

Documentación Analítica de Capacidad

Diagrama de Flujo de Navegación y Datos

AnaliticaDeCapacidad.tsx

Una página que muestra la **analítica de capacidad** de máquinas.

Recupera datos, aplica filtros, exporta e imprime resultados.

Máquina				
Tipo-Nro.	Descripción	Hs Demanda Diaria	Hs Disponibles	Estado
01 0005	BISELADORA PARA CODOS DE 2,5 HASTA 8	-	-	-
01 0008	BISELADORA NRO. 8 ° CODOS DE 45	-	-	-
01 0009	BISELADORA NRO. 9 ° PMC H/38 PULG.	-	-	-
01 0010	BISELADORA NRO. 10 ° CODOS 45 H/10 PULG.)	-	-	-
01 0012	BISELADORA LOSINO	-	-	-
02 0002	TORNO DE BISELADO NRO: 2 (LATHE)	-	-	-
02 0024	TORNO VERTICAL TOS	-	-	-
04 0009	OXICORTE C-PLATO GIRATORIO MEDIANO	-	-	-
04 0010	OXICORTE C-PLATO GIRATORIO CHICO	-	-	-
04 0021	PANTOGRAFO C/CONTROL NUMERICO	-	-	-
07 0001	CURVADORA N°1	-	-	-
07 0002	CURVADORA N°2	-	-	-
07 0004	CURVADORA N°4	-	-	-

Figura 1 – Pantalla principal de Analítica de capacidad con listado de Máquina, búsqueda y acciones de exportación e Impresión.

Descripción

- Carga lista de maquinaType vía API.
- Ofrece filtros y control de páginas.
- Permite exportar e imprimir en Excel o PDF.

Flujo de Datos

- useEffect llama a hacerPeticionApi() para GET.
- Estados: datosAnaliticaCapacidad, pagina.
- useMemo filtra según filtrar.filtro.descripcion1.
- Renderiza <AnaliticaDeCapacidadTabla> con datos paginados.

Endpoints

GET Datos de Analítica de Capacidad

Puntos Clave

- **useTraerConfiguracion** controla estado de carga.
- **useFiltrar** gestiona texto de búsqueda.
- **useDescargarExcel** y **useImprimirTabla** ofrecen exportación.
- **ControlPaginas** ajusta paginación.

Ejemplo de Código

js

```
useEffect(() => {
  hacerPeticionApi().then(respuesta => {
    config.setCargando(false);
    const dac = respuesta.datos?.datosAnaliticaCapacidad;
    if (!dac) return mostrarError(respuesta);
    setDatosAnaliticaCapacidad(dac);
  });
}, []);
```

AnaliticaDeCapacidadTabla.tsx

Componente que **renderiza la tabla** de analítica de capacidad.

Separa cabecera y cuerpo con estilos personalizados.

Descripción

- Recibe datos paginados y página actual.
- Genera estilos de cabecera con `generarEstilosDeCabezaDeTabla`.
- Muestra filas con `pagina()` o mensaje de vacío.

Props

- `datosAnaliticaCapacidad: maquinaType[]`
- `pagina: number`
- `tablaRef: RefObject<HTMLTableElement>`

Renderizado de Tabla

- `<thead>` con títulos multi-nivel.
- `<tbody>` con filas y celdas formateadas.
- `<NoHayElementos />` si no hay datos.

Navegación

- **useAcceder** permite click en fila.
- `onClick` navega a ruta `/maquina?id=<m.id>`.

ADCMaquina.tsx

Vista de **Procesos de una Máquina** seleccionada.

Muestra drag & drop, filtros y guardado de orden.

Sec	Cliente	Art.	Fam.	Med1	Med2	Espesor	Et/S	Calidad	Norma	Letra	Letra Or.	DiasXOP	AcuDs	Corre MP	Stk.Pt	SemStk	Fch.PedidoVenta	Fch.CompromisoEntrega	Fch. OF
=	1 (pendiente)	3029	C45	250	0	STANDARD	07/09	A234-WPB	ASME B16.9	FACI		0.91	0.91	C	0	-	-	28/3/2025	
=	2 (pendiente)	3042	C45	400	0	LIVIANO	07/09	A234 B	ASME B16.9	FACP		0.50	1.48	C	0	-	-	28/3/2025	
=	3 (pendiente)	3029	C45	250	0	STANDARD	07/09	A234-WPB	ASME B16.9	FACJ		0.69	2.10	C	0	-	-	28/3/2025	
=	4 (pendiente)	3046	C45	400	0	SCH.128	07/09	A234-WPB	ASME B16.9	FACQ		0.25	2.35	C	0	-	-	28/3/2025	
=	5 (pendiente)	3046	C45	400	0	SCH.128	07/09	A234-WPB	ASME B16.9	FACR		0.08	2.43	C	0	-	-	28/3/2025	
=	6 (pendiente)	3046	C45	400	0	SCH.128	07/09	A234-WPB	ASME B16.9	FACS		0.08	2.51	C	0	-	-	28/3/2025	

Figura 2 - Vista de Analitica de capacidad para un Proceso de una Máquina seleccionado.

