

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.

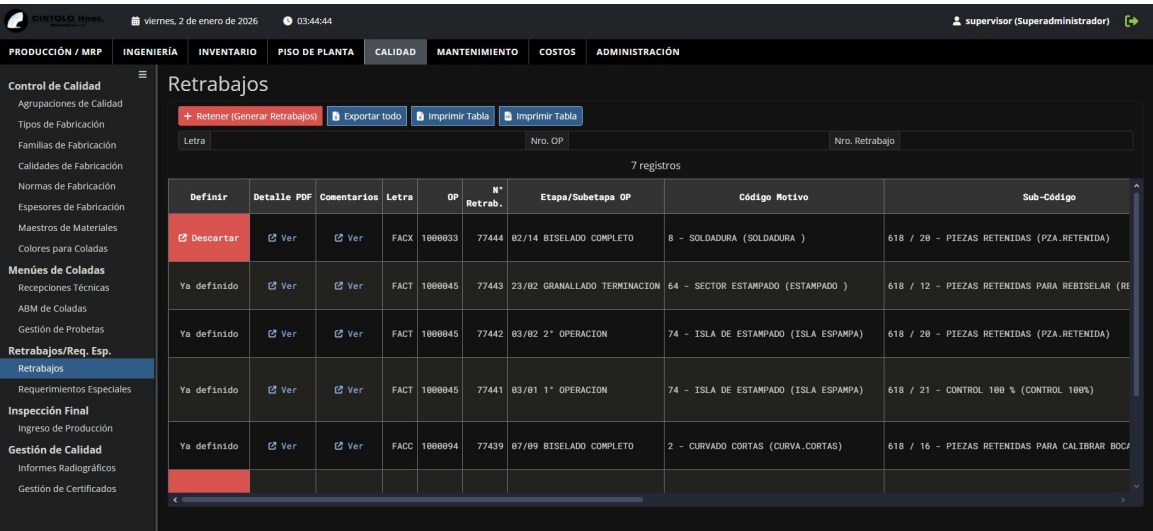


Figura 1 – Pantalla principal de Retrabajos con listado, filtros y acciones de retener, exportación e impresión.

js

```
// Carga inicial de datos
useEffect(() => {
  hacerPeticiónApi().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 re**tr**abajo 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 re**tr**abajos 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).

DEFINIR RETRABAJO 77444 de la OP FACX - 1000033

*En esta instancia solo se admite declarar descarte de todas las piezas. Para declarar recuperos, hay que completar primero el esquema de partes diarios.

Cantidad Retenida: 5

Cantidad Recuperada: 0

Cantidad a Transformación: 5

Cantidad a Chatarra: 0

Guardar Cambios

Cancelar

| Definir | Detalle PDF | C |
|---|---|---|
| <input checked="" type="checkbox"/> Descartar | <input checked="" type="checkbox"/> Ver | |
| Ya definido | <input checked="" type="checkbox"/> Ver | |
| Ya definido | <input checked="" type="checkbox"/> Ver | |
| Ya definido | <input checked="" type="checkbox"/> Ver | |
| Ya definido | <input checked="" type="checkbox"/> Ver | |
| Ya definido | <input checked="" type="checkbox"/> Ver | |
| Ya definido | <input checked="" type="checkbox"/> Ver | |
| Ya definido | <input checked="" type="checkbox"/> Ver | |

| Nro. Retrabajo | Sub-Código |
|--|------------|
| 618 / 20 - PIEZAS RETENIDAS (PZA.RETENIDA) | |
| DO) 618 / 12 - PIEZAS RETENIDAS PARA REBISLAR (RE | |
| SPAMPA) 618 / 20 - PIEZAS RETENIDAS (PZA.RETENIDA) | |
| SPAMPA) 618 / 21 - CONTROL 100 % (CONTROL 100%) | |
| 618 / 16 - PIEZAS RETENIDAS PARA CALIBRAR BOCA | |

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 guardarlo.
- 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(() => {
  hacerPeticiónApi({ 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.