

## Diagrama de flujo de navegación y datos

Este diagrama muestra cómo interactúan los componentes de Forecast con la API y entre sí.

Se reflejan las rutas de datos, las acciones de usuario y las actualizaciones de estado.

### ForecastEditor.tsx

Componente para editar una fila de planificación de venta en línea dentro de la tabla.

Carga datos auxiliares, valida entradas y envía las peticiones **PUT** y **DELETE**.

- **Props**
  - forecast: objeto con datos del registro a editar.
  - setDatosForecast: actualiza el listado tras la operación.
  - setIdMostrarEditar: controla la visibilidad del formulario.
- **Estados**
  - articuloId, cantidad, cliente, monedaId, precio
  - fechaCompromiso: Date convertido desde SQL.
  - Listas: productosFabricados, monedas, clientes.
  - guardando: indica operación en curso.
- **Hooks y validación**
  - useEffect para cargar listas desde ?abm=true.
  - useMemo calcula esValido si todos los campos cumplen.
  - useMostrarError muestra errores de API.
- **Endpoints utilizados**

Método	Ruta	Descripción
GET	/forecast?abm=true	Carga productos, monedas y clientes
PUT	/forecast	Actualiza una planificación de venta
DELETE	/forecast	Elimina una planificación de venta

- **Ejemplo de llamada PUT**

js

```
const respuesta = await hacerPeticionApi({
  method: 'PUT',
  body: {
    articuloId,
    cantidad,
    cliente,
    fechaCompromiso: formatearFechaHoraJsParaSql(fechaCompromiso),
```

```

    forecastId: forecast.id,
    monedaId,
    precio
  }
});

```

## ForecastAgregar.tsx

Formulario para crear una nueva planificación de venta.

Carga datos auxiliares y envía una petición **POST** con el nuevo registro.

- **Props**
- `setDatosForecast`: actualiza el listado tras creación.
- `setMostrarAgregar`: oculta el formulario al cancelar o completar.
- **Estados**
- `productoFabricadoId`, `cantidad`, `clienteId`, `monedaId`, `precio`
- `fechaCompromiso`: Date seleccionada.
- Listas: `productosFabricados`, `monedas`, `clientes`.
- `guardando`: controla el estado de envío.
- **Hooks y validación**
- `useEffect` para obtener datos desde `?abm=true`.
- `useMemo` determina `esValido` según campos requeridos.
- `useMostrarError` gestiona errores de API.
- **Endpoints utilizados**

Método	Ruta	Descripción
GET	/forecast?abm=true	Carga productos, monedas y clientes
POST	/forecast	Crea una nueva planificación

- **Ejemplo de llamada POST**

js

```

const respuesta = await hacerPeticiónApi({
  method: 'POST',
  body: {
    clienteId,
    fechaCompromiso: formatearFechaHoraJsParaSql(fechaCompromiso),
    productosFabricados: [{
      cantidad,
      fechaCompromiso: formatearFechaHoraJsParaSql(fechaCompromiso),
      monedaId,
      precio,
      productoFabricadoId
    }]
  }
});

```

## ForecastTabla.tsx

Componente presentacional que muestra la lista paginada de forecasts.

No realiza peticiones API; solo formatea y despliega datos.

- **Props**
- `datosForecast: vendidoType[]`
- `pagina: number`
- `productoFabricadoId: number` (filtro por producto)
- **Lógica interna**
- `useState` para `idMostrarEditar`, controla edición inline.
- Usa servicios:
- paginado para dividir resultados.
- `formatearFecha` para fechas legibles.
- `generarEstilosDeCabezaDeTabla` para estilos dinámicos.
- **Estructura de la tabla**
- Dos filas de encabezado con metadatos de cliente y producto.
- Filas de datos agrupadas por orden de venta.
- Destaca filas eliminadas con clase `bg-warning`.
- Muestra `NoHayElementos` si no hay datos.

## Forecast.tsx

Componente principal que orquesta carga, filtrado, exportación e importación de forecasts.

Gestiona el estado global de la vista y delega en los subcomponentes.

- **Estados**
- `datosForecast: vendidoType[] | null`
- `mostrarAgregar: boolean`
- `pagina: number`
- **Hooks propios**
- `useTraerConfiguracion`: controla indicador de carga.
- `useMostrarError`: gestiona errores de API.
- `useDescargarExcel`: exporta datos a Excel.
- `useFiltrar`: mantiene filtros de búsqueda.
- `useAcceder`: navegación controlada a sugerencias OP.
- **Efectos y funciones**
- `useEffect` inicial:

js

```
hacerPeticionApi().then(respuesta => {  
  config.setCargando(false);  
  setDatosForecast(respuesta.datos?.datosForecast);  
});
```

- `enviarArchivoCsv`: envía CSV de forecasts nuevos:

js

```
const formData = new FormData();
formData.append('file', selectedFile);
const respuesta = await hacerPeticiónApi({
  method: 'POST',
  formData,
  url: window.location.pathname + '/csv'
});
setDatosForecast(respuesta.datos?.datosForecast);
```

- **Endpoints utilizados**

Método	Ruta	Descripción
GET	/forecast	Obtiene lista de planning ventas
POST	/forecast/csv	Importa datos desde CSV

- **Flujo de renderizado**
- Muestra un spinner Cargando mientras carga datos.
- Despliega botones de exportar y sugerencias.
- Incluye filtros por orden, producto, cliente y estado.
- Alterna entre `ForecastTabla` y `ForecastAgregar` según `mostrarAgregar`.

## Card de recomendación

### Validación previa

Verifica que todos los campos estén completos antes de aceptar.