

MANUAL TÉCNICO



Carmen Pulido Díaz

PFC – DAW 2025/26

Contenido

1.	Descripción general	2
2.	Tecnologías utilizadas.....	2
3.	Estructura de carpetas.....	2
4.	Principales componentes.....	3
5.	Dependencias.....	3
6.	Configuración y despliegue.....	4
	• Acceso a la aplicación:.....	4
	• Instalación y ejecución en local:	4
7.	Seguridad y autenticación.....	4
8.	Base de Datos	5
	• Tablas	5
	• Relación productos-ventas	5
	• Otras consideraciones.....	5
	• Esquema BBDD	6

1. Descripción general

TechnoFix es una aplicación web para la gestión de reparaciones, clientes, ventas y usuarios en un entorno de taller técnico. Permite controlar el flujo de trabajo, la facturación y la administración de clientes y productos.

2. Tecnologías utilizadas

- **Frontend:** React 19, Vite, Bootstrap 5, styled-components
- **Backend:** Supabase (PostgreSQL, Auth, Storage)
- **Otros:** html2canvas, jsPDF, jsPDF-autotable

3. Estructura de carpetas

```
src/
```

```
  assets/
```

```
  components/
```

```
  contexts/
```

```
  hooks/
```

```
  pages/
```

```
  routers/
```

```
  store/
```

```
  styles/
```

```
  supabase/
```

```
  utils/
```

```
  App.jsx
```

```
  index.js
```

```
  main.jsx
```

4. Principales componentes

- **Clientes:** Gestión, búsqueda, edición y generación de PDF de información de clientes.
- **Reparaciones:** Alta, modificación, consulta y generación de PDF de las mismas.
- **TPV (Caja):** Venta de productos, gestión de líneas, cobro con selección de cliente y listado de facturaciones con generación de PDF.
- **Usuarios:** Sólo para Administradores. Administración (para dar roles) y registro de usuarios. El usuario admin no podrá eliminarse ni editarse rol.
- **Empresa:** Sólo para Administradores. Personalización de logo y nombre de empresa en la aplicación, así como en los documentos, en los cuáles se puede editar su pie de página.

5. Dependencias

- **@supabase/supabase-js:** Cliente oficial de Supabase. En TechnoFix se utiliza para gestionar la autenticación de usuarios, el acceso seguro a la base de datos (clientes, reparaciones, productos, ventas, usuarios) y el almacenamiento de archivos (como logos y PDFs) directamente desde el frontend.
- **bootstrap:** Framework CSS. Se emplea para dar soporte visual y responsive a algunos componentes y tablas, facilitando la adaptación a dispositivos móviles y mejorando la experiencia de usuario.
- **express:** Framework para Node.js. Se utiliza en scripts auxiliares para la generación de PDFs o tareas de backend puntuales, aunque la aplicación principal es frontend y serverless.
- **html2canvas:** Permite capturar partes del DOM y convertirlas en imágenes. En TechnoFix se usa para generar capturas de pantalla de tickets o listados, que luego pueden ser exportados o incluidos en PDFs.
- **jspdf:** Librería para crear y manipular archivos PDF desde el frontend. Se utiliza para que el usuario pueda descargar informes, tickets o listados de clientes y reparaciones en formato PDF.
- **jspdf-autotable:** Plugin para jsPDF que facilita la creación de tablas dentro de los PDFs. En TechnoFix se usa para exportar listados de clientes, reparaciones o ventas en formato tabla dentro de los PDFs generados.
- **react:** Biblioteca principal para construir la interfaz de usuario basada en componentes reutilizables. Toda la lógica de vistas, navegación y estado se gestiona con React.
- **react-dom:** Permite renderizar componentes React en el DOM del navegador. Es esencial para que la aplicación funcione en cualquier navegador moderno.

6. Configuración y despliegue

- **Acceso a la aplicación:**

La versión en producción está disponible públicamente en la siguiente URL:
<https://carmenpd29.github.io/TechnoFix/login>

- **Instalación y ejecución en local:**

Si se desea probar o modificar el proyecto en local:

1. Clonar el repositorio desde GitHub.
2. Instalar dependencias: ' npm install '
3. Para desarrollo local, ejecutar: ' npm run dev '
4. Para producción, compilar con: ' npm run build '
5. El contenido generado en la carpeta ' docs/ ' se puede publicar en GitHub Page.
6. Configurar las variables de entorno de Supabase en el archivo correspondiente ('.env' o directamente en el código).
7. No es necesario configurar ni desplegar un servidor backend propio, ya que Supabase gestiona la base de datos, autenticación y almacenamiento.
8. Si se modifica la seguridad, revisar las políticas RLS en Supabase y la persistencia de sesión en el store de usuario (Zustand).

7. Seguridad y autenticación

- **Validación** y sanitización de entradas en formularios y búsquedas.
- **Autenticación y autorización con Supabase:** solo usuarios autenticados pueden acceder a rutas privadas.
- **Protección de rutas en frontend:** ningún usuario sin sesión puede acceder a vistas internas ni por URL directa.
- **Persistencia de sesión:** en el estado de usuario tras recargas y navegación interna.
- **Gestión de roles:** solo los administradores pueden gestionar usuarios y configuración de empresa. El usuario admin no puede ser editado ni eliminado desde la interfaz.
- **Políticas RLS** (Row Level Security) activas en la base de datos para proteger los datos según el usuario autenticado.

8. Base de Datos

• Tablas

- **clientes**: Almacena los datos de los clientes (nombre, apellidos, teléfono, NIF, dirección, correo).
- **usuarios**: Gestiona los usuarios de la aplicación, con campos para nombre, email, rol y UID de autenticación.
- **reparaciones**: Registra las reparaciones realizadas, asociando cada una a un cliente mediante 'idcliente'. Incluye artículo, descripción, fechas, precio y observaciones.
- **productos**: Catálogo de productos disponibles para la venta, con información como nombre, descripción, precio, IVA, stock, categoría y código.
- **categorías_productos**: Define las categorías a las que puede pertenecer un producto.
- **ventas**: Recoge cada venta realizada, asociando la venta a un cliente ('cliente_id'), con fecha, totales, método de pago, estado y notas.
- **detalles_venta**: Representa las líneas de cada venta (productos vendidos en cada ticket), incluyendo cantidad, precio unitario, IVA, subtotal, nombre y código del producto. Se relaciona con la tabla 'ventas' mediante 'venta_id'.
- **configuracion_empresa**: Permite personalizar el nombre, logo y mensaje de la empresa.

• Relación productos-ventas

No existe una clave foránea directa entre 'detalles_venta' y 'productos'. En su lugar, los datos del producto (nombre y código) se almacenan en cada línea de venta para mantener el histórico, aunque el producto cambie o se elimine posteriormente. Esto asegura la integridad de los tickets y facturas antiguas.

• Otras consideraciones

- **Storage**: Se utiliza el almacenamiento de Supabase para guardar los logos de empresa y los PDFs generados (informes, tickets, etc.).
- **Seguridad**: Todas las tablas relevantes tienen políticas RLS (Row Level Security) activas para garantizar que cada usuario solo accede a sus propios datos.

- Esquema BBDD