Descripción

- Carga un maquinaType con sus procesosDeOp.
- Ofrece filtros: letra, número OP y etapa/subetapa.
- Permite reordenar procesos y guardar cambios.

Flujo de Datos

- GET inicial con hacerPeticionApi().
- Estado: maquina, guardando, pagina.
- useMemo filtra y ordena procesos.
- guardar() lanza PATCH con objeto maquina.

Endpoints

GET Máquina con Procesos

PATCH Guardar Orden de Procesos

Funcionalidades

- Botón **Volver** via useNavegar.
- Exportar e imprimir tabla (Excel/PDF).
- Guardar cambios deshabilita botones mientras guarda.
- Control de páginas ajusta vista global.

Ejemplo de Guardar

js

```

const guardar = async () => {
  setGuardando(true);
  const respuesta = await hacerPeticionApi({
    method: 'PATCH',
    body: { maquina }
  });
  setGuardando(false);
  if (!respuesta.datos?.maquina) return mostrarError(respuesta);
  setMaquina(respuesta.datos.maquina);
};

```

ADCMaquinaCientes.tsx

Modal sencillo para **mostrar clientes** asociados a un proceso.

Solo controla apertura y cierre del diálogo.

Figura 3 - Vista de Análítica de capacidad para mostrar clientes asociados a un proceso.

Descripción

- Estructura de modal Bootstrap.
- Cabecera con título “Clientes”.
- Cuerpo vacío para contenido futuro.
- Footer con botón **Cerrar**.

Props

- `cerrarModal: () => void` — callback para cerrar modal.

Uso

- Se monta cuando `procesoId` en `ADCMaquinaTabla` es distinto de 0.
- `onClick` en botones invoca `cerrarModal`.

ADCMaquinaTabla.tsx

Tabla de procesos con **drag & drop** y **react-table**.

Incluye navegación a clientes y verificación de permisos.

Descripción

- Usa `@tanstack/react-table` para columnas dinámicas.
- Permite reordenar filas con `react-dnd`.
- Renderiza modal de clientes al click en secuencia.

Columnas

- Más de 30 columnas: Secuencia, Cliente, PF ID, Fam, Medidas, Et/Se, Calidad, Norma, Letra, etc.
- Cada celda activa `onClick` para abrir modal.

Drag & Drop

- `useDrag` en botón de agarre.
- `useDrop` en fila objetivo para intercambio.
- Al soltar, actualiza `procesosDeOp` con nueva secuencia.

Permisos

- `useTraerUsuario` recupera datos de usuario.
- `tienePermiso(usuario, maquina)` controla visibilidad del agarre.

Modal de Clientes

- Estado local `procesoId` indica fila activa.
- Renderiza `<ADCMaquinaClientes>` cuando `procesoId ≠ 0`.

Estilos de Tabla

- `generarEstilosDeCabezaDeTabla([' '])` ajusta estilos CSS.
- Aplica en `<style>` dentro del componente.

No Hay Elementos

- Si `procesos.length === 0` muestra `<NoHayElementos />`.

Ejemplo de Fila Arrastrable

js

```
const DraggableRow = ({ proceso }) => {
  const [, dropRef] = useDrop({ /* configuración */ });
  const [{ isDragging }, dragRef, previewRef] = useDrag({ /* config */ });
  return (
    <tr ref={previewRef} style={{ opacity: isDragging ? 0.5 : 1 }}>
      {tienePermiso(usuario, maquina) && (
        <td ref={dropRef}>
          <button ref={dragRef} className='btn btn-primary'>
            <i className='fa-solid fa-grip-lines' />
          </button>
        </td>
      )}
      {proceso.getVisibleCells().map((cell, i) => (
        <td key={i}>
```

```
        {flexRender(cell.column.columnDef.cell, cell.getContext())}
    </td>
)}
```

```
</tr>
);
};
```

¡Listo! Este documento explica con detalle cada componente, sus datos y flujos.