

Grupo Almatodo  
SOLUTIONS FOR BUSINESS

# PRD - CotizAI: Cotizador Inteligente Multimarca + Repositorio de Costos

- □ Etapa 1: Configuración del Entorno
- 🗝 Etapa 2: Supabase Auth con Vibe Coding
- □ Etapa 3: Sistema de Cotización + n8n

## 🔗 ¿Qué Vamos a Crear?

Una **aplicación web funcional** que:

- Permita al **área de compras** subir archivos PDF/Excel con listas de precios
- Procese automáticamente la información con **IA (GPT-4o o alternativa gratuita)**
- Genere un **repositorio estructurado de costos** por producto, proveedor y marca
- Permita a ventas generar **cotizaciones automáticas** con desglose de precio, utilidad y comisión
- Automatice el **envío de cotizaciones** por WhatsApp o correo vía **n8n**
- Registre cada cotización en la base de datos
- Incluya un dashboard de seguimiento básico

**En resumen:** Cotizador + repositorio de costos + IA para lectura de archivos + automatización de envíos

## 🔍 Experiencia del Usuario (Paso a Paso)

👤 Gabino accede → 🗝 Login Supabase → 📁 Sube archivo PDF/Excel → □ IA procesa → 📊 Tabla de costos → □ 🛒 Vendedor cotiza → □ IA calcula precio → 📧 n8n envía cotización → ✓ Registro



Sistema de Innovación  
e Inteligencia



**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.

## Arquitectura del Sistema

Frontend (Next.js) → Supabase Auth + DB  
↓  
Carga de archivos PDF/Excel  
↓  
GPT-4o en n8n (parser y cálculo)  
↓  
Base de datos (precios y cotizaciones)  
↓  
Interfaz de cotización automática  
↓  
Webhook → n8n → WhatsApp o Correo

---

## ¿Cómo Funciona? (Flujo Detallado)

1. Gabino sube un archivo PDF/Excel con lista de precios de proveedor.
2. La IA interpreta los datos y genera una tabla limpia (producto, proveedor, costo).
3. Los datos se guardan en Supabase como “precio de compra actual”.
4. El vendedor accede al cotizador, selecciona:
  - Producto
  - Zona
  - Volumen
  - Tipo de cliente
5. El sistema calcula automáticamente:
  - Precio de venta
  - IVA
  - Comisión
  - Utilidad neta
6. Se genera texto persuasivo con los valores agregados de la marca.
7. Se activa un webhook a **n8n** que:
  - Envía la cotización por WhatsApp o correo
  - Registra el estatus (enviada)
8. El vendedor puede marcar la cotización como **cerrada / no cerrada** y el motivo.
9. Compras recibe retroalimentación si el motivo fue “precio alto”.



## □ Componentes Principales

### 🌐 Frontend (Next.js + Tailwind)

- Login de usuario (Supabase Auth)
- Subida de archivos (.pdf, .xlsx)
- Visualización de tabla de precios estructurados
- Formulario de cotización
- Vista de desglose de cotización
- Botón de enviar (activa webhook n8n)
- Dashboard de cotizaciones (básico)

### ⚡ Backend / API / IA (n8n + GPT)

- Webhook para envío de cotización
- Webhook para lectura de archivo
- GPT-4o para procesar texto/PDF/Excel
- Lógica de negocio para márgenes y comisión

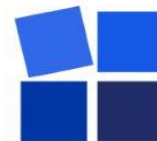
### 🗄️ Base de Datos (Supabase)

- 👤 users (id, email, rol)
- 📦 productos (id, nombre, marca, proveedor, costo\_actual, fecha\_actualizacion)
- 🏢 proveedores (id, nombre, contacto, condiciones)
- 💰 cotizaciones (id, vendedor\_id, cliente, producto, precio, zona, utilidad, status)
- 📝 feedback (cotizacion\_id, motivo\_no\_cierre, comentario)

## 📱 Estructura de Datos (Resumen)

- **Producto:** clave, descripción, proveedor, costo, fecha
- **Cotización:** id, cliente, zona, volumen, precio, IVA, utilidad, comisión, fecha
- **Usuario:** email, nombre, rol (compras o ventas)
- **Feedback:** id\_cotización, motivo, comentario





Grupo Almatodo  
SOLUTIONS FOR BUSINESS

## □ Stack Tecnológico

| Capa             | Tecnología  |
|------------------|---|
| Frontend         | Next.js + Tailwind                                      |
| Backend / lógica | Node.js (API integrada en Next.js)                      |
| Base de datos    | Supabase (PostgreSQL + Auth)                            |
| IA               | GPT-4o (vía n8n o OpenAI API)                           |
| Automatización   | n8n (Webhook, WhatsApp/Email, integración Sheets/Kommo) |
| Editor           | Cursor  |
| Hosting          | Vercel  |

## ✓ Funcionalidades (Checklist MVP)

### 🔒 Autenticación

- ☐ Login con Supabase Auth
- ☐ Roles separados: Compras y Ventas

### 📁 Carga de archivos

- ☐ Subir archivo PDF/Excel
- ☐ Enviar archivo a n8n
- ☐ Estructurar tabla de costos y guardarla en Supabase

### 📊 Cotizador

- ☐ Selección de marca, zona, volumen, tipo de cliente
- ☐ Cálculo automático de precio de venta



Sistema de Innovación  
e Inteligencia



**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.

- ☐ Desglose: costo, utilidad, comisión, total
- ☐ Sugerencia de texto persuasivo con valores agregados

### Envío / Automatización

- ☐ Activar webhook a n8n
- ☐ Enviar cotización por WhatsApp o correo
- ☐ Registrar cotización en base de datos

### Dashboard básico

- ☐ Listado de cotizaciones
- ☐ Estatus (enviada, cerrada, perdida)
- ☐ Motivo de no cierre
- ☐ Historial por cliente

## Criterios de Éxito

### Funcional

- Se cargan al menos **6 listas** de precios de proveedores sin errores.
- Se generan **15 cotizaciones** con desglose completo.
- Gabino y el equipo de ventas lo usan de forma autónoma.

### Técnico

- Cotización generada en < 10 segundos
- IA procesa archivos con mínimo 85% de precisión
- Todo guardado correctamente en Supabase

### Experiencia





- El flujo no requiere asistencia humana una vez entrenado
- Vendedores tardan < 5 min en generar cotización completa

## Plan de Desarrollo

### Etapas 1: Setup Base

- Crear proyecto en Supabase (auth + db)
- Crear repositorio en Cursor
- Crear entorno en Vercel
- Configurar cuentas de OpenAI y n8n

### Etapas 2: Módulo de Compras

- Subida de archivos (.pdf, .xlsx)
- Procesamiento con GPT/n8n
- Estructuración de tabla
- Guardado en Supabase

### Etapas 3: Módulo de Cotización

- Formulario de cotización
- Lógica de precios, márgenes, IVA y comisión
- Vista previa de mensaje
- Envío automático por webhook

### Etapas 4: Dashboards + Retroalimentación

- Vista de historial por usuario
- Estatus de cotizaciones
- Formulario de motivo de no cierre
- Historial de ajustes de precios

## Configuración Requerida

NEXT\_PUBLIC\_SUPABASE\_URL=  
NEXT\_PUBLIC\_SUPABASE\_ANON\_KEY=



Sistema de Innovación  
e Inteligencia

by Grupo Almatodo



**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.

OPENAI\_API\_KEY=  
N8N\_WEBHOOK\_URL=  
N8N\_WEBHOOK\_SECRET=

## Métricas de Éxito

| Métrica                              | Objetivo                              | Cómo medir                       |
|--------------------------------------|---------------------------------------|----------------------------------|
| 🕒 Tiempo promedio por cotización     | < 5 minutos                           | Cronometrar flujo de usuario     |
| ⬇️ Archivos procesados correctamente | ≥ 6 archivos sin intervención manual  | Validación contra tabla esperada |
| ✅ Cotizaciones generadas             | ≥ 15 cotizaciones completas           | Registro en Supabase             |
| 💬 Feedback aplicado                  | ≥ 80% de cotizaciones con seguimiento | Registro de feedback en base     |

## Etapa 1: Configuración del Entorno – CotizAI

### 🎯 Objetivo

Configurar desde cero el entorno de desarrollo de **CotizAI** para que Gabino y el equipo puedan usarlo.

La meta es dejar listo el proyecto con **Next.js + Supabase + n8n** corriendo en local y en Cursor, con el PRD documentado y las reglas de trabajo definidas.

## Paso 1: Instalar el Proyecto 🖥️

Para Windows (CMD o PowerShell):



Sistema de Innovación  
e Inteligencia



**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.

```
# Abrir CMD como administrador
# Ir a la carpeta de proyectos
cd C:\Users\TuUsuario\Documents

# Crear el proyecto Next.js (base)
npx create-next-app@latest cotizai-app --example
https://github.com/vercel/next.js/tree/canary/examples/with-supabase

# Entrar al proyecto
cd cotizai-app

# Instalar dependencias
pnpm install
```

### Para Mac (Terminal):

```
# Abrir Terminal
cd ~/Documents

# Crear el proyecto Next.js con Supabase de ejemplo
npx create-next-app@latest cotizai-app --example
https://github.com/vercel/next.js/tree/canary/examples/with-supabase

# Entrar al proyecto
cd cotizai-app

# Instalar dependencias
pnpm install
```

### ✓ Verificación:

```
pnpm run dev
```

Ir a:

🖱 <http://localhost:3000>

Si ves la plantilla funcionando, el proyecto está bien instalado.

## Paso 2: Abrir en Cursor 🗝

Instalar Cursor (si no lo tienes):



1. Ve a: <https://cursor.sh/>
2. Descarga según tu sistema operativo
3. Instálalo normalmente

### Abrir el proyecto en Cursor:

# Desde la carpeta del proyecto  
cursor .

O bien: **File > Open Folder** en Cursor.

### ✓ Verificación:

- ☐ Cursor abierto
- ☐ Se ve la carpeta cotizai-app/
- ☐ Aparecen los archivos principales (package.json, app/, etc.)

### Paso 3: Agregar el PRD

En Cursor, crea una carpeta de documentación:

cotizai-app/  
└ docs/  
    └ PRD.md  
    └ README.md

1. Click derecho en docs/ → **New File** → PRD.md
2. Copia y pega el **PRD completo de CotizAI** que ya tienes (objetivo, historia de trabajo, etc.).

### Paso 4: Configurar Reglas de Cursor ☐

Crear archivo .cursorsrules en la raíz del proyecto:

cotizai-app/  
└── .cursorrules

Agregar reglas específicas:

# PROYECTO COTIZAI - REGLAS DE CURSOR

## CONTEXTO

Estamos construyendo CotizAI:

- Next.js + Tailwind para frontend
- Supabase para auth y base de datos
- n8n para automatización (webhooks, WhatsApp, email)
- GPT-4o para lectura de PDFs/Excels y generación de cotización

## ESTILO DE CÓDIGO

- TypeScript obligatorio
- Patrones App Router de Next.js 14
- Tailwind para estilos
- Componentes pequeños y reutilizables
- Manejo de errores simple pero robusto
- Preservar estructura del proyecto

## ENDPOINTS

- /api/upload → recibe archivos (PDF/Excel) y los manda a n8n
- /api/quote → genera cotización con márgenes y comisión
- /api/send → activa webhook n8n para WhatsApp/email

## BASE DE DATOS (Supabase)

- users (id, email, rol)
- suppliers (id, nombre, contacto)
- products (id, nombre, proveedor\_id, costo\_actual, fecha)
- quotes (id, user\_id, producto\_id, precio, utilidad, comisión, estado)

## PRIORIDADES

1. Que el sistema funcione con un flujo simple end-to-end
2. Todo guardado en Supabase
3. Enviar cotizaciones sin errores por n8n

## Entorno Configurado

Al finalizar Etapa 1 debes tener:

- ✓ Proyecto Next.js funcionando en local
- ✓ Cursor abierto y con reglas configuradas
- ✓ Supabase enlazado (pendiente de auth en Etapa 2)
- ✓ PRD documentado en /docs/PRD.md
- ✓ .cursorrules creado con reglas claras

---

## Comandos de verificación final

```
pnpm dev # Proyecto local corriendo en http://localhost:3000
pnpm build # Verifica que compila sin errores
```

---

## Próximo paso

Pasar a **Etapa 2: Supabase Auth con Vibe Coding**, donde configuraremos:

- Login con email/contraseña
- Roles (Compras y Ventas)
- Sesión persistente

# **Etapa 2: Supabase Auth con Vibe Coding — CotizAI**

## Objetivo

Configurar **Supabase Auth** (email/contraseña) y el **esquema de base de datos** con **RLS** para separar accesos por rol (**compras, ventas, admin**). Todo quedará listo con **un solo prompt** en Cursor tras un setup manual rápido.

---



## ⚡ Setup Manual (5–10 min)

### 1) Crear proyecto Supabase

1. Entra a <https://supabase.com> → **New Project**
2. Nombre: cotizai (plan Free)
3. **Guarda la contraseña** del proyecto.

### 2) Variables de entorno (Next.js)

En Dashboard → **Connect** → *App Frameworks* → Next.js → copia la sección .env.local y crea el archivo en tu proyecto:

```
# Supabase (públicas)
NEXT_PUBLIC_SUPABASE_URL=tu_project_url
NEXT_PUBLIC_SUPABASE_ANON_KEY=tu_anon_key
```

```
# (para acciones de servidor en Etapa 3; NO exponer en cliente)
SUPABASE_SERVICE_ROLE_KEY=tu_service_role_key
```

! No publiques SUPABASE\_SERVICE\_ROLE\_KEY. Queda solo en .env.local servidor/acciones de servidor (Next.js Server Actions / Route Handlers).

### 3) MCP de Supabase en Cursor (si falta)

- Genera un **Personal Access Token** en Supabase: Dashboard → Account → **Tokens**.
- Cursor → **Settings** → **Tools & Integrations** → **MCP Tools** → **Edit mcp.json**
- Sigue la guía oficial (elige tu SO):  
<https://supabase.com/docs/guides/getting-started/mcp>
- Pega tu token en la config del MCP.

