

## Navegación y Flujo de Datos de Informes Radiográficos

A continuación, se describe la arquitectura y flujo principal de navegación y datos entre los componentes de la funcionalidad de **Informes Radiográficos**. Este diagrama ayuda a visualizar la interacción entre componentes, el flujo de datos y los puntos principales de entrada/salida de información.

## Componentes y Documentación Detallada

### InformesRadiograficos.tsx

Este componente es la **pantalla principal** de los Informes Radiográficos. Permite visualizar, filtrar, exportar, imprimir y agregar nuevos informes.

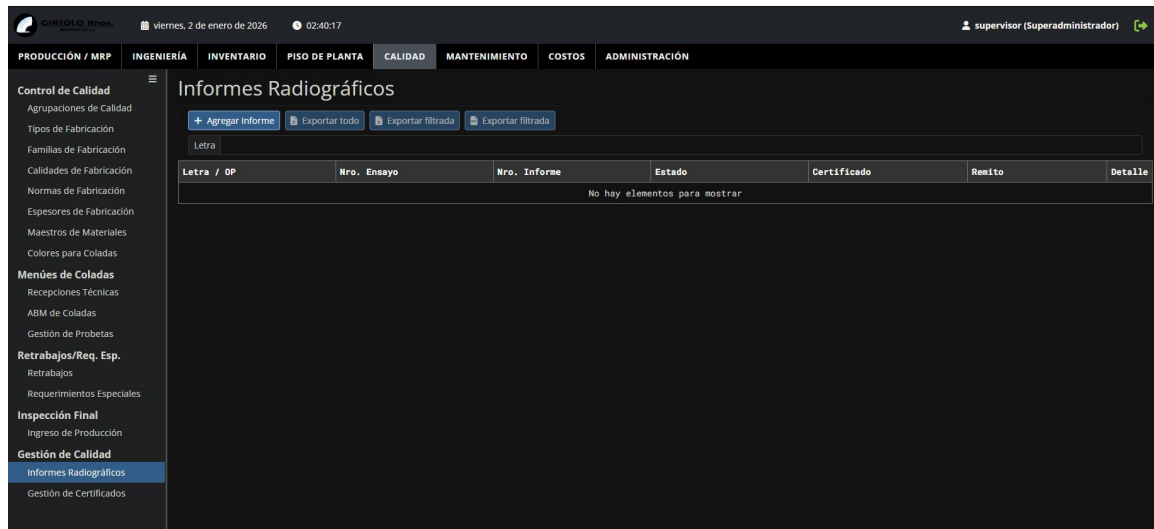


Figura 1 – Pantalla principal de Informes Radiográficos con listado, filtros y acciones de agregar, exportación e impresión.

### Principales responsabilidades

- Obtener datos iniciales de informes radiográficos y órdenes de producción con un **endpoint GET**.
- Mostrar un listado filtrable y exportable.
- Permitir agregar informes (si el usuario tiene nivel adecuado).
- Controlar el estado de carga y mostrar errores.

### Endpoints utilizados

#### 1. Obtener datos iniciales (informes y OPs)

### Lógica relevante

- **Filtrado:** Filtra por letra OP usando un input.
- **Exportar:** Permite exportar todos los informes o sólo los filtrados (Excel/PDF).

- **Agregar informe:** Abre modal y actualiza datos al crear uno nuevo.

### Observaciones sobre fallas

- Se accede a `informesRadiograficos` en el hook `useMemo`, pero en el código actual la variable parece comentada (`// const [informesRadiograficos, setInformesRadiograficos]`). Si está realmente comentada, el filtrado nunca funcionaría.
- El método de impresión/descarga recibe parámetros que pueden ser nulos si la tabla aún no está montada.
- No hay manejo de errores ante la ausencia de datos en el render principal (solo en la carga inicial).

### InformesRadiograficosTabla.tsx

Tabla principal para mostrar la lista de informes radiográficos. Permite acceder al detalle de cada uno.

### Principales responsabilidades

- Mostrar los informes recibidos en una tabla clara y responsive.
- Acceso al detalle de informe para usuarios autorizados.
- Integración con el componente de estado de placa.

### Endpoints utilizados

- **No realiza peticiones directamente.** Recibe datos procesados del padre.

### Observaciones sobre fallas

- El acceso a la navegación depende del hook `useAcceder`, pero si no hay permisos, el botón de detalle simplemente no aparece. No hay feedback visual.
- El orden de los informes se realiza cada render (`sort`), lo que podría afectar performance si la lista es grande.

### VistaEstadoInformesRX.tsx

Componente pequeño encargado de mostrar el estado de cada informe/placa mediante un badge de color.

### Principales responsabilidades

- Visualizar el estado de una placa de forma clara y visual.

### Observaciones sobre fallas

- Si se recibe un valor estado fuera de los valores esperados, no retorna nada. Podría ser útil un valor por defecto.

## InformesRadiograficosAgregar.tsx

Modal para agregar un nuevo informe radiográfico.

### Principales responsabilidades

- Permite al usuario ingresar todos los datos de un nuevo informe.
- Valida campos obligatorios.
- Envía los datos al backend utilizando un **endpoint POST**.
- Actualiza la lista de informes global tras agregar el nuevo registro.

