

## Diagrama de flujo de navegación y datos

Este diagrama muestra cómo interactúan los componentes de UI con los endpoints de la API. Resume la navegación entre vistas y el flujo de datos.

## Endpoints por Componente

Esta sección lista los endpoints que consume cada componente. Cada ruta sigue el prefijo base `/inventario/...`

- **ProductosFabricados.tsx**
- GET `/inventario/stockfabricados`
- **ProductosFabricadosAgregar.tsx**
- POST `/inventario/stockfabricados`
- **DetallePF.tsx**
- GET `/inventario/stockfabricados/detalle?id={id}`
- PATCH `/inventario/stockfabricados/detalle?id={id}`
- **DetallePFDescripcion.tsx** y **DetallePFPoliticaStock.tsx**
- No consumen endpoints directamente; reciben datos de su componente padre.

### DetallePFPoliticaStock.tsx

Componente que muestra la **política de stock**, **datos contables**, **costos** y **almacenamiento** de un producto fabricado. Se organiza en varias tarjetas de Bootstrap.

### Props

- `articulosMedida`: `unidadDeMedidaType[]`
- `cuentasCompra`: `cuentaType[]`
- `cuentasInventario`: `cuentaType[]`
- `cuentasVentas`: `cuentaType[]`
- `detallePf`: `productoFabricadoType`
- `estructuras`: `estructuraCostosType[]`
- `opcionesIva`: `opcionIvaType[]`
- `tiposDeEmbalaje`: `embalajeType[]`

### Funcionalidad

- Despliega **Stock Mínimo** y **Stock Crítico** en modo lectura.
- Selecciona **IVA** y cuentas contables (inventario, ventas, compras).
- Elige **estructura de costos** y **versión de ficha técnica**.
- Muestra tabla de **costos finales** con total.
- Presenta **dimensiones**, **embalaje** y **peso** con sus unidades.

### Endpoints

No realiza peticiones propias; todos los datos provienen de sus props.

## Posibles fallas en la lógica

- Los **onChange** de los `<select>` solo registran en consola sin actualizar el estado ni enviar cambios.
- La pestaña **Costo Real** no está implementada.
- Se accede a `detallePf.fichasTecnicas[0]` en el pie de tabla sin validar que exista.

js

```
<select
  className='form-select form-select-sm'
  value={detallePf.ivaId}
  onChange={e => console.log(e.target.value)} // Falta handler real
>
```

## DetallePFDescripcion.tsx

Despliega información básica del producto fabricado en modo solo lectura, con alertas si está dado de baja.

## Props

- `articulosFamilias`: `familiaPfType[]`
- `articulosMedida`: `unidadDeMedidaType[]`
- `articulosTipos`: `tipoDeArticuloType[]`
- `detallePf`: `productoFabricadoType`

## Funcionalidad

- Muestra **Código, Tipo, Descripción, Unidad de Medida y Familia**.
- Indica con un banner si el producto está dado de baja.

## Endpoints

No realiza llamadas; depende de los datos precargados por el padre.

## Posibles fallas en la lógica

- Los `<select>` capturan el evento con `console.log` sin reflejar cambios reales.

js

```
<select
  className='form-select form-select-sm'
  value={detallePf.tipoId}
  onChange={e => console.log(e.target.value)} // Handler incompleto
>
```

## DetallePF.tsx

Coordina la vista de detalle, incluye los subcomponentes de descripción y política de stock, y gestiona navegación y altas/bajas.

## Hooks y estado

- useState para cuentas, estructuras, familias, tipos, IVA, embalaje y datos del producto.
- useTraerConfiguracion controla el spinner de carga.
- useMostrarError gestiona la presentación de errores.
- useNavegar navega entre rutas.
- useTraerUsuario obtiene el perfil de usuario.
- useEffect carga datos iniciales con hacerPeticiónApi.
- editarProductoFabricado envía PATCH para dar de baja o reactivar.

## Funcionalidad

- Carga datos y maneja estado de carga.
- Botones para volver al listado y ver movimientos de stock.
- Renderiza DetallePFDescripcion y DetallePFPoliticaStock.
- AceptarCancelarBtn para eliminar o reactivar.

Figura 1 – Vista de detalle de un Producto Fabricado, mostrando información descriptiva, política de stock y acciones disponibles.

## Endpoints

- GET /inventario/stockfabricados/detalle?id={id}
- PATCH /inventario/stockfabricados/detalle?id={id}

## Posibles fallas en la lógica

- El botón **Aceptar** está permanentemente deshabilitado (disabled={true}), impidiendo guardar cambios.
- No se captura ni maneja un posible error en la petición PATCH (catch).
- config.setCargando(false) solo en el then, no en errores.

js

```
<AceptarCancelarBtn
```

```

    aceptarFn={() => editarProductoFabricado()}
    disabled={true} // Siempre deshabilitado
  />

```

## ProductosFabricados.tsx

Vista principal del listado de productos fabricados. Ofrece filtros, paginación, exportación y alta de productos.

The screenshot shows a web application interface for 'Productos Fabricados'. It includes a sidebar with navigation links like 'ABMs de ítems', 'Materias Primas', 'Productos Fabricados', 'Ingeniería de Procesos', 'Fichas Técnicas', 'Grupos de Fichas Técnicas', 'Hojas de Procesos', 'Pesadas de Productos', 'Tiempo Setup de Máquinas', 'Operaciones Provisorias', and 'Lotes Económicos'. The main area displays a table of products with columns: Cód., Descripción, Stock Mín., Stock Crí., Peso Bruto, and Peso Neto. The table is paginated, showing 'Página 1 de 387' and '1931' items. There are buttons for '+ Alta de Productos Fabricados', 'Exportar todo', 'Imprimir Tabla', and 'Imprimir Tabla'. A 'Descargas' panel on the right shows two downloaded files: 'productos-fabricados.pdf' and 'productos-fabricados.xls', both marked 'Descarga finalizada'. A 'Borrar' button and a 'Mostrar más' link are also visible.

