

Diagrama de flujo de navegación y datos

Este diagrama muestra la interacción entre componentes y las llamadas a la API. Incluye la navegación entre “Opciones” y el módulo de “Plantas”.

[Ver diagrama completo de navegación y flujo de datos](#)

Endpoints utilizados por componente

- **Opciones.tsx**
- POST /api/opciones?opcion={n}
- **Plantas.tsx**
- GET /api/plantas
- **PlantasAregar.tsx**
- POST /api/plantas
- **PlantasEditar.tsx**
- PATCH /api/plantas
- DELETE /api/plantas
- **PlantasTabla.tsx**
- No realiza llamadas directas a API

Fallas en la lógica

- En **PlantasEditar.tsx**, el campo IIBB muestra cuitNuevo en vez de iibbNuevo.
- En **PlantasAregar.tsx**, el checkbox de IIBB está deshabilitado y no permite IIBB distinto de CUIT.
- Gran parte del código de **PlantasEditar**, **PlantasAregar**, **Plantas**, **PlantasTabla** está comentado, impidiendo funcionalidad activa.
- En **Opciones.tsx**, la recarga sugerida tras la animación no se ejecuta ni se documenta.
- useTeclaEnter en **PlantasEditar** podría disparar acciones sin control de foco.

Opciones.tsx

Componente que muestra acciones de sincronización y mantenimiento.

Permite forzar, pausar y eliminar transacciones vía modal de confirmación.

Elemento	Descripción
opciones array	Lista de botones con etiqueta, mensaje de confirmación y código.
aceptar(opción...)	Despliega modal, ejecuta hacerPeticionApi y muestra resultado.

Hooks y servicios utilizados:

- useTraerConfiguracion: controla estado de carga.
- useTraerUsuario: obtiene nivel de usuario.
- useMostrarError: muestra errores API.
- hacerPeticionApi: realiza llamadas POST.

Ejemplo de llamada a la API:

js

```
await hacerPeticionApi({
  method: 'POST',
  queryParams: { opcion }
});
```

PlantasEditar.tsx

Componente para editar o eliminar una planta en la tabla.

Actualmente todo el código está comentado, pendiente de habilitar.

Propiedades (props):

Prop	Tipo	Descripción
planta	plantaType	Datos de la planta a editar
setIdMostrarEdicion	Dispatch<SetStateAction<number>>	Controla fila en modo edición
setPlantas	Dispatch<SetStateAction<plantaType[]>>	Actualiza lista de plantas tras cambio

Estados internos:

- nombreNuevo, direccionNueva, cuitNuevo, iibbNuevo: campos de formulario.
- guardando: indicador de petición en curso.

Flujo principal:

- Captura cambios en inputs.
- En Enter o botón “Aceptar” ejecuta PATCH o DELETE.

- Actualiza la lista de plantas o muestra error.

PlantasAregar.tsx

Formulario para dar de alta nuevas plantas.

El componente está comentado y su lógica queda en standby.

Propiedades (props):

Prop	Tipo	Descripción
setMostrarAgregar	Dispatch<SetStateAction<boolean>>	Muestra u oculta el formulario
setPlantas	Dispatch<SetStateAction<plantaType[]>>	Actualiza lista tras crear planta

Estados internos:

Estado	Tipo	Función
nombre	string	Nombre de la planta
direccion	string	Dirección de la planta
cuit	string	CUIT formateado
iibb	string	IIBB de la planta (se iguala o personaliza)
iibbMismo	boolean	Indica si IIBB es igual a CUIT
guardando	boolean	Indica petición en curso

Hook clave:

- useMemo para validar formulario.

Plantas.tsx

Contenedor principal de la sección “Plantas”.

Gestiona carga, alta e impresión/descarga de la tabla.

Estados y referencias:

Elemento	Tipo	Descripción	

<code>idMostrarEdicion</code>	<code>number</code>	ID de planta en modo edición	
<code>mostrarAregar</code>	<code>boolean</code>	Controla visibilidad de formulario alta	
<code>plantas</code>	<code>`plantaType[]`</code>	<code>null`</code>	Lista de plantas
<code>tablaRef</code>	<code>RefObject<HTMLTableElement></code>	Referencia para impresión/exportación	

Hooks y servicios:

- `useTraerConfiguracion`: controla indicador de carga.
- `useDescargarExcel`: descarga datos en XLSX.
- `useImprimirTabla`: imprime la tabla en Excel/PDF.
- `useMostrarError`: gestiona errores API.
- `useTraerUsuario`: obtiene nivel de usuario.

Flujo principal:

1. Al montar, ejecuta GET /api/plantas.
2. Muestra componentes `PlantasTabla` o `PlantasAregar`.
3. Permite alternar entre alta y listado.
4. Ofrece exportar o imprimir según permisos.

PlantasTabla.tsx

Renderiza la tabla de plantas y controla edición en línea.

Filtre botones de editar según permisos de usuario.

Propiedades (props):

Prop	Tipo	Descripción	
<code>idMostrarEdicion</code>	<code>number</code>	ID de la fila en edición	
<code>plantas</code>	<code>`plantaType[]`</code>	<code>null`</code>	Datos a mostrar
<code>setIdMostrarEdicion</code>	<code>Dispatch<SetStateAction<number>></code>	Activa/desactiva modo edición	
<code>setPlantas</code>	<code>Dispatch<SetStateAction<plantaType[]>></code>	Actualiza lista tras editar/eliminar	

tablaRef	RefObject<HTMLTableElement>	Referencia para export/impresión	
----------	-----------------------------	----------------------------------	--

Funciones clave:

- tienePermiso(usuario, planta): valida acceso a edición.
- generarEstilosDeCabezaDeTabla: ajusta estilos CSS dinámicos.
- Renderiza <NoHayElementos/> si no hay datos.

Flujo:

- Muestra botón de editar si el usuario tiene nivel mayor.
- Alterna la fila de edición con <PlantasEditar/>.
- Mantiene diseño responsive con clases Bootstrap.

¡Documentación completa y lista para revisión!