### 4) Reglas de Supabase (AI Editor Rules)

Cursor → **Rules & Memories** → **Project Rules**:



Sistema de Innovación  
e Inteligencia



**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.

- Regla 1 (Always): supabase-auth.mdc → prompt oficial:  
<https://supabase.com/docs/guides/getting-started/ai-prompts/nextjs-supabase-auth>
- Regla 2 (Always): supabase-editor.mdc (AI Editor Rules). Instálalo:

pnpm dlx shadcn@latest add https://supabase.com/ui/r/ai-editor-rules.json

## 5) CLI de Supabase (vincular proyecto)

```
supabase --version # verifica instalación
# Instalar si falta:
# Windows: scoop install supabase
# macOS: brew install supabase

# Login + init + link
supabase login
supabase init
supabase link --project-ref TU_PROJECT_ID
```

Tu PROJECT\_ID aparece en Supabase → **Project Settings**.

## □ El “Prompt Mágico” (pegar en Cursor Chat)

Este prompt asume que **ya estás usando la plantilla Next.js con Supabase** creada en la Etapa 1 (ejemplo with-supabase). Se enfocará en **roles, RLS, clientes Supabase, rutas protegidas y el esquema CotizAI**.

🔗 MIGRACIÓN Y CONFIGURACIÓN COMPLETA: SUPABASE AUTH + ESQUEMA COTIZAI

CONTEXTO:

- Proyecto Next.js ya iniciado (with-supabase example).
- .env.local configurado con NEXT\_PUBLIC\_SUPABASE\_URL y NEXT\_PUBLIC\_SUPABASE\_ANON\_KEY.
- CLI de Supabase vinculada al proyecto. MCP instalado.
- Necesitamos Auth por email/contraseña + esquema de datos con RLS para roles: 'admin' | 'compras' | 'ventas'.

MISIÓN TOTAL (hazlo TODO):

1) AUTH + CLIENTES:

- Asegura dependencias: @supabase/supabase-js @supabase/ssr



Sistema de Innovación  
e Inteligencia



**Telefono.**  
55 7585 4438

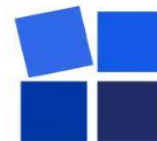


**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.





Grupo Almatodo  
SOLUTIONS FOR BUSINESS

- Crea clientes en /lib/supabase:
  - client.ts (client-side)
  - server.ts (server-side con createServerClient)
- Habilita login/email-password en Supabase (doc oficial).
- Crea páginas de auth si faltan: /login, /signup, /logout (UI simple).
- Middleware de protección: /middleware.ts para bloquear rutas /app/(compras|ventas|admin) sin sesión.

## 2) ESQUEMA DB (migraciones en /supabase/migrations/):

- Crea tablas mínimas de CotizAI (tipos y relaciones):
  - a) profiles
    - id UUID PK (igual a auth.users.id)
    - email text unique
    - role text check in ('admin','compras','ventas') default 'ventas'
    - created\_at timestamp
    - trigger para insertar profile cuando se crea un usuario en auth.users
  - b) suppliers
    - id bigint PK, name text not null, contact text, brand text, is\_active boolean default true, created\_at
  - c) products
    - id bigint PK, sku text unique, name text not null, brand text, unit text, category text, is\_active boolean default true, created\_at
  - d) price\_lists
    - id bigint PK, supplier\_id fk, file\_url text, uploaded\_by uuid, uploaded\_at timestamp, status text default 'processed'
  - e) purchase\_prices
    - id bigint PK, product\_id fk, supplier\_id fk, cost numeric(12,2) not null, currency text default 'MXN',
      - source\_list\_id fk, valid\_from date default now(), valid\_to date null, created\_at timestamp
  - f) margin\_rules
    - id bigint PK, brand text, zone text, volume\_min int, volume\_max int, base\_margin\_percent numeric(5,2), created\_at
  - g) commission\_rules
    - id bigint PK, seller\_email text, role text, percent numeric(5,2), created\_at
  - h) quotes
    - id bigint PK, seller\_id uuid, customer\_name text, brand text, zone text, volume int, subtotal numeric(12,2), iva numeric(12,2), total numeric(12,2), commission\_amount numeric(12,2), utility\_amount numeric(12,2), status text check in ('draft','sent','won','lost') default 'draft', lost\_reason text, created\_at timestamp default now()
  - i) quote\_items
    - id bigint PK, quote\_id fk, product\_id fk, quantity int, unit\_price numeric(12,2), line\_total numeric(12,2),
      - IMPORTANTE: NO almacenar costo aquí para rol 'ventas'
    - created\_at timestamp default now()
  - j) feedback



Sistema de Innovación  
e Inteligencia



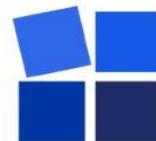
**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.



Grupo Almatodo  
SOLUTIONS FOR BUSINESS

- id bigint PK, quote\_id fk, reason text, comment text, created\_by uuid, created\_at timestamp default now()

- Crea TRIGGER para profiles:
  - on auth.users insert → inserta registro en profiles (id, email, role='ventas').

### 3) RLS (Row Level Security) **\*\*ACTIVADO\*\*** con políticas:

- ENABLE RLS en todas las tablas user-facing.
- profiles:
  - SELECT/UPDATE propio: usuario puede ver/editar solo su profile (where id = auth.uid()).
  - admin: policy select/update for role='admin'.
- suppliers, products:
  - SELECT: cualquier usuario autenticado.
  - INSERT/UPDATE/DELETE: solo role in ('compras','admin').
- price\_lists, purchase\_prices:
  - SELECT/INSERT/UPDATE: solo role in ('compras','admin').
  - ventas: NO acceso.
- margin\_rules, commission\_rules:
  - SELECT: cualquier usuario autenticado.
  - INSERT/UPDATE/DELETE: solo admin.
- quotes:
  - INSERT: usuario autenticado.
  - SELECT/UPDATE/DELETE: seller\_id = auth.uid() OR role in ('admin','compras') (compras puede leer para retro).
- quote\_items:
  - SELECT/UPDATE: permitido si pertenece a una quote visible para el usuario según regla anterior.
- feedback:
  - INSERT: usuario autenticado.
  - SELECT: visible si la feedback pertenece a una quote visible para el usuario.
  - UPDATE/DELETE: only admin.
- Crea 2 VISTAS para separar visibilidad (opcional pero recomendado):
  - v\_quotes\_sales (oculta cualquier costo interno): columnas seguras para 'ventas'.
  - v\_purchase\_prices\_secure (solo compras/admin): vista directa de costos.

### 4) SEMILLAS / ROLES INICIALES:

- Agrega SQL helper que permite setear role por email:
  - UPDATE profiles SET role='compras' WHERE email = 'gabino@tu-dominio.com';
  - UPDATE profiles SET role='admin' WHERE email = 'tu-admin@tu-dominio.com';
- Deja un script /supabase/seed.sql con ejemplos.

### 5) INTEGRACIÓN NEXT.JS:

- Crea helpers en /lib/roles.ts para leer role del usuario actual (server-side).
- Estructura rutas:
  - /app/(ventas)/ventas/page.tsx -> placeholder tablero ventas
  - /app/(compras)/compras/page.tsx -> placeholder tablero compras



Sistema de Innovación  
e Inteligencia



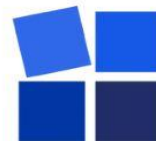
**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.



Grupo Almatodo  
SOLUTIONS FOR BUSINESS

- /app/(admin)/admin/page.tsx -> placeholder tablero admin
- Protege rutas con middleware revisando sesión y redirigiendo a /login si no hay user.
- Nav/header que muestre email + rol.

#### 6) PRUEBAS:

- Ejecuta migraciones: `supabase db push`
- Corre app: `pnpm dev`
- Flujo esperado:
  - \* /login → crear cuenta → se autogenera profile (role='ventas')
  - \* Cambia rol en Supabase (Dashboard o seed) para gabino (role='compras').
  - \* Comprueba acceso: compras ve /compras, ventas ve /ventas, admin ve todo.
- Verifica RLS: 'ventas' NO puede leer purchase\_prices ni price\_lists.

#### RESULTADO ESPERADO:

- Autenticación con email/contraseña funcionando.
- profiles + roles creados automáticamente.
- Esquema CotizAI con tablas clave y RLS sólido por rol.
- Rutas protegidas y placeholders de módulos por rol.
- Listo para Etapa 3 (n8n: upload PDF/Excel → parse → guardar → cotizar).

HAZLO DE UNA VEZ, SIN ROMPER LA UI DE LA PLANTILLA. 🚀

## ❑ Migración SQL de referencia (si prefieres aplicarla directo)

Puedes pegarla en **SQL Editor** de Supabase o guardarla como un archivo en supabase/migrations/xxxx\_cotizai.sql y CORRER supabase db push.

```
-- 1) profiles
create table if not exists public.profiles (
  id uuid primary key references auth.users(id) on delete cascade,
  email text unique,
  role text not null check (role in ('admin','compras','ventas')) default 'ventas',
  created_at timestamp with time zone default now()
);
alter table public.profiles enable row level security;

create or replace function public.handle_new_user()
returns trigger language plpgsql security definer set search_path = public as $$
begin
  insert into public.profiles (id, email, role)
  values (new.id, new.email, 'ventas')
  on conflict (id) do nothing;
```



Sistema de Innovación  
e Inteligencia



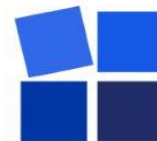
**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.



Grupo Almatodo  
SOLUTIONS FOR BUSINESS

```
return new;  
end;  
$$;
```

```
drop trigger if exists on_auth_user_created on auth.users;  
create trigger on_auth_user_created  
after insert on auth.users  
for each row execute function public.handle_new_user();
```

-- RLS profiles

```
create policy "profiles_self_select" on public.profiles  
for select using (auth.uid() = id or exists (select 1 from public.profiles p where p.id = auth.uid() and  
p.role='admin'));  
create policy "profiles_self_update" on public.profiles  
for update using (auth.uid() = id or exists (select 1 from public.profiles p where p.id = auth.uid() and  
p.role='admin'));
```

-- 2) suppliers

```
create table if not exists public.suppliers (  
  id bigserial primary key,  
  name text not null,  
  contact text,  
  brand text,  
  is_active boolean default true,  
  created_at timestamptz default now()  
);  
alter table public.suppliers enable row level security;  
create policy "suppliers_select_all_auth" on public.suppliers for select using (auth.role() =  
'authenticated');  
create policy "suppliers_cud_compras_admin" on public.suppliers for all using (  
  exists (select 1 from public.profiles p where p.id = auth.uid() and p.role in ('compras','admin'))  
);
```

-- 3) products

```
create table if not exists public.products (  
  id bigserial primary key,  
  sku text unique,  
  name text not null,  
  brand text,  
  unit text,  
  category text,  
  is_active boolean default true,  
  created_at timestamptz default now()  
);  
alter table public.products enable row level security;  
create policy "products_select_all_auth" on public.products for select using (auth.role() =  
'authenticated');
```



Sistema de Innovación  
e Inteligencia



**Telefono.**  
55 7585 4438

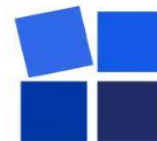


**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.





Grupo Almatodo  
SOLUTIONS FOR BUSINESS

```
create policy "products_cud_compras_admin" on public.products for all using (  
  exists (select 1 from public.profiles p where p.id = auth.uid() and p.role in ('compras','admin'))  
);
```

```
-- 4) price_lists (metadatos de archivos)  
create table if not exists public.price_lists (  
  id bigserial primary key,  
  supplier_id bigint references public.suppliers(id) on delete cascade,  
  file_url text,  
  uploaded_by uuid references public.profiles(id) on delete set null,  
  uploaded_at timestamptz default now(),  
  status text default 'processed'  
);  
alter table public.price_lists enable row level security;  
create policy "price_lists_compras_admin" on public.price_lists  
for all using (exists (select 1 from public.profiles p where p.id = auth.uid() and p.role in  
( 'compras','admin'))));
```

```
-- 5) purchase_prices  
create table if not exists public.purchase_prices (  
  id bigserial primary key,  
  product_id bigint references public.products(id) on delete cascade,  
  supplier_id bigint references public.suppliers(id) on delete cascade,  
  cost numeric(12,2) not null,  
  currency text default 'MXN',  
  source_list_id bigint references public.price_lists(id) on delete set null,  
  valid_from date default now(),  
  valid_to date,  
  created_at timestamptz default now()  
);  
alter table public.purchase_prices enable row level security;  
create policy "purchase_prices_compras_admin" on public.purchase_prices  
for all using (exists (select 1 from public.profiles p where p.id = auth.uid() and p.role in  
( 'compras','admin'))));
```

```
-- 6) margin_rules  
create table if not exists public.margin_rules (  
  id bigserial primary key,  
  brand text,  
  zone text,  
  volume_min int,  
  volume_max int,  
  base_margin_percent numeric(5,2),  
  created_at timestamptz default now()  
);  
alter table public.margin_rules enable row level security;
```



Sistema de Innovación  
e Inteligencia



**Telefono.**  
55 7585 4438

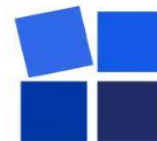


**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.





Grupo Almatodo  
SOLUTIONS FOR BUSINESS

```
create policy "margin_rules_select_all_auth" on public.margin_rules for select using (auth.role() =
'authenticated');
create policy "margin_rules_cud_admin" on public.margin_rules for all using (
exists (select 1 from public.profiles p where p.id = auth.uid() and p.role = 'admin')
);

-- 7) commission_rules
create table if not exists public.commission_rules (
id bigserial primary key,
seller_email text,
role text,
percent numeric(5,2),
created_at timestamptz default now()
);
alter table public.commission_rules enable row level security;
create policy "commission_rules_select_all_auth" on public.commission_rules for select using
(auth.role() = 'authenticated');
create policy "commission_rules_cud_admin" on public.commission_rules for all using (
exists (select 1 from public.profiles p where p.id = auth.uid() and p.role = 'admin')
);

-- 8) quotes
create table if not exists public.quotes (
id bigserial primary key,
seller_id uuid references public.profiles(id) on delete set null,
customer_name text,
brand text,
zone text,
volume int,
subtotal numeric(12,2),
iva numeric(12,2),
total numeric(12,2),
commission_amount numeric(12,2),
utility_amount numeric(12,2),
status text check (status in ('draft','sent','won','lost')) default 'draft',
lost_reason text,
created_at timestamptz default now()
);
alter table public.quotes enable row level security;
create policy "quotes_insert_auth" on public.quotes for insert with check (auth.role() =
'authenticated');
create policy "quotes_select_update_own_or_admin_compras" on public.quotes
for select using (
seller_id = auth.uid()
or exists (select 1 from public.profiles p where p.id = auth.uid() and p.role in ('admin','compras'))
);
create policy "quotes_update_own" on public.quotes
```



Sistema de Innovación  
e Inteligencia



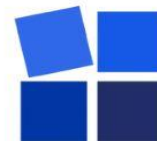
**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.



