

# Diagrama de flujo de navegación y datos

Este diagrama muestra el flujo principal entre componentes y las peticiones HTTP para gestionar requerimientos especiales.

## RequerimientosEspeciales.tsx

Este componente principal carga y muestra la lista de requerimientos especiales. Permite generar nuevos, exportar e imprimir.

### Funciones clave

- Carga inicial de datos al montar el componente
- Aplicación de filtros por letra, OP y número de requerimiento
- Control de paginación y operaciones globales

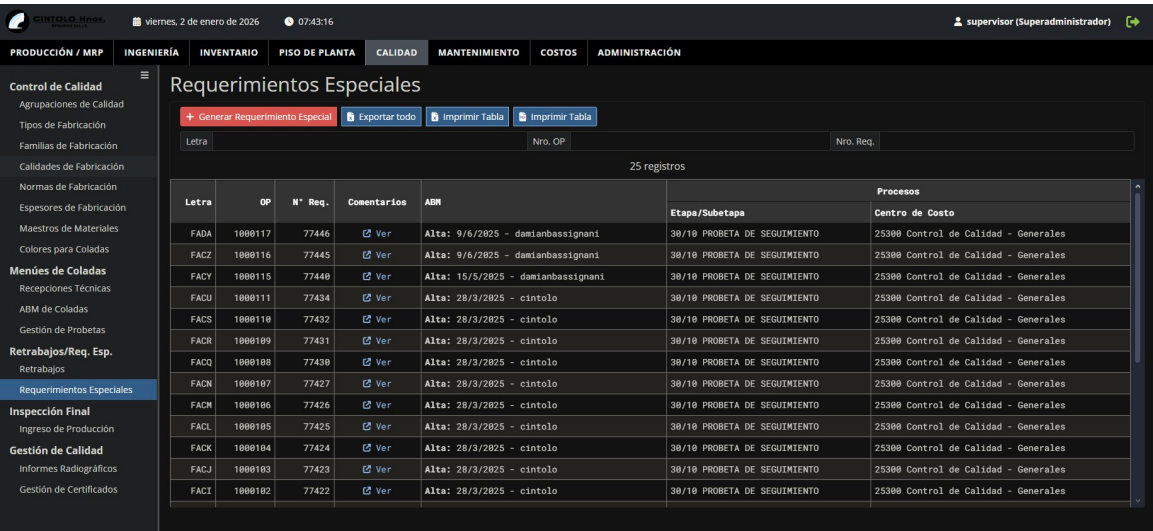


Figura 1 – Pantalla principal de Requerimientos Especiales con listado, filtros y acciones de generación, exportación e impresión.

### Endpoints utilizados

Componente	Método	Endpoint inferido	Descripción
RequerimientosEspeciales.tsx	GET	/api/requerimientos-especiales	Obtiene todos los requerimientos

### Peticiones internas

```
js
// Al montar:
hacerPeticionApi()
// Para Excel/PDF e impresión no invoca API adicional
```

### Fallas en la lógica

- Falta declarar mostrarAbm y su setter. Provoca error al abrir modal.

- `parseInt` en filtros puede generar NaN si el campo está vacío.
- `useImprimirTabla(tablaRef?.current)` recibe null inicialmente.

#### Error Crítico

La variable `mostrarAbm` no está declarada, provoca fallo al abrir modal.

## RequerimientosEspecialesTabla.tsx

Este componente presenta la tabla de datos y gestiona la apertura del modal de comentarios.

### Descripción

- Recibe lista paginada y la referencia de la tabla
- Despliega filas con procesos secundarios
- Abre modal de comentarios al hacer clic

### Lógica de renderizado

- Inserta estilos de cabecera con `generarEstilosDeCabezaDeTabla`
- Maneja «No hay elementos» si no hay datos
- `RowSpan` calcula mal la cantidad de filas cuando no hay procesos

js

```
// Cálculo incorrecto de rowSpan
rowSpan={(re.procesos?.length || 1) + 1}
```

### Fallas en la lógica

- `rowSpan` suma 1 extra, rompe estructura si no hay procesos.
- Uso de `requerimientos?.length` y `!!requerimientos.length` poco consistente.

