

# Diagrama de flujo de navegación y datos ↵

## Componentes y Endpoints

A continuación se listan los componentes y los **endpoints** que invocan para interactuar con la API.

- **PesadasDeProductos**
- Endpoint: GET /v1/pesadas
- **PesadasOp**
- Endpoint: PATCH /v1/pesadas
- **PesadasTabla**
- No utiliza endpoints.
- **PesadasInformacion**
- No utiliza endpoints.

### GET /v1/pesadas

### PATCH /v1/pesadas

### PesadasDeProductos.tsx

Este componente orquesta la carga, filtrado y visualización de las **pesadas de productos**.

Utiliza hooks personalizados y distribuye datos a tres subcomponentes.

- Gestiona estados:
- pesadas: lista completa de órdenes de producción.
- muestra: lista filtrada por letra, descripción y transaccionables.
- pagina: página actual para paginación.
- opSeleccionada y mostrarInformacion: control de modales.
- Llama a la API al montar el componente.
- Proporciona botones para exportar, imprimir y ver información.
- Pasa referencias y callbacks a PesadasTabla, PesadasInformacion y PesadasOp.

Pesadas de Productos Fabricados

Exportar todo | Imprimir Tabla | Imprimir Tabla | Información

Letra | Producto | Ver solo Transaccionables

35 registros

Debe pesar próximo IPD	FT tipo P	Prod.	Alt.	Letra	Fam.	Med. 1	Med. 2	% var.	Pesada IPD	Peso Neto FT	Peso Bruto FT	P: Actualizó Maestros	S: Se sacaron las marcas	F: Fue pesada en IPD	Fecha Pesaje	Fecha Segundo Pesaje
st	st	59095	10	FADA	REE	100	50	-99.99	0.000	0.176	0.219	st	st	st		
st	st	59016	10	FACZ	REF	200	100	-100.00	0.000	0.520	0.740	st	st	st		
st	st	58014	10	FACY	REC	100	50	-100.00	0.000	0.150	0.189	st	st	st		
st	st	2655	10	FACX	CRL	600	0	-100.00	0.000	12.216	14.656	st	st	st		
st	st	2655	10	FACW	CRL	600	0	-100.00	0.000	12.216	14.656	st	st	st		
st	st	1825	10	FACV	CRC	1200	0	-100.00	0.000	40.650	54.814	st	st	st		
st	st	12848	10	FACU	CAS	600	0	0.00	24.000	24.000	12.736	st	st	st	8/5/2025	
st	st	3646	10	FACS	C45	400	0	-100.00	0.000	4.050	5.700	st	st	st		
st	st	3646	10	FACR	C45	400	0	-100.00	0.000	4.050	5.700	st	st	st		
st	st	3646	10	FACQ	C45	400	0	-100.00	0.000	4.050	5.700	st	st	st		
st	st	3642	10	FACP	C45	400	0	-100.00	0.000	1.890	2.103	st	st	st		
st	st	3654	10	FACO	C45	600	0	-100.00	0.000	8.750	12.092	st	st	st		
st	st	2145	10	FACN	CRL	800	0	-100.00	0.000	33.600	39.573	st	st	st		

Figura 1 - Pantalla principal de Pesadas de Productos con listado, filtros y acciones de exportar, impresión y ver información.

tsx

```
useEffect(() => {
  hacerPeticionApi().then(respuesta => {
    config.setCargando(false);
    const ps = respuesta.datos?.pesadas as ordenDeProduccionType[];
    if (!ps) {
      mostrarError(respuesta);
      return;
    }
    setPesadas(ps);
  });
}, []);
```

Endpoints utilizados:

- **GET /v1/pesadas**

## PesadasTabla.tsx

Presenta los datos en una tabla paginada y con formato condicional.

Permite seleccionar una fila si es **transaccionable**.

- Props recibidas:
- pagina, pesada[], setOpSeleccionada, tablaRef, transacciona.
- Genera estilos de encabezado dinámicos.
- Utiliza paginado para dividir resultados.
- Emplea componentes auxiliares: SiNo, NoHayElementos.

tsx

```
<table className='small' ref={tablaRef}>
  <thead>
    <tr>
      <th className='text-center'>Debe pesar próximo IPD</th>
      <!-- ... más columnas ... -->
    </tr>
```

```
</thead>
<tbody>
  {paginado(pesada, pagina).map(op => (
    <tr key={op.pesadaId}
        className={transacciona(op) ? 'deschover' : ''}
        onClick={() => transacciona(op) && setOpSeleccionada(op)}>
      <!-- celdas -->
    </tr>
  )) || <NoHayElementos />}
</tbody>
</table>
```

## Endpoints utilizados:

## PesadasInformacion.tsx

Modal que despliega la **documentación** del circuito de pesaje.

No interactúa con APIs, sólo muestra contenido estático y tablas explicativas.

- Captura la tecla ESC para cerrar.
  - Inyecta estilos CSS mínimos en el componente.
  - Contiene explicaciones detalladas y una tabla de combinaciones.

tsx

```
<h3 className='modal-title'>
    INFORMACIÓN DEL CIRCUITO "PESAJE IF"
</h3>
<!-- contenido largo con explicaciones -->
```

## Endpoints utilizados:

# PesadasOp.tsx

Modal de acción para **Aceptar**, **Ignorar** o **Separar** una pesada.

Envía la opción seleccionada al servidor y actualiza el estado global y local.



Figura 2 – Vista operativa de Pesadas de productos, mostrando el número de OP y las acciones disponibles para aceptar, ignorar o separar registros.

- Hooks de estado:
- `cargandoLocal`, `cantidadAlternativasActualizadas`.
- Despacha un modal de confirmación (Redux).
- Ejecuta `hacerPeticionApi` con método PATCH.
- Actualiza la lista de pesadas en el componente padre.

`tsx`

```
await hacerPeticionApi({
  method: 'PATCH',
  body: {
    opcion,
    ordenDeProduccionId: opSeleccionada.id
  }
});
```

Endpoints utilizados:

- **PATCH** /v1/pesadas

## Detección de posibles fallas lógicas ⚠

A continuación se señalan observaciones sobre la implementación:

- Se invoca `setPagina(1)` dentro de un `useMemo`, lo cual genera efectos secundarios indebidos.
- Los botones de impresión muestran el mismo texto (“Imprimir Tabla”) para PDF y Excel.
- La inyección de estilos con `<style>` en varios componentes puede provocar colisiones globales.
- `useImprimirTabla(tablaRef?.current, 'pesadas-de-productos')` podría ejecutar con `tablaRef.current` nulo.
- No hay manejo explícito de errores de red en `hacerPeticionApi`, más allá del modal genérico.
- Falta diferenciación entre la exportación completa y la impresión en el nombre de los botones.

Este análisis sugiere refactorizar los hooks de efecto, unificar estilos en archivos CSS y mejorar la validación de referencias.