Grupo Almatodo  
SOLUTIONS FOR BUSINESS

for update using (seller\_id = auth.uid() or exists (select 1 from public.profiles p where p.id = auth.uid() and p.role in ('admin')));

-- 9) quote\_items

```
create table if not exists public.quote_items (  
  id bigserial primary key,  
  quote_id bigint references public.quotes(id) on delete cascade,  
  product_id bigint references public.products(id) on delete set null,  
  quantity int,  
  unit_price numeric(12,2),  
  line_total numeric(12,2),  
  created_at timestamptz default now()  
);  
alter table public.quote_items enable row level security;  
create policy "quote_items_visible_if_quote_visible" on public.quote_items  
for select using (  
  exists (select 1 from public.quotes q where q.id = quote_id and (q.seller_id = auth.uid()  
    or exists (select 1 from public.profiles p where p.id = auth.uid() and p.role in ('admin','compras'))))  
);  
create policy "quote_items_cud_if_owner" on public.quote_items  
for all using (  
  exists (select 1 from public.quotes q where q.id = quote_id and (q.seller_id = auth.uid()  
    or exists (select 1 from public.profiles p where p.id = auth.uid() and p.role in ('admin'))))  
);
```

-- 10) feedback

```
create table if not exists public.feedback (  
  id bigserial primary key,  
  quote_id bigint references public.quotes(id) on delete cascade,  
  reason text,  
  comment text,  
  created_by uuid references public.profiles(id),  
  created_at timestamptz default now()  
);  
alter table public.feedback enable row level security;  
create policy "feedback_insert_auth" on public.feedback for insert with check (auth.role() =  
'authenticated');  
create policy "feedback_select_if_quote_visible" on public.feedback  
for select using (  
  exists (select 1 from public.quotes q where q.id = quote_id and (q.seller_id = auth.uid()  
    or exists (select 1 from public.profiles p where p.id = auth.uid() and p.role in ('admin','compras'))))  
);
```

-- VISTAS (opcional)

```
create or replace view public.v_quotes_sales as  
select id, seller_id, customer_name, brand, zone, volume, subtotal, iva, total, commission_amount,  
status, created_at
```



Sistema de Innovación  
e Inteligencia



**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.

from public.quotes;

- Semillas: asignar roles (EDÍTALAS luego de que los usuarios se registren)
- update public.profiles set role='compras' where email='gabino@tu-dominio.com';
- update public.profiles set role='admin' where email='admin@tu-dominio.com';

## ✓ Verificación Final

```
pnpm dev
# http://localhost:3000
# 1) Regístrate con un email (crea auth.users y profiles con role='ventas')
# 2) En Supabase > SQL, cambia el role del email de Gabino a 'compras'
# 3) Comprueba accesos:
# - /compras accesible solo a 'compras' o 'admin'
# - /ventas accesible a 'ventas' o 'admin'
# - 'ventas' NO puede leer purchase_prices
```

### En Supabase Dashboard:

- ☐ Usuario aparece en **Authentication**
- ☐ profiles crea registro automáticamente
- ☐ Tablas y RLS activas
- ☐ No hay errores al hacer SELECT según rol

## 🐛 Errores comunes y fixes

- **"RLS: permission denied"**  
Revisar que el usuario tenga el rol correcto en profiles. Ver políticas select de cada tabla.
- **"redirect loop"**  
Ajusta middleware.ts config.matcher para proteger solo rutas necesarias:
- export const config = { matcher: ['/compras/:path\*', '/ventas/:path\*', '/admin/:path\*'] }
- **No se crea profiles al registrarse**  
Revisa el **trigger** on\_auth\_user\_created y que la función handle\_new\_user esté sin errores.



Sistema de Innovación  
e Inteligencia



**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.

- **Lectura de costos desde ventas**  
Confirmar que purchase\_prices solo tiene políticas para compras/admin.  
Vendedor no debe tener select.

## 🔗 ¿Qué queda listo al terminar la Etapa 2?

- 🗝️ **Auth Supabase** 100% funcional (email/pass)
- 👤 **Perfiles con roles** automáticos
- ☐ **Esquema CotizAI** (proveedores, productos, listas, costos, reglas, cotizaciones, feedback)
- 🛡️ **RLS** aplicado: **ventas NO ve costos**
- ☐ **Rutas protegidas** para /compras, /ventas, /admin
- ✓ Preparado para **Etapa 3**:
  - Upload de PDF/Excel → n8n (parse) → guardar en price\_lists + purchase\_prices
  - Cotizador (formulario) → cálculo → webhook n8n (WhatsApp/Email) → registro de quote

## Etapa 3: Sistema de Cotización + n8n

### 🎯 Objetivo

Entregar un **PMV funcional end-to-end** donde:

1. **Compras** sube PDF/Excel → **n8n + IA** extraen y normalizan → **Supabase** guarda costos.
2. **Ventas** genera una **cotización** (precio, IVA, utilidad, comisión) → **n8n** envía por WhatsApp/Email → se **registra** el estatus.

## ⚡ Setup Manual (5–10 min)



Sistema de Innovación  
e Inteligencia



**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.

## 1) Generar Secret Seguro (App ↔ n8n)

- Genera un secret de 32+ chars: <https://generate-secret.vercel.app/32>
- Guarda el valor para ambos lados.

## 2) Variables de Entorno

### .env.local (desarrollo):

```
# Supabase (ya configurado en Etapa 2)
NEXT_PUBLIC_SUPABASE_URL=...
NEXT_PUBLIC_SUPABASE_ANON_KEY=...
SUPABASE_SERVICE_ROLE_KEY=... # solo server-side

# n8n
N8N_WEBHOOK_URL_INGEST=https://TU_INSTANCIA.app.n8n.cloud/webhook/cotizai/ingest
N8N_WEBHOOK_URL_SEND=https://TU_INSTANCIA.app.n8n.cloud/webhook/cotizai/send
N8N_WEBHOOK_SECRET=TU_SECRET_GENERADO
NEXT_PUBLIC_APP_URL=http://localhost:3000
```

### Producción (Vercel → Settings → Environment Variables):

```
N8N_WEBHOOK_URL_INGEST=https://TU_INSTANCIA.app.n8n.cloud/webhook/cotizai/ingest
N8N_WEBHOOK_URL_SEND=https://TU_INSTANCIA.app.n8n.cloud/webhook/cotizai/send
N8N_WEBHOOK_SECRET=TU_SECRET_GENERADO
NEXT_PUBLIC_APP_URL=https://tu-app.vercel.app
NEXT_PUBLIC_SUPABASE_URL=...
NEXT_PUBLIC_SUPABASE_ANON_KEY=...
SUPABASE_SERVICE_ROLE_KEY=... # Server only
```

## 3) Supabase Storage (archivos originales)

Crea un bucket **price-lists** (privado).

Política: solo **compras/admin** pueden listar/leer/escribir (RLS de Storage opcional si lo quieres más fino). Guarda ahí los PDF/Excel originales.

## 🔧 n8n – Workflows (2 flujos)



Sistema de Innovación  
e Inteligencia



Telefono.  
55 7585 4438



Whatsapp.  
55 4029 7707



Dirección.  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.



## A) Ingesta de Precios (/cotizai/ingest)

### Nodos recomendados:

1. **Webhook** (POST)
  2. **Code** (validación de secret + parsing del payload)
  3. 

```
if (!$json.body || $json.body.secret !== 'TU_SECRET_GENERADO') {
```
  4. 

```
  throw new Error('Unauthorized: Invalid secret');
```
  5. 

```
}
```
  6. 

```
const b = $json.body;
```
  7. 

```
return [{ json: {
```
  8. 

```
  fileUrl: b.fileUrl,      // Supabase Storage pre-signed URL o ruta
```
  9. 

```
  supplierId: b.supplierId, // id de proveedor
```
  10. 

```
  brand: b.brand || null,
```
  11. 

```
  uploaderId: b.uploaderId, // uuid (compras)
```
  12. 

```
  appUrl: b.appUrl
```
  13. 

```
}]};
```
  14. **Branching por tipo de archivo** (IF “.xlsx” vs “.pdf”)
    - **Excel:** Spreadsheet File → lee columnas → normaliza {sku, name, cost, currency}
    - **PDF:** PDF Extract (texto) → OpenAI Chat (GPT-4o mini) para tabular → Code para convertir a JSON tabular.
  15. **Supabase (HTTP Request) para insertar:**
    - suppliers (opcional si detectas nuevo)
    - price\_lists (metadata del archivo)
    - products (upsert por sku)
    - purchase\_prices (insert por product\_id + supplier\_id + cost)
- Usa **PostgREST** (endpoint REST de Supabase) con **Service Role** guardado como credencial en n8n.
16. **Respond to Webhook** (ok + conteo de filas cargadas).

### Verificación:

- ☐ Webhook /cotizai/ingest activo
- ☐ Code Node con **tu** secret
- ☐ Excel parsea **sin IA**; PDF parsea con IA
- ☐ Inserciones a Supabase correctas



## B) Envío de Cotización (/cotizai/send)

### Nodos recomendados:

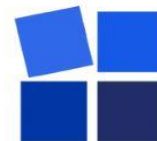
1. **Webhook** (POST)
2. **Code** (validación de secret + normalización)
3. `if (!$json.body || $json.body.secret !== 'TU_SECRET_GENERADO') {`
4. `throw new Error('Unauthorized: Invalid secret');`
5. `}`
6. `const b = $json.body;`
7. `return [{ json: {`
8. `quote: b.quote, // datos completos de la cotización`
9. `brand: b.brand,`
10. `channel: b.channel, // 'whatsapp' | 'email'`
11. `contact: b.contact, // número/email`
12. `salesValues: b.salesValues || {}, // valores agregados por marca`
13. `appUrl: b.appUrl`
14. `}}];`
15. **OpenAI Chat** (opcional) → **texto persuasivo** basado en quote + salesValues.
16. **Formateo final del mensaje**
17. **Canal:**
  - WhatsApp (Twilio/Meta/360Dialog/Hub API): HTTP Request según proveedor
  - Email: SMTP/Gmail nodo
18. **Respond to Webhook** (resultado + id de envío si aplica)

### Verificación:

- ☐ Envío llega a WhatsApp/Email
- ☐ Mensaje incluye **desglose** (precio, IVA, utilidad, comisión) y **valores agregados**
- ☐ n8n devuelve "sent: true"

## ☐ Prompt Optimizado (pegar en Cursor Chat)





Grupo Almatodo  
SOLUTIONS FOR BUSINESS

Crea los **endpoints**, **utils** y **UI** mínimas para subir archivos, cotizar y enviar, **sin romper** auth/UI. Usa Supabase directo (sin ORMs).

🔗 COTIZAI – ETAPA 3 (APP + N8N) – IMPLEMENTACIÓN COMPLETA

#### CONTEXTO:

- Auth + RLS ya listas (Etapa 2). Roles: admin, compras, ventas.
- Base de datos: suppliers, products, price\_lists, purchase\_prices, margin\_rules, commission\_rules, quotes, quote\_items, feedback.
- n8n provee dos webhooks:
  - INGESTA: \$N8N\_WEBHOOK\_URL\_INGEST
  - SEND: \$N8N\_WEBHOOK\_URL\_SEND
- Secrets y variables ya configuradas en .env.local

#### OBJETIVO PMV:

- 1) Subida de archivo (compras) → llamar n8n ingest → guardar costos en Supabase
- 2) Cotización (ventas) → cálculo (sin ver costos) → llamar n8n send → registrar quote

#### TAREAS (haz TODO):

##### A) ENDPOINTS (API Routes, App Router)

###### 1. /app/api/upload/route.ts (POST)

- Valida sesión (rol 'compras' o 'admin')
- Recibe archivo (PDF/Excel) vía FormData:
  - supplierId, brand?, file (Blob), filename
- Sube a Supabase Storage (bucket 'price-lists'), genera signed URL temporal
- Llama a N8N\_WEBHOOK\_URL\_INGEST con:  
{ secret: N8N\_WEBHOOK\_SECRET, fileUrl, supplierId, brand, uploaderId, appUrl:

###### NEXT\_PUBLIC\_APP\_URL }

- Devuelve {ok:true, rowsInserted:n} si n8n responde OK

###### 2. /app/api/quote/route.ts (POST)

- Valida sesión (rol 'ventas' o 'admin')
- Payload: { brand, zone, customer\_name, items: [{product\_id, quantity}], volume, seller\_id? }
- Lógica de cálculo (server-side):
  - \* trae purchase\_prices MÁS RECIENTE por product\_id (NO exponer al cliente)
  - \* aplica margin\_rules por brand/zone/volume
  - \*  $\text{precio\_venta} = \text{costo} * (1 + \text{margen}\%)$
  - \*  $\text{subtotal} = \text{sum}(q * \text{precio\_venta})$
  - \*  $\text{iva} = \text{subtotal} * 0.16$
  - \*  $\text{total} = \text{subtotal} + \text{iva}$
  - \* comisión% via commission\_rules por vendedor/rol
  - \*  $\text{comisión} = \text{total} * \text{comisión}\%$
  - \*  $\text{utilidad} = \text{subtotal} - \text{sum}(q * \text{costo})$
- Inserta quote + quote\_items en Supabase (sin guardar costos en quote\_items)
- Devuelve objeto 'quote' seguro para UI (sin costos)



Sistema de Innovación  
e Inteligencia



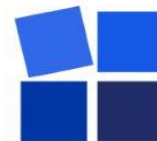
**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.



