

Diagrama de flujo de navegación y datos

Retrabajos.tsx – Control principal

Componente raíz que:

- Recupera la lista de **retrabajos** desde la API.
- Gestiona filtros (letra, N° OP, N° retrabajo).
- Controla paginación, exportación y apertura de modales.

The screenshot shows a web-based application interface for managing work orders. At the top, there's a navigation bar with links like PRODUCCIÓN / MRP, INGENIERÍA, INVENTARIO, PISO DE PLANTA, CALIDAD (highlighted in blue), MANTENIMIENTO, COSTOS, and ADMINISTRACIÓN. On the left, a sidebar menu is open under the 'Retrabajos' section, listing various quality-related modules such as Control de Calidad, Agrupaciones de Calidad, Tipos de Fabricación, Familias de Fabricación, Calidades de Fabricación, Normas de Fabricación, Especificaciones de Fabricación, Maestros de Materiales, Colores para Coladas, Menús de Coladas, Receptores Técnicos, ABM de Coladas, Gestión de Prouebas, Retrabajos/Req. Esp., and Retrabajos. The main content area displays a table titled 'Retrabajos' with columns: Definir, Detalle PDF, Comentarios, Letra, OP, N° Retrab., Etapa/Subetapa OP, Código Motivo, and Sub-Código. There are 7 registros listed. Each row includes a 'Descartar' button and icons for Ver, Exportar PDF, Imprimir Tabla, and Imprimir PDF. The table has a header row with filters for Letra, Nro. OP, and Nro. Retrabajo. At the bottom of the table, there are navigation arrows for pagination.

Figura 1 – Pantalla principal de Retrabajos con listado, filtros y acciones de retener, exportación e impresión.

js

```
// Carga inicial de datos
useEffect(() => {
  hacerPeticionApi().then(respuesta => {
    config.setCargando(false);
    const rs = respuesta.datos?.retrabajos;
    if (!rs) { mostrarError(respuesta); return; }
    setRetrabajos(rs);
  });
}, []);
```

Endpoints

Método	Ruta	Descripción
GET	/api/retrabajos	Obtiene lista completa de retrabajos.

Fallas en la lógica

- El estado `mostrarAbm` está comentado y nunca se inicializa, por lo que el modal ABM no aparece:

```
js
// const [mostrarAbm, setMostrarAbm] = useState(false);

• Los inputs numéricos convierten cadena vacía en NaN, dejando el filtro inestable.
• Botones de impresión Excel/PDF usan la misma etiqueta "Imprimir Tabla".
```

RetrabajosPdf.tsx – Vista e impresión en PDF

Muestra en modal los datos detallados de un retrabajo y ofrece:

- Descarga de PDF desde HTML.
- Impresión directa.
- Cierre con tecla Esc.

```
js
<button onClick={() => imprimirEnPdf.imprimirEnPdfDesdeHtml(contenidoRef.current, 0.24)}>
  Descargar PDF
</button>
```

Endpoints

- No realiza llamadas propias a la API. Recibe retrabajo por props.

Fallas en la lógica

- No se identifican errores evidentes; asume datos completos en props.

RetrabajosTabla.tsx – Tabla de datos

Renderiza una tabla con:

- Paginado y estilos generados dinámicamente.
- Acciones por fila: Ver PDF, Definir, Comentarios, ABM.
- Despliegue de procesos asociados a cada retrabajo.

```
js
const titulosDeTabla = ['Letra', 'OP'];
<style>{generarEstilosDeCabezaDeTabla(titulosDeTabla)}</style>
```

Endpoints

- No realiza llamadas directas; usa retrabajos de padre.

Fallas en la lógica

- Si procesos es undefined, el cálculo de rowspan puede ser incorrecto.
- La lógica de conteo de partes (reduce) aparece duplicada en varios lugares.

RetrabajosDefinir.tsx – Definir recuperación y descarte

Modal para asignar cuántas piezas se:

- Recuperan.
- Envían a transformación.
- Se descartan (chatarra).

Figura 2 - Vista del formulario para definir Retrabajo con Cantidad Retenida, Recuperada, a Transformación y a chatarra.

Envía un **PATCH** con las cantidades.

js

```
// Actualización de cantidades
const respuesta = await hacerPeticionApi({
  method: 'PATCH',
  body: {
    cantidadATransformacion,
    cantidadRecuperada,
    esDescarteDirecto,
    retrabajoId: retrabajo.id,
  }
});
```

Endpoints

Método	Ruta	Descripción
PATCH	/api/retrabajos	Actualiza cantidades de recuperación y descarte.

Fallas en la lógica

- `esValido` está definido siempre `true`; no valida límites de suma.
- La validación de rango queda sólo en el `onChange` de inputs.

RetrabajosAbm.tsx – Crear nuevo retrabajo

Modal de **ABM** que:

- Carga datos maestros: OP en marcha, etapas, motivos, submotivos.
- Arma el objeto retrabajo inicial.
- Envía **POST** para guardarla.
- Al crear, abre el PDF del nuevo retrabajo.

Figura 3 – Formulario de crear un nuevo Retrabajo con carga de Código, cantidad de procesos, cantidad de retenidos y descripción.

js

```
// Carga de datos para ABM
useEffect(() => {
  hacerPeticionApi({ queryParams: { esAbm: true } })
    .then(respuesta => { /* setEtapasSubetapas, etc. */ });
}, []);
```

Endpoints

Método	Ruta	Descripción
GET	/api/retrabajos?esAbm=true	Trae OP, etapas, motivos y submotivos para el formulario ABM.
POST	/api/retrabajos	Crea un nuevo retrabajo con los datos del formulario.

Fallas en la lógica

- Parámetro `queryParams.esAbm` no documentado en ruta.
- No reinicia estados locales en caso de error tras llamada GET.

RetrabajosComentarios.tsx – Comentarios finales

Modal para mostrar/editar:

- **Comentarios Inicio** (siempre de solo lectura).
- **Comentarios Fin** (editable si el usuario tiene permiso).

Envía un **PATCH** con el comentario final.

js

```
// Guardar comentarios fin
const respuesta = await hacerPeticionApi({
  method: 'PATCH',
  body: { comentariosFin, retrabajoId: retrabajo.id }
});
```

Endpoints

Método	Ruta	Descripción
PATCH	/api/retrabajos	Actualiza el comentario final del retrabajo.

Fallas en la lógica

- `esValido` siempre `true`; habría que validar longitud mínima.
- Límite de 1500 caracteres en `textarea` sin feedback visual.

Conclusión

La suite de componentes cubre todo el flujo CRUD de “retrabajos” con modales especializados. Se recomienda corregir los estados comentados y mejorar las validaciones (`esValido`) en los modales de definición y comentarios.