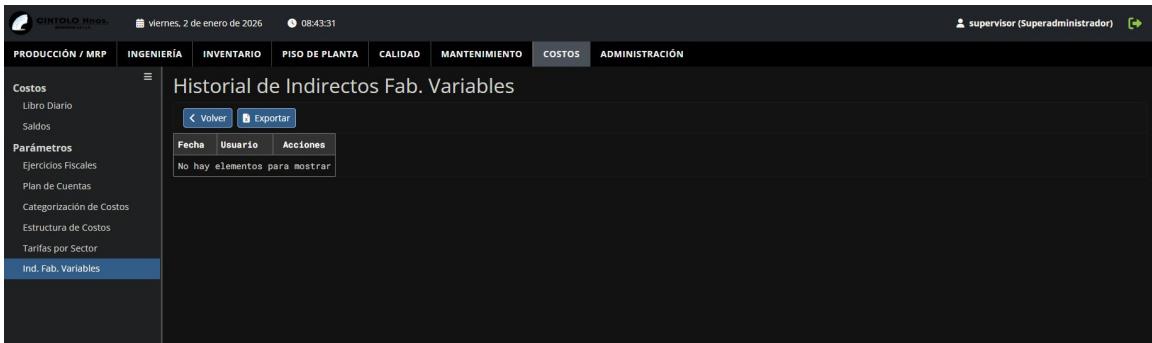


Figura 2 - Vista detallada del Historial de Indirectos Fab. Variables con listado y acción de exportación.

- **🛠 Hooks utilizados**
- `useTraerConfiguracion`: controla spinner y errores.
- `useMostrarError`: muestra alertas según respuesta API.
- `useDescargarExcel`: exporta la tabla completa.
- `useNavegar`: permite regresar a la vista principal.
- **Lógica principal**

js

```
useEffect(() => {
  hacerPeticionApi().then(respuesta => {
    config.setCargando(false);
    const hiv = respuesta.datos?.historialCostosIndirectosVariables as costosIndFabrVarType[];
    if (!hiv) {
      mostrarError(respuesta);
      return;
    }
    setHistorialIndirectosVariables(hiv);
  });
}, []);
```

1. Sigue el historial al montar.
2. Actualiza `historialIndirectosVariables` con la respuesta.
3. Despliega tabla o error según el resultado.

- **Endpoints utilizados**

- GET `window.location.pathname`
- Retorna `datos.historialCostosIndirectosVariables`.
- **⚠ Posibles fallas en la lógica**
- No distingue entre error y lista vacía.
- Botón **Consultar** en la tabla está deshabilitado sin explicación.
- No hay gestión de estados de carga secundarios.

HistorialIndVarTabla.tsx

Este componente recibe el array de historial y lo muestra en una tabla sencilla. Incluye fecha, usuario y acciones.

- **🛠 Props**

	Prop	Tipo	Descripción
historialIndirectosVariables	`costosIndFabrVarType[]`	null`	Lista de registros de historial

- **Lógica de renderizado**

- Genera estilos de encabezado con generarEstilosDeCabezaDeTabla().
- Itera sobre historialIndirectosVariables para crear filas.
- Utiliza formatearPeriodoFiscal() para la columna Fecha.
- Muestra NoHayElementos si la lista está vacía.
- **⚠ Posibles fallas en la lógica**
 - Inyecta <style> en cada render, lo que puede duplicar reglas CSS.
 - El botón **Consultar** carece de handler funcional.
 - No informa al usuario qué sucede al pulsar “Consultar”.