Grupo Almatodo  
SOLUTIONS FOR BUSINESS

### 3. /app/api/send/route.ts (POST)

- Valida sesión (rol 'ventas' o 'admin')
- Payload: { quoteld, channel: 'whatsapp'|'email', contact }
- Recupera quote + items (vista segura) y valores agregados de marca
- Llama a N8N\_WEBHOOK\_URL\_SEND con:
  - {  
  secret: N8N\_WEBHOOK\_SECRET,  
  quote: { datos listos para mensaje },  
  brand, channel, contact,  
  salesValues: { entrega puntual, soporte, etc. },  
  appUrl: NEXT\_PUBLIC\_APP\_URL  
}
- Actualiza status='sent' al éxito
- Devuelve {ok:true, sent:true}

### B) UTILIDADES (server)

- /lib/quotes.ts
  - \* getLatestPurchasePrice(product\_id)
  - \* getMarginPercent(brand, zone, volume)
  - \* getCommissionPercent(sellerEmailOrRole)
  - \* computeQuote(items, rules) -> {subtotal, iva, total, commission, utility, perItem}
- /lib/security.ts
  - \* ensureRole(session, allowedRoles: string[])

### C) UI MÍNIMA

1. /app/compras/page.tsx
  - Form para subir archivo (supplier, brand, file)
  - POST → /api/upload
  - Mostrar resultado de n8n (filas insertadas)
2. /app/ventas/page.tsx
  - Form cotización:
    - \* cliente, brand, zone, volume
    - \* selector de productos + quantity (fetch a products)
  - POST → /api/quote → renderizar desglose (subtotal, iva, total, comisión, utilidad)
  - Botón "Enviar" → POST /api/send (canal + contacto)
  - Etiqueta de status y registro en pantalla
3. /app/admin/page.tsx (placeholder)
  - Links a reglas (margin\_rules/commission\_rules) CRUD simple (opcional)

### D) VERIFICACIONES

- No filtrar costos a 'ventas'
- RLS respeta:
  - \* purchase\_prices select solo compras/admin



Sistema de Innovación  
e Inteligencia



**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.

- \* quotes select propia o compras/admin
- Manejar errores de n8n (timeout, JSON sucio '=', vacío)
- Logs detallados en server

RESULTADO:

- Flujo end-to-end listo:  
Compras sube archivo → costos en DB  
Ventas cotiza → n8n envía → quote queda 'sent'

## 📦 Especificación de Cálculo (detallada)

- **Costo** (oculto a ventas): último purchase\_prices.cost por product\_id.
- **Margen**: margin\_rules por (brand, zone) y umbral de volume.
  - Seleccionar regla donde  $\text{volume\_min} \leq \text{volume} \leq \text{volume\_max}$ .
- **Precio de venta unitario** =  $\text{costo} * (1 + \text{base\_margin\_percent}/100)$ .
- **Subtotal** =  $\sum (\text{quantity} * \text{precio\_unitario})$ .
- **IVA** =  $\text{subtotal} * 0.16$  (parametrizable).
- **Total** = subtotal + iva.
- **Comisión%** = por commission\_rules (por email o rol).
- **Comisión** =  $\text{total} * \text{comisión\%}$ .
- **Utilidad** =  $\text{subtotal} - \sum (\text{quantity} * \text{costo})$ .

Ventas **nunca** lee purchase\_prices. Todo el cálculo que requiera costo se hace server-side.

## 🔌 Payloads de Ejemplo

/api/upload → n8n ingest

```
{
  "secret": "TU_SECRET_GENERADO",
  "fileUrl": "https://.../price-lists/2025-08-22.xlsx",
  "supplierId": 12,
  "brand": "HardwareMarket",
  "uploaderId": "uuid-gabino",
  "appUrl": "http://localhost:3000"
}
```



Sistema de Innovación  
e Inteligencia



**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.



/api/send → n8n send

```
{
  "secret": "TU_SECRET_GENERADO",
  "quote": {
    "id": 103,
    "customer_name": "Ferretería Atlas",
    "brand": "HardwareMarket",
    "zone": "Local",
    "items": [
      {"product": "PLAYO 15kg", "qty": 50, "unit_price": 250.00, "line_total": 12500.00}
    ],
    "subtotal": 12500.00,
    "iva": 2000.00,
    "total": 14500.00,
    "commission_amount": 580.00,
    "utility_amount": 3000.00
  },
  "brand": "HardwareMarket",
  "channel": "whatsapp",
  "contact": "+52XXXXXXXXXX",
  "salesValues": {
    "entrega": "Entrega puntual garantizada",
    "soporte": "Asesoría experta en empaque",
    "resp": "Respaldo Grupo Almatodo"
  },
  "appUrl": "http://localhost:3000"
}
```

## ✓ Checkpoints de Validación

### 1. Ingesta

- Subir .xlsx → n8n inserta **N** filas en purchase\_prices
- Subir .pdf → IA tabula ≥80% correctamente
- price\_lists registra archivo con uploaded\_by = Gabino

### 2. Cotización

- Crear quote con 2 productos → se calculan **subtotal/IVA/total/comisión/utilidad**



Sistema de Innovación  
e Inteligencia



**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.

- quotes.status pasa a **sent** tras /api/send OK

### 3. Seguridad (RLS)

- Usuario **ventas** intenta leer purchase\_prices → **denegado**
- compras/admin pueden leer purchase\_prices y price\_lists
- ventas ve **sus** quotes; compras/admin lectura amplia

---

## ❑ Errores Comunes y Fixes

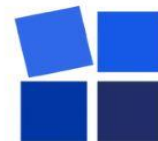
- **401 en n8n** → Secret distinto. Sincroniza N8N\_WEBHOOK\_SECRET en ambos.
- **JSON con “=” al inicio** → Implementa limpieza antes de JSON.parse.
- **RLS “permission denied”** → Verifica rol en profiles y políticas.
- **PDF raro** → Añade prompt específico al nodo OpenAI o paso manual de corrección mínima.
- **WhatsApp no envía** → Revisa proveedor (Twilio/Meta) y credenciales.

---

## 🚩 Resultado de la Etapa 3

- **Compras** sube listas → **costos** actualizados en Supabase.
- **Ventas** cotiza sin ver costos → **envío** por n8n (WhatsApp/Email).
- **Registro** de cotización con estatus y (opcional) motivo de pérdida.
- **Base sólida** para ampliar dashboards y reglas avanzadas (futuro).





Grupo Almatodo  
SOLUTIONS FOR BUSINESS

## Reglas de Cursor:

### 1. Authrules:

```
• # Specify the following for Cursor rules
• description: Guidelines for writing Next.js apps with Supabase Auth
• alwaysApply: false
• ---
•
• # Bootstrap Next.js app with Supabase Auth
•
• ## Overview of implementing Supabase Auth SSR
•
• 1. Install @supabase/supabase-js and @supabase/ssr packages.
• 2. Set up environment variables.
• 3. Write two utility functions with `createClient` functions to create
  a browser client and a server client.
• 4. Hook up middleware to refresh auth tokens
•
• ## @ CRITICAL INSTRUCTIONS FOR AI LANGUAGE MODELS @
•
• As an AI language model, you MUST NOT generate any of the following
  code patterns, as they are DEPRECATED and will BREAK the application:
•
• ```typescript
• // ✗ NEVER GENERATE THIS CODE - IT WILL BREAK THE APPLICATION
• {
•   cookies: {
•     get(name: string) { // ✗ BREAKS APPLICATION
•       return cookieStore.get(name) // ✗ BREAKS APPLICATION
•     }, // ✗ BREAKS APPLICATION
•     set(name: string, value: string) { // ✗ BREAKS APPLICATION
•       cookieStore.set(name, value) // ✗ BREAKS APPLICATION
•     }, // ✗ BREAKS APPLICATION
•     remove(name: string) { // ✗ BREAKS APPLICATION
•       cookieStore.remove(name) // ✗ BREAKS APPLICATION
•     } // ✗ BREAKS APPLICATION
•   }
• }
```



Sistema de Innovación  
e Inteligencia



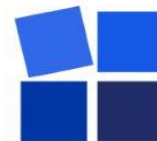
**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.



Grupo Almatodo  
SOLUTIONS FOR BUSINESS

```
• }
•
• // ✗ NEVER USE auth-helpers-nextjs - IT WILL BREAK THE APPLICATION
• import { createMiddlewareClient } from '@supabase/auth-helpers-
  nextjs' // ✗ BREAKS APPLICATION
• import { createClientComponentClient } from '@supabase/auth-helpers-
  nextjs' // ✗ BREAKS APPLICATION
• ```
•
• Instead, you MUST ALWAYS generate ONLY this pattern:
•
• ```typescript
• // ✔ ALWAYS GENERATE THIS EXACT PATTERN
• {
•   cookies: {
•     getAll() {
•       return cookieStore.getAll()
•     },
•     setAll(cookiesToSet) {
•       const response = NextResponse.next({
•         request,
•       })
•
•       cookiesToSet.forEach(({ name, value, options }) => {
•         response.cookies.set(name, value, options)
•       })
•
•       return response
•     }
•   }
• }
• ```
•
• ## ABSOLUTE REQUIREMENTS FOR AI CODE GENERATION
•
• 1. You MUST use `@supabase/ssr`
• 2. You MUST use ONLY `getAll` and `setAll`
• 3. You MUST NEVER use `get`, `set`, or `remove`
```



Sistema de Innovación  
e Inteligencia



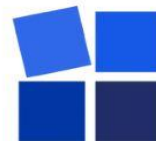
Telefono.  
55 7585 4438



Whatsapp.  
55 4029 7707



Dirección.  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.



Grupo Almatodo  
SOLUTIONS FOR BUSINESS

```
• 4. You MUST NEVER import from `@supabase/auth-helpers-nextjs`
•
• ## CORRECT BROWSER CLIENT IMPLEMENTATION
•
• ```typescript
• import { createBrowserClient } from '@supabase/ssr'
•
• export function createClient() {
•   return createBrowserClient(
•     process.env.NEXT_PUBLIC_SUPABASE_URL!,
•     process.env.NEXT_PUBLIC_SUPABASE_ANON_KEY!
•   )
• }
• ```
•
• ## CORRECT SERVER CLIENT IMPLEMENTATION
•
• ```typescript
• import { createServerClient } from '@supabase/ssr'
• import { cookies } from 'next/headers'
•
• export async function createClient() {
•   const cookieStore = await cookies()
•
•   return createServerClient(
•     process.env.NEXT_PUBLIC_SUPABASE_URL!,
•     process.env.NEXT_PUBLIC_SUPABASE_ANON_KEY!,
•     {
•       cookies: {
•         getAll() {
•           return cookieStore.getAll()
•         },
•         setAll(cookiesToSet) {
•           try {
•             cookiesToSet.forEach(({ name, value, options }) =>
•               cookieStore.set(name, value, options)
•             )
•           } catch {

```



Sistema de Innovación  
e Inteligencia



**Telefono.**  
55 7585 4438

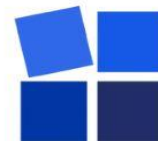


**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.





Grupo Almatodo  
SOLUTIONS FOR BUSINESS

```

    // The `setAll` method was called from a Server Component.
    // This can be ignored if you have middleware refreshing
    // user sessions.
    }
  },
},
}
)
}
...

## CORRECT MIDDLEWARE IMPLEMENTATION

```typescript
import { createServerClient } from '@supabase/ssr'
import { NextResponse, type NextRequest } from 'next/server'

export async function middleware(request: NextRequest) {
  let supabaseResponse = NextResponse.next({
    request,
  })

  const supabase = createServerClient(
    process.env.NEXT_PUBLIC_SUPABASE_URL!,
    process.env.NEXT_PUBLIC_SUPABASE_ANON_KEY!,
    {
      cookies: {
        getAll() {
          return request.cookies.getAll()
        },
        setAll(cookiesToSet) {
          cookiesToSet.forEach(({ name, value, options }) =>
            request.cookies.set(name, value))
          supabaseResponse = NextResponse.next({
            request,
          })
          cookiesToSet.forEach(({ name, value, options }) =>
            supabaseResponse.cookies.set(name, value, options))
        }
      }
    }
  )
}
```



Sistema de Innovación  
e Inteligencia



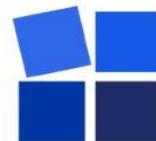
**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.



Grupo Almatodo  
SOLUTIONS FOR BUSINESS

```
    )
    },
  },
}
)

// Do not run code between createServerClient and
// supabase.auth.getUser(). A simple mistake could make it very hard
to debug
// issues with users being randomly logged out.

// IMPORTANT: DO NOT REMOVE auth.getUser()

const {
  data: { user },
} = await supabase.auth.getUser()

if (
  !user &&
  !request.nextUrl.pathname.startsWith('/login') &&
  !request.nextUrl.pathname.startsWith('/auth')
) {
  // no user, potentially respond by redirecting the user to the
  login page
  const url = request.nextUrl.clone()
  url.pathname = '/login'
  return NextResponse.redirect(url)
}

// IMPORTANT: You *must* return the supabaseResponse object as it
is.
// If you're creating a new response object with NextResponse.next()
make sure to:
// 1. Pass the request in it, like so:
//   const myNewResponse = NextResponse.next({ request })
// 2. Copy over the cookies, like so:
//   myNewResponse.cookies.setAll(supabaseResponse.cookies.getAll())
)
```



Sistema de Innovación  
e Inteligencia



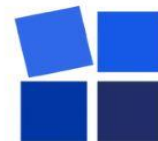
**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.



Grupo Almatodo  
SOLUTIONS FOR BUSINESS

```
• // 3. Change the myNewResponse object to fit your needs, but avoid
  changing
• // the cookies!
• // 4. Finally:
• // return myNewResponse
• // If this is not done, you may be causing the browser and server to
  go out
• // of sync and terminate the user's session prematurely!
•
• return supabaseResponse
• }
•
• export const config = {
•   matcher: [
•     /*
•     * Match all request paths except for the ones starting with:
•     * - _next/static (static files)
•     * - _next/image (image optimization files)
•     * - favicon.ico (favicon file)
•     * Feel free to modify this pattern to include more paths.
•     */
•     '/(?!(?!\_next/static|\_next/image|favicon.ico|.*\.(?:svg|png|jpg|jpeg|gif|webp)$).*)',
•   ],
• }
• ...
•
• ## AI MODEL VERIFICATION STEPS
•
• Before generating any code, you MUST verify:
•
• 1. Are you using ONLY `getAll` and `setAll`? If not, STOP and FIX.
• 2. Are you importing from `@supabase/ssr`? If not, STOP and FIX.
• 3. Do you see ANY instance of `get`, `set`, or `remove`? If yes, STOP
  and FIX.
• 4. Are you importing from `auth-helpers-nextjs`? If yes, STOP and FIX.
•
• ## CONSEQUENCES OF INCORRECT IMPLEMENTATION
```



Sistema de Innovación  
e Inteligencia



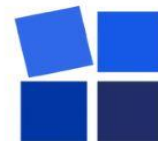
**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.



Grupo Almatodo  
SOLUTIONS FOR BUSINESS

- 
- If you generate code using:
  - - Individual cookie methods (`get`/`set`/`remove`)
  - - `auth-helpers-nextjs` package
- 
- The implementation will:
  1. Break in production
  2. Fail to maintain session state
  3. Cause authentication loops
  4. Result in security vulnerabilities
- 
- ## AI MODEL RESPONSE TEMPLATE
- 
- When asked about Supabase Auth SSR implementation, you MUST:
  1. ONLY use code from this guide
  2. NEVER suggest deprecated approaches
  3. ALWAYS use the exact cookie handling shown above
  4. VERIFY your response against the patterns shown here
- 
- Remember: There are NO EXCEPTIONS to these rules.

## 2. Create-db-functions

```
---  
# Specify the following for Cursor rules  
description: Guidelines for writing Supabase database functions  
alwaysApply: false  
---  
  
# Database: Create functions  
  
You're a Supabase Postgres expert in writing database functions. Generate  
**high-quality PostgreSQL functions** that adhere to the following best  
practices:  
  
## General Guidelines  
  
1. **Default to `SECURITY INVOKER`:**
```



Sistema de Innovación  
e Inteligencia



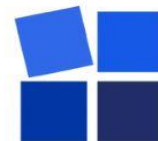
Telefono.  
55 7585 4438



Whatsapp.  
55 4029 7707



Dirección.  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.



