

Diagrama de flujo de navegación y datos

Este diagrama muestra la interacción entre los componentes principales y las peticiones a la API.

Permite visualizar cómo el usuario navega desde la lista de programas hasta las búsquedas y pruebas de funciones.

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

ProgramasAuxiliares.tsx

Este componente lista los **Programas Auxiliares** disponibles y permite navegar a su definición.

Carga inicialmente todos los programas y muestra subprogramas al seleccionar uno.

The screenshot shows a web application interface for managing auxiliary programs. The top navigation bar includes links for Producción / MRP, Ingeniería, Inventario, Piso de Planta, Calidad, Mantenimiento, Costos, and Administración. The current view is under the Administración tab. On the left, there's a sidebar with various administrative sections like Depósitos, Centros de Costo, Etapas / Subetapas, Legajos Empleados, Proveedores y Clientes, Recursos (Máquinas, Instrumentales, Matrícula), Configuración (Usuarios, Roles, Opciones, Logs), and Reglas de Negocio (Programas Auxiliares). The main content area is titled 'Programas Auxiliares' and contains a table with 21 rows, each representing a program. The columns are 'Cód.' (Code), 'Programa' (Program), and 'Acciones' (Actions). The 'Acciones' column contains a blue circular icon with a white arrow pointing right, followed by the text 'Ir a definición'. For rows 8, 10, 12, 20, and 21, there is also a green button labeled 'Ver todos' (View all) next to the 'Ir a definición' button. The table rows are numbered 1 through 21 and list various program names such as 'Perfil de Barrada', 'Sugerencias de Colores', 'Cálculo de PCM y C. Eq.', etc.

Figura 1 – Pantalla principal de Programas Auxiliares con listado y acciones ir a definición y ver todos.

Endpoints

- **GET /programas-auxiliares**

Recupera el listado completo de programas y su estructura padre-hijo.

Fallas detectadas

- Ninguna lógica crítica detectada.

ProgramaAuxiliar.tsx

Muestra el **detalle de un Programa Auxiliar**, su definición SQL y permite editar, imprimir o ver su historial.

Integra los subcomponentes de prueba y búsqueda de IDs cuando es la versión vigente.

The screenshot shows a software interface for managing auxiliary programs. The top navigation bar includes links for Producción / MRP, Ingeniería, Inventario, Piso de Planta, Calidad, Mantenimiento, Costos, and Administración. The current user is 'supervisor (Superadministrador)'. The main content area is titled 'Función de Asignación de Perfil de Barrada' (Assignment Profile Function). It displays the following details:

- Nombre de la Función: FuncionAsignacionBarrada
- Parámetro de Salida: VARIABLES
- Parámetros de Entrada: (p_opid INT)
- Observaciones: El Perfil de Barrada que es un texto
- Ejecución: Cuando se lanza una OP (no cuando solo se planifica)
- Fecha de la versión: 1/2/2025 08:46 a. m.

The code editor contains the following SQL code:

```
CREATE FUNCTION FuncionAsignacionBarrada (p_opid INT) RETURNS VARCHAR(5)
```

```
3   DECLARE v_norma INT;
4   DECLARE v_barrera INT;
5   DECLARE v_norma_v INT;
6   DECLARE v_aprox INT;
7   DECLARE v_norma INT;
8   EXECUTE sp_hex2int;
9   DECLARE v_soldado INT;
10  EXECUTE sp_hex2int;
11  DECLARE v_norma INT;
12  EXECUTE sp_hex2int;
13  DECLARE v_norma INT;
14  EXECUTE sp_hex2int;
15  EXECUTE sp_hex2int;
16  DECLARE v_barrera VARCHAR(4);
17  DECLARE v_norma VARCHAR(4);
18
19  SELECT COUNT(*) INTO v_norma FROM ordenes_de_producción op
20  LEFT JOIN fichas_técnicas_procesos ft ON op.ficha_técnica_id = ft.ficha_técnica_id
21  WHERE op.op_id = p_opid AND ft.id_subetapa_id = 84;
22
23  --> 0 normas <-> 1 norma
24  IF v_norma > 0 THEN SET v_norma = 1; END IF;
25
26  SELECT COUNT(*) INTO v_soldado
27  FROM ordenes_de_producción op
28  LEFT JOIN fichas_técnicas_procesos ft ON op.ficha_técnica_id = ft.ficha_técnica_id
29  LEFT JOIN etapas_subetapas et ON ft.etapa_id = et.id
30  WHERE op.op_id = p_opid AND (etapa = 15 OR (etapa = 6 AND subetapa = 1));
31
32  --> 0 soldados <-> 1 soldado
33  IF v_soldado > 0 THEN SET v_soldado = 1; END IF;
34
35  SELECT COUNT(*) INTO v_aprox
36  FROM ordenes_de_producción op
37  LEFT JOIN fichas_técnicas_procesos ft ON op.ficha_técnica_id = ft.ficha_técnica_id
38
39
```