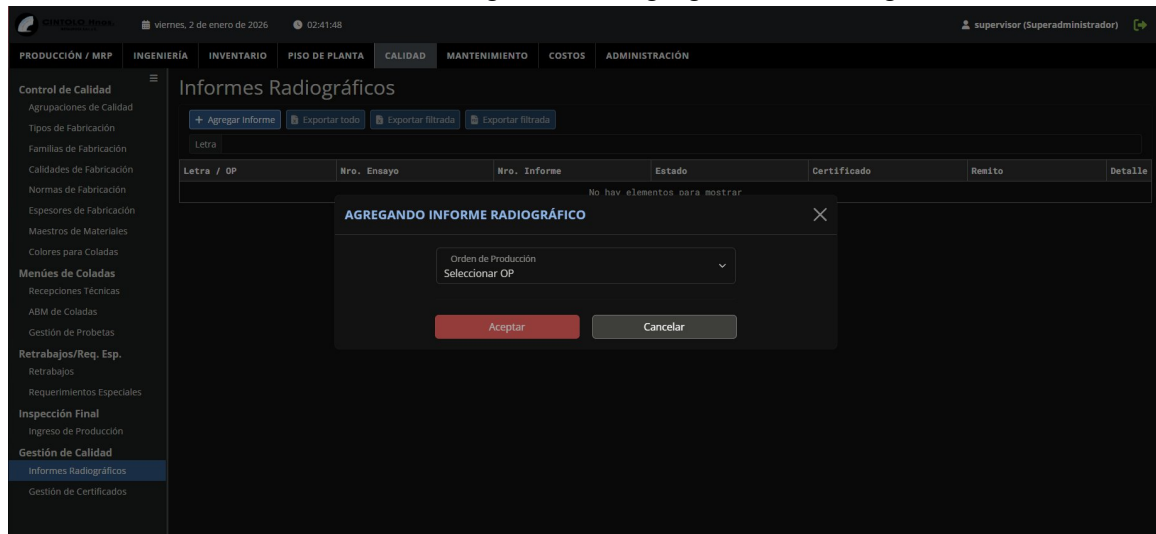


Figura 2 – Formulario de agregación de un nuevo Informe Radiográfico con carga de Orden de Producción.

### Endpoints utilizados

#### 2. Agregar nuevo informe radiográfico

### Observaciones sobre fallas

- El modal se cierra en caso de error, sin mostrar el mensaje específico al usuario.
- No está validando todos los campos importantes (solo algunos).
- Podría permitir guardar informes incompletos si el backend no valida.

## InformeRadiografico.tsx

Vista de detalle de un informe radiográfico, muestra todos los datos y sus uniones asociadas.

### Principales responsabilidades

- Muestra todos los datos de un informe radiográfico seleccionado.
- Permite agregar nuevas uniones (si el usuario puede).
- Muestra una tabla con las uniones.
- Trae los datos del informe y sus uniones mediante **endpoint GET**.

## Endpoints utilizados

### 3. Obtener detalle de informe radiográfico + uniones

#### Observaciones sobre fallas

- Hay campos duplicados o nombres de campos que no corresponden (ejemplo: "Usuario carga" muestra ensayo).
- La condición para mostrar el bloque de transformación OP está invertida (`!informeRadiografico.transfOp`).
- No permite editar aunque se habilite el modo edición (inputs deshabilitados siempre).

## InformeRadiograficoAgregarUnion.tsx

Modal para agregar una nueva unión a un informe radiográfico.

#### Principales responsabilidades

- Permite seleccionar el grupo, secuencia y estado de la nueva unión.
- Valida el campo obligatorio principal (longitud `informeRxGrupo`).
- Envía los datos al backend por **endpoint POST**.
- Actualiza los datos del informe y la lista de informes tras agregar.

## Endpoints utilizados

### 4. Agregar nueva unión a informe radiográfico

#### Observaciones sobre fallas

- El campo estado valida sólo longitud 1, pero no restringe valores válidos.
- No hay feedback de error al usuario si la creación de la unión falla.
- El botón Editar en la tabla de uniones está siempre deshabilitado, impidiendo edición real.

## InformeRadiograficoTabla.tsx

Tabla de detalle de las uniones de un informe radiográfico.

#### Principales responsabilidades

- Muestra todas las uniones asociadas a un informe en una tabla detallada.
- Permite (en teoría) editar una unión si el usuario tiene permisos, pero el botón está deshabilitado.

#### Observaciones sobre fallas

- Botón de edición de unión está **siempre deshabilitado**, por lo que nunca se puede editar una unión.

- Faltan validaciones de longitud de los arreglos antes de renderizar.
- El campo `titulosDeTabla` es un arreglo con un valor vacío, posiblemente un error.

## Resumen de Componentes vs Endpoints

Componente	Endpoint Principal	Método
InformesRadiograficos	/informes-radiograficos	GET
InformesRadiograficosAgregar	/informes-radiograficos	POST
InformeRadiografico	/informes-radiograficos/:id	GET
InformeRadiograficoAgregarUnion	/informes-radiograficos/:id/uniones	POST
InformesRadiograficosTabla, VistaEstadoInformesRX, InformeRadiograficoTabla	Sin endpoint directo, solo visualización	-

### Puntos críticos detectados

Se detectaron validaciones incompletas, feedback insuficiente al usuario en errores y botones de edición deshabilitados. Mejorar UX y robustez del flujo.

## Recomendaciones Finales

- Validar exhaustivamente los campos antes de enviar datos al backend.
- Proveer feedback visual al usuario ante errores de red o validaciones.
- Revisar lógica de edición y botonería deshabilitada en la gestión de uniones.
- Homogeneizar nombres de campos y mejorar la correspondencia label-valor en las vistas de detalle.

¿Deseas un análisis de código más detallado de algún componente específico?