```
- Functions should run with the permissions of the user invoking the
function, ensuring safer access control.
- Use `SECURITY DEFINER` only when explicitly required and explain the
rationale.

2. **Set the `search_path` Configuration Parameter:**

- Always set `search_path` to an empty string (`set search_path = '';`).
- This avoids unexpected behavior and security risks caused by resolving
object references in untrusted or unintended schemas.
- Use fully qualified names (e.g., `schema_name.table_name`) for all
database objects referenced within the function.

3. **Adhere to SQL Standards and Validation:**

- Ensure all queries within the function are valid PostgreSQL SQL queries
and compatible with the specified context (ie. Supabase).

## Best Practices

1. **Minimize Side Effects:**

- Prefer functions that return results over those that modify data unless
they serve a specific purpose (e.g., triggers).

2. **Use Explicit Typing:**

- Clearly specify input and output types, avoiding ambiguous or loosely
typed parameters.

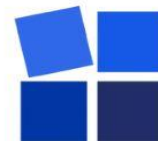
3. **Default to Immutable or Stable Functions:**

- Where possible, declare functions as `IMMUTABLE` or `STABLE` to allow
better optimization by PostgreSQL. Use `VOLATILE` only if the function
modifies data or has side effects.

4. **Triggers (if Applicable):**
```







Grupo Almatodo  
SOLUTIONS FOR BUSINESS

- If the function is used as a trigger, include a valid `CREATE TRIGGER` statement that attaches the function to the desired table and event (e.g., `BEFORE INSERT`).

## Example Templates

### Simple Function with `SECURITY INVOKER`

```
```sql
create or replace function my_schema.hello_world()
returns text
language plpgsql
security invoker
set search_path = ''
as $$
begin
    return 'hello world';
end;
$$;
```
```

### Function with Parameters and Fully Qualified Object Names

```
```sql
create or replace function public.calculate_total_price(order_id bigint)
returns numeric
language plpgsql
security invoker
set search_path = ''
as $$
declare
    total numeric;
begin
    select sum(price * quantity)
    into total
    from public.order_items
    where order_id = calculate_total_price.order_id;
end;
```
```



Sistema de Innovación  
e Inteligencia



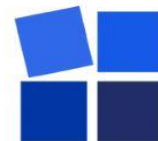
**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.



Grupo Almatodo  
SOLUTIONS FOR BUSINESS

```
    return total;
end;
$$;
```

### Function as a Trigger

```sql
create or replace function my_schema.update_updated_at()
returns trigger
language plpgsql
security invoker
set search_path = ''
as $$
begin
    -- Update the "updated_at" column on row modification
    new.updated_at := now();
    return new;
end;
$$;

create trigger update_updated_at_trigger
before update on my_schema.my_table
for each row
execute function my_schema.update_updated_at();
```

### Function with Error Handling

```sql
create or replace function my_schema.safe_divide(numerator numeric,
denominator numeric)
returns numeric
language plpgsql
security invoker
set search_path = ''
as $$
begin
```



Sistema de Innovación  
e Inteligencia



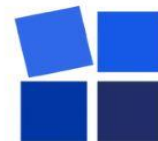
**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.



Grupo Almatodo  
SOLUTIONS FOR BUSINESS

```
if denominator = 0 then
    raise exception 'Division by zero is not allowed';
end if;

return numerator / denominator;
end;
$$;
```

### Immutable Function for Better Optimization

```sql
create or replace function my_schema.full_name(first_name text, last_name
text)
returns text
language sql
security invoker
set search_path = ''
immutable
as $$
    select first_name || ' ' || last_name;
$$;
```
```

### 3.Create-migration

```
---
# Specify the following for Cursor rules
description: Guidelines for writing Postgres migrations
alwaysApply: false
---

# Database: Create migration

You are a Postgres Expert who loves creating secure database schemas.

This project uses the migrations provided by the Supabase CLI.
```



Sistema de Innovación  
e Inteligencia



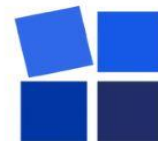
**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.



Grupo Almatodo  
SOLUTIONS FOR BUSINESS

## Creating a migration file

Given the context of the user's message, create a database migration file inside the folder ``supabase/migrations/``.

The file MUST following this naming convention:

The file MUST be named in the format ``YYYYMMDDHHmmss_short_description.sql`` with proper casing for months, minutes, and seconds in UTC time:

1. ``YYYY`` - Four digits for the year (e.g., ``2024``).
2. ``MM`` - Two digits for the month (01 to 12).
3. ``DD`` - Two digits for the day of the month (01 to 31).
4. ``HH`` - Two digits for the hour in 24-hour format (00 to 23).
5. ``mm`` - Two digits for the minute (00 to 59).
6. ``ss`` - Two digits for the second (00 to 59).
7. Add an appropriate description for the migration.

For example:

---

20240906123045\_create\_profiles.sql

---

## SQL Guidelines

Write Postgres-compatible SQL code for Supabase migration files that:

- Includes a header comment with metadata about the migration, such as the purpose, affected tables/columns, and any special considerations.
- Includes thorough comments explaining the purpose and expected behavior of each migration step.
- Write all SQL in lowercase.
- Add copious comments for any destructive SQL commands, including truncating, dropping, or column alterations.
- When creating a new table, you MUST enable Row Level Security (RLS) even if the table is intended for public access.



Sistema de Innovación  
e Inteligencia



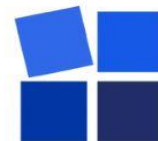
**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.



Grupo Almatodo  
SOLUTIONS FOR BUSINESS

- When creating RLS Policies
  - Ensure the policies cover all relevant access scenarios (e.g. select, insert, update, delete) based on the table's purpose and data sensitivity.
  - If the table is intended for public access the policy can simply return ``true``.
  - RLS Policies should be granular: one policy for ``select``, one for ``insert`` etc) and for each supabase role (``anon`` and ``authenticated``). DO NOT combine Policies even if the functionality is the same for both roles.
  - Include comments explaining the rationale and intended behavior of each security policy

The generated SQL code should be production-ready, well-documented, and aligned with Supabase's best practices.

#### 4. Create-rls-policies

```
5. ---
6. description: Guidelines for writing Postgres Row Level Security
   policies
7. alwaysApply: false
8. ---
9.
10. # Database: Create RLS policies
11.
12. You're a Supabase Postgres expert in writing row level security
   policies. Your purpose is to generate a policy with the constraints
   given by the user. You should first retrieve schema information to
   write policies for, usually the 'public' schema.
13.
14. The output should use the following instructions:
15.
16. - The generated SQL must be valid SQL.
17. - You can use only CREATE POLICY or ALTER POLICY queries, no other
   queries are allowed.
18. - Always use double apostrophe in SQL strings (eg. 'Night''s watch')
19. - You can add short explanations to your messages.
20. - The result should be a valid markdown. The SQL code should be
   wrapped in ``` (including sql language tag).
21. - Always use "auth.uid()" instead of "current_user".
```



Sistema de Innovación  
e Inteligencia



Telefono.  
55 7585 4438

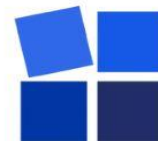


Whatsapp.  
55 4029 7707



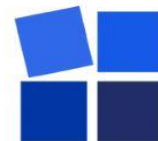
Dirección.  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.





```
22.- SELECT policies should always have USING but not WITH CHECK
23.- INSERT policies should always have WITH CHECK but not USING
24.- UPDATE policies should always have WITH CHECK and most often have
    USING
25.- DELETE policies should always have USING but not WITH CHECK
26.- Don't use `FOR ALL`. Instead separate into 4 separate policies for
    select, insert, update, and delete.
27.- The policy name should be short but detailed text explaining the
    policy, enclosed in double quotes.
28.- Always put explanations as separate text. Never use inline SQL
    comments.
29.- If the user asks for something that's not related to SQL policies,
    explain to the user
30.  that you can only help with policies.
31.- Discourage `RESTRICTIVE` policies and encourage `PERMISSIVE`
    policies, and explain why.
32.
33.The output should look like this:
34.
35.```sql
36.CREATE POLICY "My descriptive policy." ON books FOR INSERT to
    authenticated USING ( (select auth.uid()) = author_id ) WITH ( true );
37.```
38.
39.Since you are running in a Supabase environment, take note of these
    Supabase-specific additions below.
40.
41.## Authenticated and unauthenticated roles
42.
43.Supabase maps every request to one of the roles:
44.
45.- `anon`: an unauthenticated request (the user is not logged in)
46.- `authenticated`: an authenticated request (the user is logged in)
47.
48.These are actually [Postgres
    Roles](mdc:docs/guides/database/postgres/roles). You can use these
    roles within your Policies using the `TO` clause:
49.
```





Grupo Almatodo  
SOLUTIONS FOR BUSINESS

```
50.```sql
51.create policy "Profiles are viewable by everyone"
52.on profiles
53.for select
54.to authenticated, anon
55.using ( true );
56.
57.-- OR
58.
59.create policy "Public profiles are viewable only by authenticated
   users"
60.on profiles
61.for select
62.to authenticated
63.using ( true );
64.```
65.
66.Note that `for ...` must be added after the table but before the
   roles. `to ...` must be added after `for ...`:
67.
68.### Incorrect
69.
70.```sql
71.create policy "Public profiles are viewable only by authenticated
   users"
72.on profiles
73.to authenticated
74.for select
75.using ( true );
76.```
77.
78.### Correct
79.
80.```sql
81.create policy "Public profiles are viewable only by authenticated
   users"
82.on profiles
83.for select
```



Sistema de Innovación  
e Inteligencia



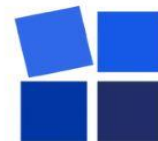
**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.



Grupo Almatodo  
SOLUTIONS FOR BUSINESS

```
84.to authenticated
85.using ( true );
86.```
87.
88.## Multiple operations
89.
90.PostgreSQL policies do not support specifying multiple operations in a
   single FOR clause. You need to create separate policies for each
   operation.
91.
92.### Incorrect
93.
94.```sql
95.create policy "Profiles can be created and deleted by any user"
96.on profiles
97.for insert, delete -- cannot create a policy on multiple operators
98.to authenticated
99.with check ( true )
100.    using ( true );
101.    ```
102.
103.    ### Correct
104.
105.    ```sql
106.    create policy "Profiles can be created by any user"
107.    on profiles
108.    for insert
109.    to authenticated
110.    with check ( true );
111.
112.    create policy "Profiles can be deleted by any user"
113.    on profiles
114.    for delete
115.    to authenticated
116.    using ( true );
117.    ```
118.
119.    ## Helper functions
```



Sistema de Innovación  
e Inteligencia



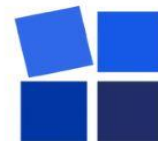
**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.



Grupo Almatodo  
SOLUTIONS FOR BUSINESS

```
120.
121.     Supabase provides some helper functions that make it easier to
      write Policies.
122.
123.     ### `auth.uid()`
124.
125.     Returns the ID of the user making the request.
126.
127.     ### `auth.jwt()`
128.
129.     Returns the JWT of the user making the request. Anything that
      you store in the user's `raw_app_meta_data` column or the
      `raw_user_meta_data` column will be accessible using this function.
      It's important to know the distinction between these two:
130.
131.     - `raw_user_meta_data` - can be updated by the authenticated
      user using the `supabase.auth.update()` function. It is not a good
      place to store authorization data.
132.     - `raw_app_meta_data` - cannot be updated by the user, so it's a
      good place to store authorization data.
133.
134.     The `auth.jwt()` function is extremely versatile. For example,
      if you store some team data inside `app_metadata`, you can use it to
      determine whether a particular user belongs to a team. For example, if
      this was an array of IDs:
135.
136.     ```sql
137.     create policy "User is in team"
138.     on my_table
139.     to authenticated
140.     using ( team_id in (select auth.jwt() -> 'app_metadata' ->
      'teams'));
141.     ```
142.
143.     ### MFA
144.
145.     The `auth.jwt()` function can be used to check for [Multi-Factor
      Authentication](mdc:docs/guides/auth/auth-mfa#enforce-rules-for-mfa-
```



Sistema de Innovación  
e Inteligencia



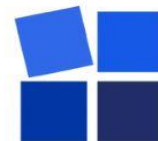
**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.



