

Diagrama de Navegación y Flujo de Datos

Componente Proyección de Deuda (ProyeccionDeuda.tsx)

Este componente carga y muestra la **lista completa de deudas** de producción.

Permite filtrar, exportar y navegar a detalle de semana firme o inspección final.

Producto	Familia	Cantidad	Cliente	Letra	Fecha Pedido	Fecha Entrega	Fecha OP	Semana	Probeta	Estado	IF	SP	Acción
1081 CODO RADIO CORTO 90° 1"	8	25	ACINVAL S.R.L.		5/4/2024	5/4/2024			S/P	V	FF	IF	Q
1082 CODO RADIO CORTO 90° 1.1/4"	8	50	ACINVAL S.R.L.		5/4/2024	5/4/2024			S/P	V	FF	IF	Q
29272 RED. EXC. 40° X 36"	54	8	SOLMET TECHNOLOGIES		31/1/2024	31/1/2024			S/P	V	FF	IF	Q
9115 RED. EXC. 4" X 3"	32	8	SOLMET TECHNOLOGIES		31/1/2024	31/1/2024			S/P	V	FF	IF	Q
29272 RED. EXC. 40° X 36"	54	8	SOLMET TECHNOLOGIES		8/2/2024	8/2/2024			S/P	V	FF	IF	Q
2801 CODO RADIO LARGO 90° 1/2"	9	3	SECCO JUAN F.		29/4/2024	29/4/2024			S/P	V	FF	IF	Q
1084 CODO RADIO CORTO 90° 1.1/2"	8	1	SECCO JUAN F.		29/4/2024	29/4/2024			S/P	V	FF	IF	Q
29272 RED. EXC. 40° X 36"	54	3	SOLMET TECHNOLOGIES		8/2/2024	8/2/2024			S/P	V	FF	IF	Q
29272 RED. EXC. 40° X 36"	54	3	SOLMET TECHNOLOGIES		31/1/2024	31/1/2024			S/P	V	FF	IF	Q
9115 RED. EXC. 4" X 3"	32	1	SOLMET TECHNOLOGIES		31/1/2024	31/1/2024			S/P	V	FF	IF	Q

Figura 1 – Pantalla principal de Proyección de deuda a clientes con listado de deudas, filtros, acciones principales y acciones de exportación e impresión.

Endpoints utilizados

- **GET /api/proyeccion-deuda**
Recupera proyeccionDeuda y pfFamilias.

Hooks y estados principales

- `useTraerConfiguracion()` → controla indicador de carga.
- `useMostrarError()` → muestra alertas en caso de fallo.
- `useFiltrar()` → gestiona filtros: cliente, fecha, producto, familia, semanal firme.
- `useDescargarExcel()` → descarga datos en Excel.
- `useImprimirTabla(tablaRef, fileName)` → imprime tabla en Excel o PDF.
- `useState`
- `página` (número de página)
- `proyeccionDeuda` (datos crudos)

Flujo de datos

1. Al montar, llama a `hacerPeticionApi()`.

2. Actualiza proyeccionDeuda y listaFamiliasPF.
3. Filtra vía useMemo según valores de filtrar.filtro.
4. Controla paginación con <ControlPaginas />.

Vista y acciones

Elemento	Descripción
Botón Exportar Excel	Descarga todo el set de datos.
Botones Imprimir	Genera Excel o PDF de los registros filtrados.
Campos de búsqueda	Cliente, Fecha Entrega, Cod. Producto, Familia, Semana.
Componente ControlPag.	Cambia página de resultados.
<ProyeccionDeudaTabla>	Renderiza los datos en tabla.

Fragmento clave de código

js

```
useEffect(() => {
  if (proyeccionDeuda) return;
  llevarAlTop();
  hacerPeticionApi().then(respuesta => {
    config.setCargando(false);
    const pdp = respuesta.datos?.proyeccionDeuda;
    const pfFam = respuesta.datos?.pfFamilias;
    if (!pdp || !pfFam) {
      mostrarError(respuesta);
      return;
    }
    setProyeccionDeuda(pdp);
    setListaFamiliasPF(pfFam);
  });
}, [filtrar.filtro, proyeccionDeuda]);
```

Componente Tabla de Proyección (ProyeccionDeudaTabla.tsx)

Este componente presenta los datos de deuda en una **tabla paginada**.

Permite fijar semana firme o consultar inspección final de cada registro.

Props recibidas

Prop	Tipo	Descripción	
muestra	deudaProduccion Type[] \	null	Registros filtrados y paginados.

pagina	number	Página actual.	
tablaRef	RefObject<HTMLTableElement>	Referencia para impresión/descarga.	

Navegación y acciones

- **FF** (Fijar Fecha Firme)

Llama a `navigate('detalle')` pasando el ítem seleccionado.

- **IF** (Inspección Final)

Llama a `navigate('inspeccionfinal')` con el estado del ítem.