Figura 2 - Vista detallada de un programa auxiliar, con acciones de editar, ver historial, observaciones, probar programa e impresión.

Endpoints

- **GET /programa-auxiliar**

Obtiene datos del programa según **id** en query params.

- **PATCH /programa-auxiliar**

Actualiza la definición y las observaciones enviadas en el body.

Fallas detectadas

- El botón “Volver a esta Versión” tiene la clase d-none y nunca se muestra.
- La función de reactivar versión está comentada, lo que puede confundir si se planeaba usar.

ProgramaAuxiliarProbar.tsx

Permite **probar** una función SQL del programa auxiliar enviando sus parámetros.

Construye dinámicamente los inputs según la firma (**nombre tipo, ...**).

Endpoints

- **GET /programa-auxiliar/prueba**

Envia los pares **nombre=valor** como query params y devuelve el resultado.

Fallas detectadas

- Si la función retorna 0, se considera error y muestra mensaje, aunque 0 podría ser un resultado válido.
- La deducción de tipo (int vs otros) solo mira si `tipo.toLowerCase() == 'int'`, no contempla otras tipificaciones.

ProgramaAuxiliarBusca.tsx

Integra tres vistas de búsqueda de **IDs auxiliares**: ficha técnica, OP y colada.

Controla la pestaña activa y muestra el componente correspondiente.

Endpoints

- Ninguno directamente; each subvista realiza su propia petición.

Fallas detectadas

- Estado inicial `vista = ''` deja la tarjeta sin contenido hasta el primer clic.

ProgramaAuxiliarBuscaFt.tsx

Busca el **ID de Ficha Técnica** según producto fabricado y alternativa.

Valida inputs numéricos y muestra resultado o mensaje si no existe.

Endpoints

- **GET /ficha-tecnica**

Query params: `productoFabricadoId, alternativa`

Fallas detectadas

- Uso de 999999999 como código de “no existe” es un poco descriptivo.

ProgramaAuxiliarBuscaOp.tsx

Busca el **ID de Orden de Producción** según la letra de OP formateada.

Aplica `formatarLetraOp` y muestra un spinner durante la petición.

Endpoints

- **GET /orden-produccion**

Query params: `letra`

Fallas detectadas

- Igual que en FT, usa 999999999 para “no existe”.
- Si la API devuelve 0, se interpreta como error aunque podría ser ID válido.

ProgramaAuxiliarBuscaCol.tsx

Busca el **ID de Colada**, indicando si es de “cintolo”.

Este archivo contiene un desajuste en el nombre del componente.

Endpoints

- **GET /colada**

Query params: colada, escintolo

Fallas detectadas

- El componente se declara como ProgramaAuxiliarBuscaFt en lugar de ProgramaAuxiliarBuscaCol.
- Produce confusión y posibles errores de importación.

ReglasYVigias.tsx

Archivo con la definición de tipos y lógica para reglas y vigías, **todo comentado**.

No exporta ni renderiza ningún componente activo.

Endpoints

- Ninguno (código deshabilitado)

Fallas detectadas

- El código está completamente comentado, no tiene funcionalidad.
- Falta exportación y uso de hooks para configuración o navegación.