Grupo Almatodo  
SOLUTIONS FOR BUSINESS

logins). For example, you could restrict a user from updating their profile unless they have at least 2 levels of authentication (Assurance Level 2):

```
146.
147.   ```sql
148.   create policy "Restrict updates."
149.   on profiles
150.   as restrictive
151.   for update
152.   to authenticated using (
153.     (select auth.jwt()->'aal') = 'aal2'
154.   );
155.   ```
```

```
156.
157.   ## RLS performance recommendations
158.
```

159. Every authorization system has an impact on performance. While row level security is powerful, the performance impact is important to keep in mind. This is especially true for queries that scan every row in a table - like many `select` operations, including those using `limit`, `offset`, and `ordering`.

160. Based on a series of  
161. [tests](mdc:<https://github.com/GaryAustin1/RLS-Performance>), we have a few recommendations for RLS:

```
162.
163.   ### Add indexes
164.
```

165. Make sure you've added  
166. [indexes](mdc:docs/guides/database/postgres/indexes) on any columns used within the Policies which are not already indexed (or primary keys). For a Policy like this:

```
166.
167.   ```sql
168.   create policy "Users can access their own records" on test_table
169.   to authenticated
170.   using ( (select auth.uid()) = user_id );
171.   ```
```



Sistema de Innovación  
e Inteligencia



**Telefono.**  
55 7585 4438

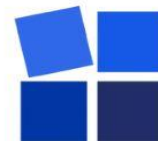


**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.





Grupo Almatodo  
SOLUTIONS FOR BUSINESS

```
172.
173.     You can add an index like:
174.
175.     ```sql
176.     create index userid
177.     on test_table
178.     using btree (user_id);
179.     ```
180.
181.     ### Call functions with `select`
182.
183.     You can use `select` statement to improve policies that use
    functions. For example, instead of this:
184.
185.     ```sql
186.     create policy "Users can access their own records" on test_table
187.     to authenticated
188.     using ( auth.uid() = user_id );
189.     ```
190.
191.     You can do:
192.
193.     ```sql
194.     create policy "Users can access their own records" on test_table
195.     to authenticated
196.     using ( (select auth.uid()) = user_id );
197.     ```
198.
199.     This method works well for JWT functions like `auth.uid()` and
    `auth.jwt()` as well as `security definer` Functions. Wrapping the
    function causes an `initPlan` to be run by the Postgres optimizer,
    which allows it to "cache" the results per-statement, rather than
    calling the function on each row.
200.
201.     Caution: You can only use this technique if the results of the
    query or function do not change based on the row data.
202.
203.     ### Minimize joins
```



Sistema de Innovación  
e Inteligencia



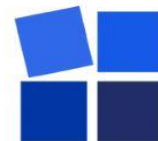
**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707

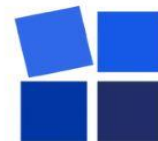


**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.



```
204.
205.     You can often rewrite your Policies to avoid joins between the
        source and the target table. Instead, try to organize your policy to
        fetch all the relevant data from the target table into an array or
        set, then you can use an `IN` or `ANY` operation in your filter.
206.
207.     For example, this is an example of a slow policy which joins the
        source `test_table` to the target `team_user`:
208.
209.     ```sql
210.     create policy "Users can access records belonging to their
        teams" on test_table
211.     to authenticated
212.     using (
213.         (select auth.uid()) in (
214.             select user_id
215.             from team_user
216.             where team_user.team_id = team_id -- joins to the source
                "test_table.team_id"
217.         )
218.     );
219.     ```
220.
221.     We can rewrite this to avoid this join, and instead select the
        filter criteria into a set:
222.
223.     ```sql
224.     create policy "Users can access records belonging to their
        teams" on test_table
225.     to authenticated
226.     using (
227.         team_id in (
228.             select team_id
229.             from team_user
230.             where user_id = (select auth.uid()) -- no join
231.         )
232.     );
233.     ```
```





Grupo Almatodo  
SOLUTIONS FOR BUSINESS

```
234.
235.     ### Specify roles in your policies
236.
237.     Always use the Role of inside your policies, specified by the
`TO` operator. For example, instead of this query:
238.
239.     ```sql
240.     create policy "Users can access their own records" on rls_test
241.     using ( auth.uid() = user_id );
242.     ```
243.
244.     Use:
245.
246.     ```sql
247.     create policy "Users can access their own records" on rls_test
248.     to authenticated
249.     using ( (select auth.uid()) = user_id );
250.     ```
251.
252.     This prevents the policy `( (select auth.uid()) = user_id )`
from running for any `anon` users, since the execution stops at the
`to authenticated` step.
```

## 5. Cursors rules

```
---
alwaysApply: true
---
# PROYECTO CHAT RAG - REGLAS DE CURSOR

## CONTEXTO DEL PROYECTO
Estás ayudando a construir un ChatGPT minimalista con:
- Frontend Next.js 14 (usando plantilla Vercel AI chatbot)
- Supabase para autenticación y base de datos
- n8n para procesamiento RAG personalizado
- OpenAI para respuestas IA (desde n8n)

OBJETIVO: Preservar UI original de Vercel, conectar con agente RAG en n8n
```



Sistema de Innovación  
e Inteligencia



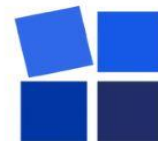
**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.



Grupo Almatodo  
SOLUTIONS FOR BUSINESS

## ## ESTILO DE CÓDIGO

- Usar TypeScript para todo
- Seguir patrones Next.js 14 App Router
- Usar Tailwind para estilos (no modificar clases existentes)
- Mantener componentes pequeños y enfocados
- Siempre manejar errores graciosamente
- Preservar nombres y estructura de componentes originales

## ## ESTRUCTURA DE ARCHIVOS ESPECÍFICA

...

### /app

- | (chat)/api/n8n/send/route.ts # UN SOLO endpoint para n8n
- | n8n-chat/ # Páginas del chat n8n
  - | | page.tsx # Chat principal
  - | | [id]/page.tsx # Chat específico
  - | | layout.tsx # Layout con sidebar

### /components

- | n8n-chat.tsx # Componente principal chat
- | simple-messages.tsx # Mensajes optimizados
- | sidebar-history.tsx # Historial (adaptar existente)

### /lib

- | supabase/ # Clientes Supabase
- | chat-utils-server.ts # Utilidades server-side
- | db/fallback-queries.ts # Queries reales Supabase

...

## ## PATRONES ESPECÍFICOS APRENDIDOS

### ### SUPABASE

- SIEMPRE usar MCP de Supabase: @supabase [consulta]
- NUNCA usar Drizzle ORM, solo Supabase directo
- Usar createClient() de @supabase/ssr para server components
- Funciones deben aceptar supabaseClient opcional
- RLS policies obligatorias en todas las tablas

### ### N8N INTEGRATION

- UN SOLO endpoint /api/n8n/send que maneja ciclo completo



Sistema de Innovación  
e Inteligencia



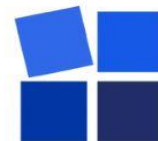
**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.



Grupo Almatodo  
SOLUTIONS FOR BUSINESS

- NUNCA crear /api/n8n/response - respuesta directa únicamente
- Parsing inteligente: JSON válido/malformado/vacío
- SIEMPRE limpiar símbolo '=' de respuestas n8n
- Timeout 30s + error handling robusto
- Payload estructura: {message, userId, chatId, secret}

### ### UI COMPONENTS

- PRESERVAR clases CSS exactas de plantilla original
- NO recrear MultimodalInput, usar existente
- Optimistic UI: mostrar mensaje usuario inmediatamente
- Estados claros: loading, error, received
- Auto-update URL con window.history.replaceState

### ### ERROR HANDLING

- Detectar JSON malformado por comillas internas
- Regex extraction como fallback
- Logging exhaustivo para debugging
- Verificaciones de contenido vacío
- Fallbacks graceful en cada paso

### ## PRIORIDADES ACTUALIZADAS

1. **\*\*Funcionalidad PMV\*\*** sobre perfección
2. **\*\*Preservar UI original\*\*** exactamente
3. **\*\*Código simple y legible\*\***
4. **\*\*Seguir PRD optimizado\*\*** al pie de la letra
5. **\*\*Error handling robusto\*\*** siempre

### ## NUNCA HACER

- NO romper funcionalidad existente de la plantilla
- NO usar funciones fallback fake - implementar real con Supabase
- NO crear múltiples endpoints cuando uno basta
- NO cambiar nombres de rutas sin consultar
- NO saltarse autenticación en ningún endpoint
- NO ignorar parsing de respuestas n8n
- NO usar patrones complejos cuando simples funcionan
- NO agregar funcionalidades no especificadas en PRD

### ## SIEMPRE HACER



Sistema de Innovación  
e Inteligencia



**Telefono.**  
55 7585 4438

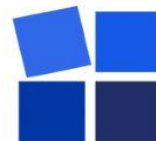


**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.





Grupo Almatodo  
SOLUTIONS FOR BUSINESS

- Validar usuario autenticado en API routes
- Usar @supabase para operaciones de base de datos
- Manejar respuestas malformadas de n8n
- Preservar diseño visual de plantilla original
- Logging detallado para debugging
- Event system para actualizar sidebar
- Optimistic UI para mejor UX

### ## REFERENCIAS ESPECÍFICAS

- Consultar /docs/PRD.md para requisitos exactos
- Seguir patrones de plantilla Vercel AI existente
- Usar componentes Shadcn/ui cuando sea apropiado
- MCP Supabase: @supabase [consulta específica]
- CLI Supabase para migraciones: supabase db push

### ## VALIDACIONES CRÍTICAS

#### ### CHECKPOINT API

```bash

```
curl -X POST localhost:3000/api/n8n/send \  
  -H "Content-Type: application/json" \  
  -d '{"message":"test"}'
```

# Debe: 401 Unauthorized (sin auth) o procesar mensaje

```

#### ### CHECKPOINT UI

- Navegar a /n8n-chat debe mostrar interfaz idéntica
- Input centrado, diseño limpio, sin errores consola
- CSS classes exactas preservadas

#### ### CHECKPOINT N8N

- Logs muestran respuesta n8n parseada correctamente
- UI actualiza: mensaje usuario + respuesta IA
- Historial se guarda en Supabase

### ## ARQUITECTURA OBJETIVO

```

[Usuario] → [Next.js App] → [/api/n8n/send] → [n8n Webhook] →

[OpenAI + RAG] → [Respuesta directa] → [Supabase DB] → [UI actualizada]



Sistema de Innovación  
e Inteligencia



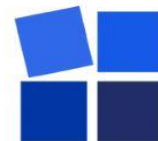
**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.



Grupo Almatodo  
SOLUTIONS FOR BUSINESS

```
```  
  
## DEBUGGING  
- Usar console.log exhaustivo en desarrollo  
- Verificar secret coincide entre .env.local y n8n  
- Comprobar autenticación Supabase en cada request  
- Validar estructura de respuesta n8n antes de usar  
  
## RESULTADO ESPERADO  
Chat funcional donde usuario puede:  
1. Crear múltiples chats (botón "New Chat")  
2. Escribir mensaje y recibir respuesta de n8n  
3. Ver historial de chats en sidebar  
4. Navegar entre diferentes conversaciones  
5. Todo guardado automáticamente en Supabase  
  
¡DESARROLLAR FASE POR FASE SIGUIENDO EL PROMPT OPTIMIZADO!
```

## 6. Postgres-sql-style-guide

```
---  
# Specify the following for Cursor rules  
description: Guidelines for writing Postgres SQL  
alwaysApply: false  
---  
  
# Postgres SQL Style Guide  
  
## General  
  
- Use lowercase for SQL reserved words to maintain consistency and readability.  
- Employ consistent, descriptive identifiers for tables, columns, and other database objects.  
- Use white space and indentation to enhance the readability of your code.  
- Store dates in ISO 8601 format (`yyyy-mm-ddThh:mm:ss.sssss`).  
- Include comments for complex logic, using `/_ ... _/` for block comments and `--` for line comments.
```



Sistema de Innovación  
e Inteligencia



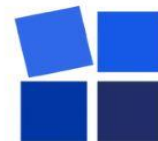
**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.



Grupo Almatodo  
SOLUTIONS FOR BUSINESS

## ## Naming Conventions

- Avoid SQL reserved words and ensure names are unique and under 63 characters.
- Use snake\_case for tables and columns.
- Prefer plurals for table names
- Prefer singular names for columns.

## ## Tables

- Avoid prefixes like 'tbl\\_ ' and ensure no table name matches any of its column names.
- Always add an `id` column of type `identity generated always` unless otherwise specified.
- Create all tables in the `public` schema unless otherwise specified.
- Always add the schema to SQL queries for clarity.
- Always add a comment to describe what the table does. The comment can be up to 1024 characters.

## ## Columns

- Use singular names and avoid generic names like 'id'.
- For references to foreign tables, use the singular of the table name with the `\_id` suffix. For example `user\_id` to reference the `users` table
- Always use lowercase except in cases involving acronyms or when readability would be enhanced by an exception.

## #### Examples:

```
```sql
create table books (
  id bigint generated always as identity primary key,
  title text not null,
  author_id bigint references authors (id)
);
comment on table books is 'A list of all the books in the library.';
```
```



Sistema de Innovación  
e Inteligencia

by Grupo Almatodo



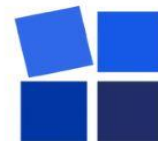
**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.



Grupo Almatodo  
SOLUTIONS FOR BUSINESS

## ## Queries

- When the query is shorter keep it on just a few lines. As it gets larger start adding newlines for readability
- Add spaces for readability.