Estructura de columnas

Columna	Tooltip
Producto	Id y descripción del producto.
Familia	Familia del producto.
Cantidad	Cantidad vendida.
Cliente	Nombre del cliente.
Letra	Letra de pedido.
Fecha Pedido	Fecha de emisión de orden de venta.
Fecha Entrega	Fecha compromiso de entrega.
Fecha OP	Fecha de lanzamiento de orden producción.
Semana	Día de la semana en firme.
Probeta	Estado de prueba.
Estado	Estado de la deuda.
IF SP	Cantidad en inspección final sin probeta.
Acción	Botones FF y IF.

Fragmento clave de código

jsx

```
<tbody>
  {muestra?.length
    ? paginado(muestra, pagina).map(s =>
```

```

<tr key={s.id}>
  <td>{s.productoFabricadoId} {s.productoFabricadoDescripcion}</td>
  {/* ...otras celdas */}

  <td>
    <button onClick={() => handleFijarFechaFirme(s)}>FF</button>
    <button onClick={() => handleConsultarIF(s)}>IF</button>
  </td>
</tr>
)
: <NoHayElementos />
</tbody>

```

Componente Planificación Semana Firme (ProyeccionDeudaEnFirme.tsx)

Aquí el usuario **asigna fecha firme** y cantidad a una orden de producción seleccionada.

Valida rangos y envía la planificación al servidor.

Figura 2 - Vista de Planificación de deuda firme para una planificación de Semana firme Seleccionada.

Endpoint utilizado

- **POST /api/proyeccion-deuda/semana-firme**

Body:

json

```
{
  "orden_de_produccion_id": number,
  "usuarioID": number,
  "fecha_firme": "YYYY-MM-DDTHH:MM:SSZ"
}
```

Hooks y estados

- **useLocation()** → obtiene selectedItem desde navegación.

- `useTraerConfiguracion()` → maneja indicador de carga.
- `useMostrarError()` → lanza errores.
- `useTraerUsuario()` → obtiene usuario actual.
- `useState`
- `fechaFirmeComprometida (Date)`
- `cantidadFirmeComprometida (number)`
- `guardando (boolean)`

Vista y validación

- Muestra datos de orden: producto, cliente, letra, cantidad adeudada.
- **DatePicker** para seleccionar fecha firme.
- Campo numérico para cantidad, con límites de 1 a `cantidadAdeudada`.
- Botones **Aceptar** y **Cancelar**.

Fragmneto clave de código

js

```
const handleAdd = async () => {
  setGuardando(true);
  await hacerPeticionApi({
    method: 'POST',
    body: {
      orden_de_produccion_id: selectedItem.ordenDeProduccionId,
      usuarioID: usuario.id,
      fecha_firme: fechaFirmeComprometida,
    }
  });
  setGuardando(false);
  navegar('..', null);
};
```

Componente Inspección Final (ProyeccionDeudaInspeccionFinal.tsx)

Este componente muestra el **estado de inspección final** de un producto fabricado.

Consulta stock aprobado, comprometido y calcula disponible.

Endpoint utilizado

- **GET** /api/proyeccion-deuda/inspeccion-final

Query param: `idPF`: id de producto fabricado.

Responde `inspeccionFinal: InspeccionFinalCantidadType[]`.

Hooks y estados

- `useLocation()` → obtiene `selectedItem`.
- `useTraerConfiguracion()` → controla carga.
- `useMostrarError()` → muestra errores.

- `useTraerUsuario()` → usuario actual.
- `useState`
- `InspeccionFinalCantidad` (datos de IF)
- `IFDisponible` (calculado)
- guardando (boolean)

Flujo de datos

1. `useEffect` dispara `fetchIFCantidades` al cargar componente.
2. Llama `hacerPeticionApi({ method: 'GET', queryParams: { idPF } })`.
3. Actualiza `InspeccionFinalCantidad` y calcula `IFDisponible`.

Vista y cálculos

Campo	Valor
Stock	<code>cantidadAprobada</code> .
Comprometido	<code>cantidadRetenida</code> .
Disponible	<code>aprobada - retenida</code> .

El campo **Disponible** cambia de color si queda negativo.

Fragmento clave de código

js

```
const fetchIFCantidades = async (idPF) => {
  const respuesta = await hacerPeticionApi({
    method: 'GET',
    queryParams: { idPF },
  });
  const IFC = respuesta.datos?.inspeccionFinal[0];
  setInspeccionFinalCantidad(IFC);
  setIFDisponible(IFC.cantidadAprobada - IFC.cantidadRetenida);
  config.setCargando(false);
};
```

Nota: El componente se encuentra definido a nivel de frontend, pero su funcionamiento es parcial y no permite la visualización efectiva del detalle.

¡Listo! Esta documentación abarca la navegación, los componentes y los endpoints clave.