

Prompt para Lovable — Portal de clientes

Pega esto como primer mensaje en Lovable. Reemplaza todo lo que está entre `[corchetes]` antes de enviar.

Quiero que construyas un portal de clientes llamado "**[NOMBRE_DE_TU_EMPRESA] · Mi Caso**", para que cada cliente pueda ver en línea el avance de su `[caso/trámite/pedido/proceso]` sin tener que preguntar "¿cómo va mi caso?" por WhatsApp. Usa React + Tailwind en el frontend y Supabase como backend (auth, base de datos y storage para documentos).

Objetivo

Reducir a cero las consultas repetidas de estado por WhatsApp, dándole a cada cliente una línea de tiempo clara de su caso, acceso a sus documentos, y notificaciones automáticas cuando algo cambia.

Roles y permisos

- **Cliente:** inicia sesión (por correo o número de teléfono) y ve únicamente su(s) propio(s) caso(s): línea de tiempo de hitos, documentos, y estado actual. No puede ver casos de otros clientes.
- **Gestor / Administrador:** ve y gestiona todos los casos, actualiza hitos, sube documentos, y dispara notificaciones.

Usa Supabase Auth y Row Level Security para que un cliente solo pueda leer las filas donde `cliente_id` sea su propio usuario.

Modelo de datos

Tabla **clientes**:

- `id`, `nombre`, `correo`, `telefono`

Tabla **casos**:

- `id` (uuid, PK)
- `cliente_id` (FK)
- `tipo` (enum: `[TIPO_DE_CASO_1]`, `[TIPO_DE_CASO_2]`)
- `estado_general` (enum: `en_curso`, `en_espera`, `completado`, `cancelado`)

- `fecha_inicio` (timestamp)
- `gestor_asignado_id` (FK a `usuarios`)

Tabla `hitos` (los pasos de la línea de tiempo):

- `id, caso_id` (FK)
- `nombre` (ej. "[HITO_1: Documentos recibidos]", "[HITO_2: En revisión]", "[HITO_3: Aprobado]")
- `orden` (número, para ordenar la línea de tiempo)
- `fecha_estimada` (fecha, opcional)
- `fecha_completado` (timestamp, nullable — si es null, el hito no ha ocurrido)
- `descripcion` (texto, visible para el cliente)

Tabla `documentos`:

- `id, caso_id` (FK), `nombre, url` (Supabase Storage), `subido_por` (`cliente` | `gestor`), `fecha_subida`

Tabla `notificaciones_enviadas` (log, no editable desde la UI):

- `id, caso_id, canal` (`correo` | `whatsapp`), `mensaje, fecha_envio`

Pantallas

1. **Login del cliente** — por correo o número de teléfono (magic link o código, para que no tengan que recordar contraseña).
2. **Mi caso** (vista del cliente) — línea de tiempo vertical con los hitos, marcando cuáles ya ocurrieron (con fecha) y cuál es el siguiente pendiente. Estado general destacado arriba. Sección de documentos descargables. Si el cliente puede subir documentos, un botón de carga.
3. **Panel del gestor** — lista de todos los casos con filtros por estado y gestor asignado. Buscador por nombre de cliente.
4. **Detalle de caso** (vista del gestor) — editar hitos (marcar como completado, lo que dispara la notificación automática), subir documentos, cambiar estado general, ver historial de notificaciones enviadas.
5. **Gestión de clientes** (admin) — alta de nuevos clientes y sus casos.

Reglas de negocio

- Cuando un gestor marca un hito como completado (`fecha_completado` se llena), el sistema debe registrar una notificación pendiente en `notificaciones_enviadas` y — si tienes conectado n8n o una función de Supabase Edge — disparar el envío real por correo o WhatsApp al cliente.

- Un cliente nunca ve casos ni datos de otro cliente (Row Level Security estricta).
- El estado general del caso se puede calcular automáticamente a partir del último hito completado, o gestionarse manualmente por el gestor — decide cuál prefieres y dilo explícito en el prompt real.
- Diseño cálido y tranquilizador para el cliente (esto reduce ansiedad de "no sé qué está pasando"), y denso/funcional para la vista del gestor.

Extensión opcional (pídesela a Lovable después del primer MVP)

- Integración con n8n: cuando se completa un hito, un webhook dispara el envío real de WhatsApp/correo al cliente.
- Calificación de satisfacción del cliente al cerrar el caso.
- Chat simple dentro del portal para preguntas puntuales, en vez de WhatsApp.