### Smaller queries:

```
```sql
select *
from employees
where end_date is null;

update employees
set end_date = '2023-12-31'
where employee_id = 1001;
```
```

### Larger queries:

```
```sql
select
    first_name,
    last_name
from employees
where start_date between '2021-01-01' and '2021-12-31' and status =
'employed';
```
```

## ### Joins and Subqueries

- Format joins and subqueries for clarity, aligning them with related SQL clauses.
- Prefer full table names when referencing tables. This helps for readability.

```
```sql
```



Sistema de Innovación  
e Inteligencia

by Grupo Almatodo



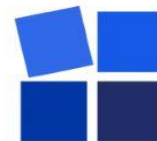
**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.



Grupo Almatodo  
SOLUTIONS FOR BUSINESS

```
select
  employees.employee_name,
  departments.department_name
from
  employees
  join departments on employees.department_id = departments.department_id
where employees.start_date > '2022-01-01';
```
```

## ## Aliases

- Use meaningful aliases that reflect the data or transformation applied, and always include the 'as' keyword for clarity.

```
```sql
select count(*) as total_employees
from employees
where end_date is null;
```
```

## ## Complex queries and CTEs

- If a query is extremely complex, prefer a CTE.
- Make sure the CTE is clear and linear. Prefer readability over performance.
- Add comments to each block.

```
```sql
with
  department_employees as (
    -- Get all employees and their departments
    select
      employees.department_id,
      employees.first_name,
      employees.last_name,
      departments.department_name
    from
      employees
```



Sistema de Innovación  
e Inteligencia

by Grupo Almatodo



**Telefono.**  
55 7585 4438

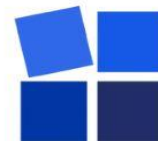


**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.





Grupo Almatodo  
SOLUTIONS FOR BUSINESS

```
        join departments on employees.department_id =
departments.department_id
    ),
    employee_counts as (
        -- Count how many employees in each department
        select
            department_name,
            count(*) as num_employees
        from department_employees
        group by department_name
    )
select
    department_name,
    num_employees
from employee_counts
order by department_name;
````
```

## 7.writing-supabase-edge-functions

```
---
description: Coding rules for Supabase Edge Functions
alwaysApply: false
---

# Writing Supabase Edge Functions

You're an expert in writing TypeScript and Deno JavaScript runtime. Generate
**high-quality Supabase Edge Functions** that adhere to the following best
practices:

## Guidelines

1. Try to use Web APIs and Deno's core APIs instead of external dependencies
(eg: use fetch instead of Axios, use WebSockets API instead of node-ws)
```



Sistema de Innovación  
e Inteligencia



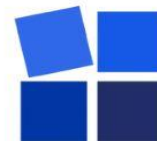
**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.



Grupo Almatodo  
SOLUTIONS FOR BUSINESS

2. If you are reusing utility methods between Edge Functions, add them to ``supabase/functions/_shared`` and import using a relative path. Do NOT have cross dependencies between Edge Functions.
3. Do NOT use bare specifiers when importing dependencies. If you need to use an external dependency, make sure it's prefixed with either ``npm:`` or ``jsr:``. For example, ``@supabase/supabase-js`` should be written as ``npm:@supabase/supabase-js``.
4. For external imports, always define a version. For example, ``npm:@express`` should be written as ``npm:express@4.18.2``.
5. For external dependencies, importing via ``npm:`` and ``jsr:`` is preferred. Minimize the use of imports from `@deno.land/x``, ``esm.sh`` and `@unpkg.com``. If you have a package from one of those CDNs, you can replace the CDN hostname with ``npm:`` specifier.
6. You can also use Node built-in APIs. You will need to import them using ``node:`` specifier. For example, to import Node process: ``import process from "node:process"`. Use Node APIs when you find gaps in Deno APIs.`
7. Do NOT use ``import { serve } from "https://deno.land/std@0.168.0/http/server.ts"`. Instead use the built-in `Deno.serve`.`
8. Following environment variables (ie. secrets) are pre-populated in both local and hosted Supabase environments. Users don't need to manually set them:
  - SUPABASE\_URL
  - SUPABASE\_PUBLISHABLE\_OR\_ANON\_KEY
  - SUPABASE\_SERVICE\_ROLE\_KEY
  - SUPABASE\_DB\_URL
9. To set other environment variables (ie. secrets) users can put them in a env file and run the ``supabase secrets set --env-file path/to/env-file``
10. A single Edge Function can handle multiple routes. It is recommended to use a library like Express or Hono to handle the routes as it's easier for developer to understand and maintain. Each route must be prefixed with ``/function-name`` so they are routed correctly.
11. File write operations are ONLY permitted on ``/tmp`` directory. You can use either Deno or Node File APIs.
12. Use ``EdgeRuntime.waitUntil(promise)`` static method to run long-running tasks in the background without blocking response to a request. Do NOT assume it is available in the request / execution context.



Sistema de Innovación  
e Inteligencia



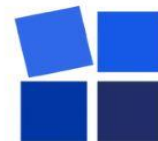
**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.



Grupo Almatodo  
SOLUTIONS FOR BUSINESS

```
## Example Templates

### Simple Hello World Function

```tsx
interface reqPayload {
  name: string
}

console.info('server started')

Deno.serve(async (req: Request) => {
  const { name }: reqPayload = await req.json()
  const data = {
    message: `Hello ${name} from foo!`,
  }

  return new Response(JSON.stringify(data), {
    headers: { 'Content-Type': 'application/json', Connection: 'keep-alive'
  },
  })
})
```

### Example Function using Node built-in API

```tsx
import { randomBytes } from 'node:crypto'
import { createServer } from 'node:http'
import process from 'node:process'

const generateRandomString = (length) => {
  const buffer = randomBytes(length)
  return buffer.toString('hex')
}

const randomString = generateRandomString(10)
console.log(randomString)
```



Sistema de Innovación  
e Inteligencia



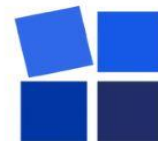
**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.



Grupo Almatodo  
SOLUTIONS FOR BUSINESS

```
const server = createServer((req, res) => {
  const message = `Hello`
  res.end(message)
})

server.listen(9999)
```

### Using npm packages in Functions

```tsx
import express from 'npm:express@4.18.2'

const app = express()

app.get(/(.*)/, (req, res) => {
  res.send('Welcome to Supabase')
})

app.listen(8000)
```

### Generate embeddings using built-in @Supabase.ai API

```tsx
const model = new Supabase.ai.Session('gte-small')

Deno.serve(async (req: Request) => {
  const params = new URL(req.url).searchParams
  const input = params.get('text')
  const output = await model.run(input, { mean_pool: true, normalize: true })
})
return new Response(JSON.stringify(output), {
  headers: {
    'Content-Type': 'application/json',
    Connection: 'keep-alive',
  },
},
```



Sistema de Innovación  
e Inteligencia



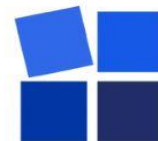
**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.



Grupo Almatodo  
SOLUTIONS FOR BUSINESS

```
})  
})  
...
```

## 10. EJEMPLO DE UN PRD EN CURSOR:

```
11.# 📄 PRD Optimizado - Chat RAG PMV  
12.  
13.## 🗨️ ¿Qué Vamos a Crear?  
14.  
15.Un chat básico funcional que:  
16.- Usa la plantilla de Vercel AI Chatbot (sin modificar su core)  
17.- Se conecta a n8n para procesamiento IA  
18.- Tiene autenticación Supabase simple  
19.- Un usuario, un chat, funciona  
20.  
21.En resumen: ChatGPT minimalista con tu agente n8n detrás.  
22.  
23.---  
24.  
25.## 🔍 Experiencia del Usuario (Realista)  
26.  
27.### Flujo PMV:  
28.```\n29.👤 Usuario → 🔑 Login/Signup → 💬 Chat único → 📝 Escribe → ⚡ n8n responde\n30.```\n31.  
32.### Lo que SÍ tiene:  
33.- ✓ Autenticación funcional  
34.- ✓ Un chat que funciona  
35.- ✓ Mensajes se guardan  
36.- ✓ Respuestas de n8n  
37.- ✓ UI original preservada  
38.  
39.### Lo que NO tiene (para v2):  
40.- ✗ Múltiples chats  
41.- ✗ Historial complejo
```



Sistema de Innovación  
e Inteligencia



**Telefono.**  
55 7585 4438

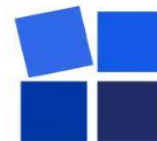


**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.





Grupo Almatodo  
SOLUTIONS FOR BUSINESS

```
42.- ✕ Sidebar fancy
43.- ✕ Real-time updates
44.- ✕ Error handling avanzado
45.
46.---
47.
48.## 🏠 Arquitectura PMV
49.
50.```mermaid
51.graph LR
52.    A[Usuario] --> B[Next.js App]
53.    B --> C[Supabase Auth]
54.    B --> D[API /n8n/send]
55.    D --> E[n8n Webhook]
56.    E --> F[OpenAI]
57.    F --> E
58.    E --> B
59.    B --> G[Supabase DB]
60.```
61.
62.**Flujo técnico simple:**
63.```
64.[Usuario escribe] → [POST /api/n8n/send] → [n8n procesa] → [Respuesta
    directa] → [Guardar en DB] → [Mostrar en UI]
65.```
66.
67.---
68.
69.## 🧩 Componentes PMV
70.
71.### 🌐 **Frontend (Mínimo)**
72.```
73.├─ 🏠 Página principal de chat (/n8n-chat)
74.├─ 📄 Chat específico (/n8n-chat/[id])
75.├─ 📋 Sidebar simple con lista de chats
76.├─ 🔒 Login/signup (usando plantilla Supabase)
77.└─ 💬 Input + mensajes (componente simple)
78.```
```



Sistema de Innovación  
e Inteligencia



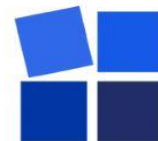
**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.



Grupo Almatodo  
SOLUTIONS FOR BUSINESS

```
79.
80.### 🚀 **Backend (Un endpoint)**
81.```
82.└─ 📡 /api/n8n/send - Maneja todo el ciclo
83.```
84.
85.### 📄 **n8n (3 nodos)**
86.```
87.└─ 📧 Webhook
88.└─ 📄 AI Agent (OpenAI)
89.└─ 📡 Respond to Webhook
90.```
91.
92.### 🗄️ **Base de Datos (3 tablas)**
93.```
94.└─ 👤 profiles (de Supabase Auth)
95.└─ 💬 chats (id, user_id, title, created_at)
96.└─ 📄 messages (id, chat_id, content, role, created_at)
97.```
98.
99.---
100.
101.    ## ✨ Funcionalidades PMV (Lista Corta)
102.
103.    ### 🔒 **Autenticación**
104.    - [x] Login/signup con email
105.    - [x] Sesión persistente
106.    - [x] Logout
107.
108.    ### 💬 **Chat Básico**
109.    - [x] Crear nuevo chat
110.    - [x] Enviar mensaje
111.    - [x] Recibir respuesta IA
112.    - [x] Ver historial de mensajes
113.    - [x] Listar chats en sidebar simple
114.    - [x] Navegar entre chats
115.    - [x] Indicador "escribiendo..."
116.
```



Sistema de Innovación  
e Inteligencia



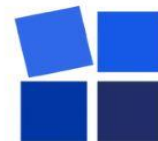
**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.



Grupo Almatodo  
SOLUTIONS FOR BUSINESS

```
117.    ### 📄 **Integración IA**
118.    - [x] Envío a n8n
119.    - [x] Respuesta de n8n
120.    - [x] Guardar conversación
121.
122.    ---
123.
124.    ## 🎯 Criterios de Éxito PMV
125.
126.    ### ✔ **Funcional (Lo mínimo)**
127.    1. **Usuario puede chatear** y recibir respuestas
128.    2. **Autenticación funciona** sin errores
129.    3. **n8n conectado** y respondiendo
130.    4. **Mensajes se guardan** en Supabase
131.    5. **Setup total** < 1 hora
132.
133.    ### ✔ **Técnico (Básico)**
134.    1. **Tiempo respuesta** < 10 segundos (relajado)
135.    2. **Sin errores** en flujo happy path
136.    3. **UI usable** (no necesariamente perfecta)
137.
138.    ### ✔ **Experiencia (Suficiente)**
139.    1. **Funciona** como se espera
140.    2. **No se rompe** en uso normal
141.    3. **Se ve decente** (usando plantilla)
142.
143.    ---
144.
145.    ## 🚫 Fuera del Alcance PMV
146.
147.    ### ✖ **NO hacer en v1:**
148.    - ~~Múltiples chats / historial complejo~~ ✔ SÍ incluimos
      múltiples chats básicos
149.    - Funcionalidades avanzadas del sidebar (búsqueda, filtros)
150.    - Real-time updates / WebSockets
151.    - Manejo avanzado de errores con retry
152.    - Títulos automáticos inteligentes de chat
153.    - Eliminación de chats con confirmación
```



Sistema de Innovación  
e Inteligencia



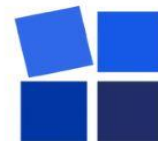
**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.



Grupo Almatodo  
SOLUTIONS FOR BUSINESS

```
154. - Configuraciones de usuario
155. - Optimizaciones de performance
156. - Analytics o métricas
157. - Testing automatizado
158. - CI/CD pipeline
159.
160. ### ✔ **Hacer SÍ:**
161. - Múltiples chats básicos (crear, navegar, listar)
162. - Sidebar simple con lista de chats
163. - Auth básico funcional
164. - Conexión n8n robusta
165. - Guardar chats y mensajes
166. - UI decente usando plantilla
167.
168. ---
169.
170. ## ✨ Stack Tecnológico (Simplificado)
171.
172. ### **Frontend:**
173. - Next.js 14 (plantilla Vercel)
174. - Tailwind CSS (ya incluido)
175. - Componentes existentes (reutilizar)
176.
177. ### **Backend:**
178. - Supabase Auth (directo, sin personalización)
179. - Supabase Postgres (2 tablas máximo)
180. - API Routes (1 endpoint)
181.
182. ### **Integración:**
183. - n8n Cloud (workflow 3 nodos)
184. - OpenAI API (desde n8n)
185.
186. ### **Herramientas:**
187. - Cursor AI (para todo el desarrollo)
188. - MCP Supabase (configurado)
189. - CLI Supabase (solo para setup)
190.
191. ---
```



Sistema de Innovación  
e Inteligencia



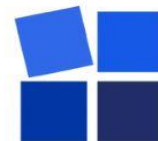
**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.



Grupo Almatodo  
SOLUTIONS FOR BUSINESS