Cód.	Descripción	Stock Mín.	Stock Crí.	Peso Bruto	Peso Neto
27373	TE RED. 6" X 4"	0	0	28.852	17
6877	TE NORMAL 38"	0	0	883.4	786.74
25420	CODO RADIO LARGO 90° 12"	0	0	282.8	116.2
26389	RED. CON. 38" X 28"	0	0	348.648	186.4
27241	TE RED. 38" X 24"	0	0	753.427	580.554
33564	NIPLE TRANS. 4" L.250 MM	0	0	6.884	5.58
26421	RED. CON. 6" X 4"	0	0	4.488	3.6
27245	TE RED. 6" X 4"	0	0	13.9	11.9
26393	RED. CON. 6" X 3"	0	0	4.488	3.5
26493	RED. CON. 8" X 4"	0	0	7.446	6.1
26634	RED. CON. 16" X 12"	0	0	61.584	42.565
26635	RED. CON. 16" X 12"	0	0	59.3	39.2
27248	TE RED. 12" X 8"	0	0	187.915	82.786
24783	RED. EXC. 12" X 8"	0	0	33.6	26.1

Figura 2 – Listado de Productos Fabricados con filtros, paginación y acciones de exportación e impresión.

## Hooks y estado

- `tablaRef` para exportar/imprimir.
- Estados para tipos, familias, espesores, normas, unidades, productos y control de vista.
- Hooks: `useDescargarExcel`, `useFiltrar`, `useImprimirTabla`, `useMostrarError`, `useTraerConfiguracion`.
- `useEffect` carga datos iniciales con `hacerPetitionApi`.

## Funcionalidad

- Filtrado por código y descripción.
- Paginación con `ControlPaginas`.
- Alterna entre tabla y formulario de alta (`ProductosFabricadosAgregar`).
- Exporta todo a Excel y genera impresión de la tabla filtrada.

## Endpoints

- `GET /inventario/stockfabricados`

## Posibles fallas en la lógica

- setPagina(1) dentro de useMemo provoca bucle de renderizado infinito.
- La variable calidadesDePf está comentada, pero se invoca setCalidadesDePF en useEffect, causando error en tiempo de ejecución.

js

```
const muestra = useMemo(() => {
  setPagina(1); // No debe usarse dentro de useMemo
  return productosFabricados?.filter(...)
}, [filtrar.filtro, productosFabricados]);
```

## ProductosFabricadosTabla.tsx

Renderiza la tabla de productos fabricados, con estilos dinámicos y navegación al detalle.

### Props

- pagina: número de página actual.
- productosFabricados: lista de productos.
- tablaRef: referencia para exportación/imprimir.

### Funcionalidad

- Genera estilos de cabecera con generarEstilosDeCabezaDeTabla.
- Paginación con paginado.
- Resalta datos de baja y sin idErp.
- Permite navegar al detalle si hay permiso.

### Endpoints

No realiza peticiones; consume datos de sus props.

### Posibles fallas en la lógica

No se detectan problemas críticos en la renderización de la tabla.

## ProductosFabricadosAgregar.tsx

Formulario para dar de alta nuevos productos fabricados. Valida campos y envía datos a la API.

The screenshot shows a web application interface for adding a manufactured product. The main form is titled 'Agregar Producto Fabricado'. It includes a 'Descripción' field at the top. Below it, there are three main sections: 'Categorización', 'Dimensiones', and 'Ventas'. The 'Categorización' section has dropdowns for 'Tipo de Producto', 'Familia del Producto', 'Espesor del Producto', 'Cantidad del Producto', and 'Norma del Producto'. The 'Dimensiones' section has input fields for 'Diámetro 1 (centímetros de pulgada)', 'Diámetro 2 (centímetros de pulgada)', 'Paso Bruto (Pg.)', and 'Paso Neto (Pg.)'. The 'Ventas' section has a 'Unidad de Medida' dropdown and a 'Política de Stock' section with 'Stock Mínimo' and 'Stock Crítico' input fields. At the bottom, there are 'Guardar Cambios' and 'Cancelar' buttons.

Figura 3 – Formulario de alta de Productos Fabricados con carga de datos generales, dimensiones y política de stock.

## Props

- Listas de calidades, espesores, familias, normas, tipos y unidades.
- `setMostrarAgregar`: controla la visibilidad del formulario.
- `setProductosFabricados`: actualiza la lista tras creación.

## Estado y validación

- Estados para cada campo (tipo, familia, espesor, calidad, norma, dimensiones, pesos, stock).
- `esValido` con `useMemo` valida campos obligatorios.

## Funcionalidad

- Permite ingresar **descripción, categorización, dimensiones, pesos y política de stock**.
- Envía datos con POST y actualiza la tabla.
- Muestra modal si falla el envío al ERP.
- Soporta envío con tecla Enter.

## Endpoints

- `POST /inventario/stockfabricados`

## Posibles fallas en la lógica

- El array de dependencias de `esValido` omite **calidadId**, **normalId**, **diametro1** y **diametro2**, por lo que no se recalcula al cambiar esos campos.
- El campo **unidadDeMedidaId** está deshabilitado sin explicación de usuario.
- No se muestra indicador de carga durante la petición al crear.

js

```
const esValido = useMemo(() => {
```

```
// Faltan dependencias clave
return calidadId > 0 && descripcion.length > 6 && diametro1 > 0 && ...
}, [descripcion, espesorId, familiaId, pesoBruto, pesoNeto, stockCritico, stockMinimo, tipoId,
unidadDeMedidaId]);
```