## RequerimientosEspecialesComentarios.tsx

Modal para visualizar y editar comentarios de un requerimiento especial.

### Descripción

- Muestra comentarios de inicio y fin
- Habilita edición sólo si el usuario tiene permiso
- Envía PATCH para actualizar comentarios

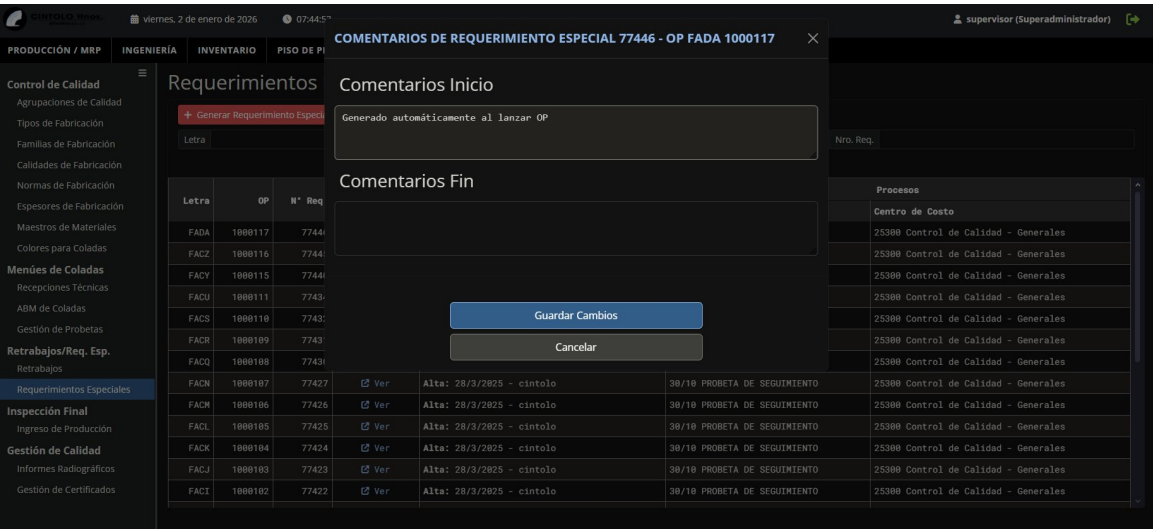


Figura 2 – Vista de detalle de Comentarios de un Requerimiento Especial.

Endpoints utilizados

Componente	Método	Endpoint inferido	Descripción
RequerimientosEspecialesComentarios	PATCH	/api/requerimientos-especiales	Actualiza comentarios de fin

Peticiones internas

```
js
// Al guardar:
hacerPeticiónApi({
  method: 'PATCH',
  body: { comentariosFin, reqEspecialId }
})
```

Fallas en la lógica

- esValido siempre es true; falta validación real de campos.

RequerimientosEspecialesAbm.tsx

➡ Modal para crear un nuevo requerimiento especial con sus procesos.

Descripción

- Carga opciones de etapas/subetapas y OP en marcha
- Permite configurar cantidad de procesos y piezas retenidas
- Envía POST para crear el requerimiento

Figura 3 – Formulario de generación de un nuevo Requerimiento Especial con carga de Orden de Producción, Cantidad de Procesos, Cantidad de Piezas y comentarios.

Endpoints utilizados

Componente	Método	Endpoint inferido	Descripción
RequerimientosEspecialesAbm.tsx	GET	/api/requerimientos-especiales?esAbm=true	Obtiene datos para ABM
POST	/api/requerimientos-especiales	Crea nuevo requerimiento especial	

Peticiones internas

js

```
// Al iniciar modal:
hacerPeticiónApi({ queryParams: { esAbm: true } })
// Al aceptar:
hacerPeticiónApi({
  method: 'POST',
  body: { requerimientoEspecial }
})
```

Fallas en la lógica

- Parámetro anular en aceptar no se utiliza.
- No muestra estado de carga mientras espera datos ABM.
- No valida comentarios de inicio ni otros campos opcionales.

Conclusión

Este conjunto de componentes ofrece un CRUD básico sobre requerimientos especiales.

Se recomienda corregir las variables no declaradas y ajustar validaciones para evitar errores de renderizado.