

# Documentación del Módulo de Calidad

Este documento describe la estructura, componentes y flujo de datos del módulo Calidad. Incluye diagramas de navegación, detalle de props, funcionalidades y análisis de posibles fallas.

## Diagrama de flujo de navegación y datos

Este diagrama muestra cómo el usuario accede al módulo de Calidad y cómo fluyen los datos entre componentes.

## Componente: ColadaCertificados

ColadaCertificados permite visualizar, adjuntar y descargar PDFs de certificados de usina y planillas dimensionales. Soporta modos de edición y vista previa ampliada.

### Props

Prop	Tipo	Descripción	
cargandoDimensional	boolean	Muestra spinner durante carga de planilla dimensional	
cargandoUsina	boolean	Muestra spinner durante carga de certificado de usina	
modoEdicion	boolean	Habilita campos y botones de edición	
repcionTecnica	repcionTecnicaType	Objeto con datos y archivos de la recepción técnica	
setCambioDimensional	Dispatch<SetStateAction<boolean>>	Marca que el archivo dimensional fue modificado	
setCambioUsina	Dispatch<SetStateAction<boolean>>	Marca que el archivo de usina fue modificado	
setRepcionTecnica	Dispatch<SetStateAction<repcionTecnicaType \ null>>		Actualiza el estado de recepcionTecnica

## Flujo de datos y estado

- Se crean URLs con URL.createObjectURL para previsualizar PDFs.
- Los estados numeroDePaginaUsina y numeroDePaginaDimensional manejan la página mostrada.
- Los flags verCertificado y verPlanilla controlan el modal de vista ampliada.

## Funcionalidades principales

- Previsualiza la primera página de los PDFs.
- Permite adjuntar o cambiar archivos en modo edición.
- Descarga el PDF usando el hook useDescargarPdf.

## Endpoints utilizados

No hay llamadas a endpoints directas en el componente. El hook useDescargarPdf abstrae la petición:

js

```
// Ejemplo de uso interno  
descargarPdf(evento, 'tipo=usina')  
// Equivale a: GET /api/calidad/pdf?tipo=usina  
  
descargarPdf(evento, 'tipo=dimensional')  
// Equivale a: GET /api/calidad/pdf?tipo=dimensional
```

## Puntos críticos y posibles fallas

- No se revocan las URLs creadas con URL.createObjectURL, lo que puede generar fugas de memoria.
- Sólo se muestra la primera página del PDF; falta control de paginación.
- El atributo readOnly en <input type="file"> no impide la selección de archivos (debería usarse disabled).

### Possible fuga de memoria

Se crean URL con URL.createObjectURL sin revocarlas con URL.revokeObjectURL.

## Componente: ColadaDetalle

Muestra datos del material, proveedor, colada, agrupación de calidad y color. Permite editar algunos campos en modo edición.

### Props

Prop	Tipo	Descripción	
agrupacionesCalidas	agrupacionCalidadType[]	null	Lista de agrupaciones de

			calidad disponibles
modoEdicion	boolean	Indica si el formulario está en modo edición	
receptionTecnica	receptionTecnicaType	Objeto con datos de la recepción técnica	
setReceptionTecnica	Dispatch<SetStateAction<receptionTecnicaType \>	null>>	Función para actualizar el estado de receptionTecnica

## Funcionalidades principales

- Despliega información de artículo y proveedor en campos de solo lectura.
- Permite cambiar articuloAgrupacionCalidadId y colorId mediante <select>.
- Ordena y muestra opciones de agrupaciones y colores sugeridos.

## Endpoints utilizados

Este componente no hace llamadas a endpoints. Todos los datos llegan vía props desde el componente padre.

## Puntos críticos y posibles fallas

- El uso de useRef para capturar colorId fija el valor inicial y no refleja cambios posteriores.
- El campo colada no es editable: el onChange está comentado, posiblemente por error.

### Color estático

El uso de useRef impide que cambie colorId al actualizar receptionTecnica.

## Componente: MóduloCalidad

Módulo principal responsable de enrutar el submódulo Calidad según las rutas asignadas al usuario.

### Descripción

- Extrae el usuario con sus rutas usando el hook useTraerUsuario.
- Busca la ruta que coincide con rutasCalidad.path.
- Si la ruta existe, renderiza <EnrutadorSecundario ruta={ruta} />.

## Flujo de enrutamiento

- `useTraerUsuario` → obtiene datos de usuario y rutas.
- Filtra rutas por `rutasCalidad.path`.
- Renderiza el enrutador secundario sólo si la ruta está permitida.

## Endpoints utilizados

No realiza llamadas directas. El hook `useTraerUsuario` gestiona internamente:

- GET /api/usuario para obtener datos de sesión y rutas.

## Puntos críticos y posibles fallas

- No ofrece un mensaje o redirección alternativa si la ruta no existe.
- Depende de que `usuario.rutas` esté correctamente poblado por el backend.