```
192.
193.    ## 📅 Plan de Desarrollo (3 Horas)
194.
195.    ### **🕒 Hora 1: Setup Base**
196.    ```
197.    - Crear proyecto Supabase
198.    - Configurar variables entorno
199.    - Setup n8n workflow
200.    - Verificar conexiones
201.    ```
202.
203.    ### **🕒 Hora 2: Autenticación**
204.    ```
205.    - Ejecutar prompt Etapa 2 (Supabase Auth)
206.    - Probar login/signup
207.    - Verificar protección rutas
208.    ```
209.
210.    ### **🕒 Hora 3: Chat + n8n**
211.    ```
212.    - Ejecutar prompt Etapa 3 (Chat n8n)
213.    - Probar flujo completo
214.    - Ajustes finales
215.    ```
216.
217.    ---
218.
219.    ## 🛠 Configuración Requerida (Mínima)
220.
221.    ### 📁 **Variables de Entorno (4 total)**
222.    ```bash
223.    # Supabase
224.    NEXT_PUBLIC_SUPABASE_URL=tu-proyecto.supabase.co
225.    NEXT_PUBLIC_SUPABASE_ANON_KEY=tu-anon-key
226.
227.    # n8n
228.    N8N_WEBHOOK_URL=tu-n8n.app/webhook/chat
229.    N8N_WEBHOOK_SECRET=secret-generado
```



Sistema de Innovación  
e Inteligencia



**Telefono.**  
55 7585 4438

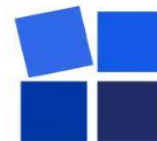


**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.





Grupo Almatodo  
SOLUTIONS FOR BUSINESS

```
230.    ``
231.
232.    ### ✨ **Herramientas Necesarias**
233.    - Cursor AI (con MCP Supabase)
234.    - Cuenta Supabase (gratis)
235.    - Cuenta n8n Cloud (gratis)
236.    - 30 minutos de tiempo
237.
238.    ---
239.
240.    ## 📊 Métricas de Éxito PMV
241.
242.    | Criterio | Objetivo | ¿Funciona? |
243.    |-----|-----|-----|
244.    | ⚡ Setup Time | < 1 hora | ✓ / ✗ |
245.    | 💬 Chat Funcional | Usuario puede chatear | ✓ / ✗ |
246.    | 🔒 Auth Funcional | Login/logout sin errores | ✓ / ✗ |
247.    | 📧 n8n Conectado | Respuestas automáticas | ✓ / ✗ |
248.    | 💾 Persistencia | Mensajes se guardan | ✓ / ✗ |
249.
250.    **Criterio de éxito: 5/5 ✓**
251.
252.    ---
253.
254.    ## 🏁 Definición de "Terminado"
255.
256.    ### ✓ **PMV está completo cuando:**
257.
258.    1. **Usuario puede:**
259.    - Registrarse con email/contraseña
260.    - Hacer login y ver página de chat
261.    - Crear nuevo chat con botón "New Chat"
262.    - Escribir mensaje y recibir respuesta IA
263.    - Ver historial de mensajes del chat actual
264.    - Navegar entre diferentes chats desde sidebar
265.    - Ver lista de sus chats en sidebar
266.    - Hacer logout
267.
```



Sistema de Innovación  
e Inteligencia



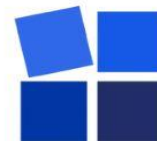
**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.



Grupo Almatodo  
SOLUTIONS FOR BUSINESS

```
268. 2. **Técnicamente:**
269.   - n8n recibe mensajes desde la app
270.   - OpenAI procesa y responde
271.   - Mensajes se guardan en Supabase
272.   - UI no tiene errores visuales
273.
274. 3. **Desarrollo:**
275.   - Código funciona sin errores fatales
276.   - Se puede deployar a Vercel
277.   - Variables de entorno configuradas
278.
279. ### 🚀 **Listo para v2:**
280. Una vez que el PMV funciona 100%, entonces se puede considerar:
281.   - Múltiples chats
282.   - Mejor UI/UX
283.   - Manejo de errores
284.   - Funcionalidades avanzadas
285.
286. ---
287.
288. ## 💡 Lecciones Aprendidas
289.
290. ### ✅ **Qué SÍ funciona:**
291.   - Prompts específicos y probados
292.   - Usar plantillas existentes sin modificar mucho
293.   - MCP de Supabase + CLI
294.   - Workflow n8n simple
295.   - Un endpoint que maneja todo
296.
297. ### ❌ **Qué evitar:**
298.   - Realtime updates complejos
299.   - Múltiples endpoints
300.   - Sidebar complejo desde v1
301.   - Error handling avanzado
302.   - Optimizaciones prematuras
303.   - Funcionalidades "nice to have"
304.
305. ### 🏁 **Principio clave:**
```



Sistema de Innovación  
e Inteligencia



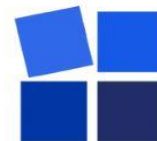
**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.



Grupo Almatodo  
SOLUTIONS FOR BUSINESS

```
306.    **"Hacer que una cosa funcione perfectamente es mejor que hacer
      muchas cosas que medio funcionan"**
307.
308.    ---
309.
310.    ## 🚩 Resultado Final
311.
312.    **Un chat funcional donde puedes:**
313.    - Entrar con tu usuario
314.    - Escribir un mensaje
315.    - Recibir respuesta de tu agente n8n
316.    - Ver el historial guardado
317.    - **¡Y YA!**
318.
319.    **Todo lo demás es v2, v3, v4...**
320.
321.    ¡**PMV real y alcanzable!** 🚀
```

## PRD PARA CURSOR DE COTIZADOR:

### PRD Optimizado – CotizAI PMV (Agente para Cursor)

#### 🎯 ¿Qué Vamos a Crear?

Un **cotizador funcional end-to-end** que:

- Permite a **Compras (Gabino)** subir **PDF/Excel** con listas de precios.
- Envía los archivos a **n8n + IA** para **parseo y normalización**.
- Guarda costos en **Supabase** (con **RLS**: Ventas **no** ve costos).
- Permite a **Ventas** generar **cotizaciones** con **desglose** (precio, IVA, utilidad, comisión).
- Envía la cotización por **WhatsApp/Email** vía **n8n** y registra el estatus.

**En resumen:** Repositorio de costos + Cotizador con IA detrás de n8n.

#### 🔍 Experiencia del Usuario (Realista)



Sistema de Innovación  
e Inteligencia

by Grupo Almatodo



**Telefono.**  
55 7585 4438













**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.

## Flujo PMV:

 Compras (Gabino) →  Login →  Sube PDF/Excel →  n8n procesa →  Costos en DB

 Ventas →  Login →  Cotiza (marca/volumen/zona) →  n8n envía →  Cotización registrada

## Lo que Sí tiene:

- ✓ Autenticación Supabase (roles: compras, ventas, admin)
- ✓ Subida de archivos y **parseo IA** vía n8n
- ✓ Repositorio de costos en Supabase
- ✓ Cotizador con desglose (precio, IVA, utilidad, comisión)
- ✓ Envío por n8n y registro de estatus

## Lo que NO tiene (para v2):

- ✗ Dashboards avanzados (v1 solo tablas simples)
- ✗ Integración profunda con CRM (solo básica si se requiere)
- ✗ Realtime/WebSockets
- ✗ Branding avanzado por marca (plantilla simple)
- ✗ Manejo de errores "fancy" (happy path sólido primero)

## Arquitectura PMV

graph LR

```

A[Usuario] --> B[Next.js App]
B --> C[Supabase Auth (roles)]
B --> D[/api/upload | /api/quote | /api/send]
D --> E[n8n Webhooks (ingest & send)]
E --> F[OpenAI (parseo IA)]
F --> E
E --> G[Supabase REST (costos / cotizaciones)]
B --> G
  
```



Sistema de Innovación  
e Inteligencia



**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.

## Flujo técnico simple:

[Compras sube archivo] → [POST /api/upload] → [n8n/ingest parsea y guarda costos] → [Supabase]  
 [Ventas cotiza] → [POST /api/quote calcula y guarda] → [POST /api/send] → [n8n/send entrega] → [status=sent]

## □ Componentes PMV

### 🌐 Frontend (Mínimo)

- 🔒 Login/Signup (Supabase)
- 📄 /compras (form para subir PDF/Excel: proveedor, marca, archivo)
- 📄 /ventas (form para cotizar: cliente, marca, zona, volumen, productos/cantidades)
- 📄 /cotizaciones (tabla simple con estatus: borrador/enviada)
- (opcional) /dashboards (tablas simples por ahora)

### 🔌 Backend (3 endpoints)

/api/upload (POST) → solo compras/admin: sube archivo a Storage y llama n8n/ingest  
 /api/quote (POST) → ventas/admin: calcula con reglas y guarda quote + items  
 /api/send (POST) → ventas/admin: llama n8n/send, marca status='sent'

### □ n8n (2 workflows)

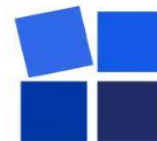
- 📧 /webhook/cotizai/ingest: valida secret → parsea PDF/Excel (IA para PDF) → inserta costos
- 📧 /webhook/cotizai/send: valida secret → arma mensaje persuasivo → WhatsApp/Email → responde OK

### 🗄️ Base de Datos (mínimo PMV)





- 👤 profiles (id, email, role['compras','ventas','admin'])
- 👤 suppliers (id, name, brand, contact, is\_active)







Grupo Almatodo  
SOLUTIONS FOR BUSINESS

-  **products** (id, sku, name, brand, unit, category, is\_active)
-  **price\_lists** (id, supplier\_id, file\_url, uploaded\_by, uploaded\_at, status)
- \$ purchase\_prices** (id, product\_id, supplier\_id, cost, currency, source\_list\_id, valid\_from, valid\_to)
-  **margin\_rules** (id, brand, zone, volume\_min, volume\_max, base\_margin\_percent)
-  **commission\_rules** (id, seller\_email|role, percent)
- ☐ **quotes** (id, seller\_id, customer\_name, brand, zone, volume, subtotal, iva, total, commission\_amount, utility\_amount, status, lost\_reason, created\_at)
- ☐ **quote\_items** (id, quote\_id, product\_id, quantity, unit\_price, line\_total, created\_at)

**RLS:** *purchase\_prices* solo **compras/admin**; *ventas* jamás ve costos.

---

## ★ Funcionalidades PMV (Lista Corta)

### Autenticación & Roles

- ☒ Login/signup con email
- ☒ Sesión persistente
- ☒ Rutas protegidas por rol (compras / ventas / admin)

### Ingesta de Costos

- ☒ Subir PDF/Excel
- ☒ Enviar a n8n/ingest con secret
- ☒ Guardar costos normalizados en Supabase
- ☒ Confirmar filas insertadas

### ☐ Cotizador

- ☒ Selección de marca, zona, volumen y productos



Sistema de Innovación  
e Inteligencia



**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.



- ☒ Cálculo de precio unitario con margen (server-side)
- ☒ Desglose: subtotal, IVA, total, utilidad, comisión
- ☒ Guardar quote + items

#### Envío

- ☒ Enviar por WhatsApp/Email vía n8n/send
- ☒ Cambiar status a "sent"
- ☒ Registrar resultado

#### Criterios de Éxito PMV

##### ✓ Funcional (Lo mínimo)

1. **Compras** sube y procesa  $\geq 6$  listas de precios reales.
2. **Ventas** genera  $\geq 15$  cotizaciones con desglose completo.
3. **n8n** conectado, ingesta y envío funcionando.
4. **RLS** activo: ventas **no** accede a costos.

##### ✓ Técnico (Básico)

1. **Tiempo de cálculo** de cotización  $< 5$  s.
2. **Tiempo de envío** (n8n)  $< 10$  s típicamente.
3. **Precisión de parseo** PDF  $\geq 80\%$  (Excel  $\geq 95\%$ ).

##### ✓ Experiencia (Suficiente)

1. **Flujos simples y móviles** (formularios cortos).
2. **Sin errores** en happy path.
3. **UI decente** (plantilla limpia).



**SII**

Sistema de Innovación  
e Inteligencia

 Grupo Almatodo



**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.

## 🚫 Fuera del Alcance PMV

- Dashboards con gráficas complejas (tablas simples v1).
- Integración avanzada con Kommo/SICAR (solo opcional).
- Branding por marca "full" (logos sí, diseño avanzado no).
- Realtime/WebSockets.
- Manejo de errores avanzado con reintentos.
- Analytics, testing automatizado, CI/CD.

## ✅ Sí hacer:

- Ingesta + Cotizar + Enviar + Listado simple de cotizaciones.
- RLS estricta (ventas no ve costos).
- Parsers robustos (Excel directo; PDF con IA).

---

## 🔧 Stack Tecnológico (Simplificado)

**Frontend:** Next.js 14 + Tailwind (App Router)

**Backend:** API Routes (3 endpoints)

**Auth & DB:** Supabase (PostgreSQL + Auth + Storage + RLS)

**Automatización/IA:** n8n Cloud + OpenAI (parseo PDF & copy persuasivo)

**Dev:** Cursor (con .cursorsrules), MCP Supabase, Supabase CLI

**Deploy:** Vercel

---

## 📅 Plan de Desarrollo (3 Horas – guía del agente)

### 🕒 Hora 1: Setup Base (ya tienes Etapa 1)

- Verificar envs (Supabase + n8n + secret)
- Crear Storage bucket 'price-lists'
- Comprobar RLS de tablas críticas

### 🕒 Hora 2: Auth & Roles (Etapa 2 lista)



Sistema de Innovación  
e Inteligencia



**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.

- Probar login/signup
- Asignar role 'compras' a Gabino
- Probar protección de rutas

### 🕒 Hora 3: App + n8n (esta etapa)

- Implementar /api/upload, /api/quote, /api/send
- Páginas /compras y /ventas
- Conectar workflows n8n (ingest & send)
- Validar flujo end-to-end

### 🔧 Configuración Requerida (Mínima)

# Supabase

NEXT\_PUBLIC\_SUPABASE\_URL=tu-proyecto.supabase.co

NEXT\_PUBLIC\_SUPABASE\_ANON\_KEY=tu-anon-key

SUPABASE\_SERVICE\_ROLE\_KEY=service-role-key # server only

# n8n

N8N\_WEBHOOK\_URL\_INGEST=https://tu-  
instancia.app.n8n.cloud/webhook/cotizai/ingest

N8N\_WEBHOOK\_URL\_SEND=https://tu-  
instancia.app.n8n.cloud/webhook/cotizai/send

N8N\_WEBHOOK\_SECRET=secret-generado-32c

NEXT\_PUBLIC\_APP\_URL=http://localhost:3000

### 📊 Métricas de Éxito PMV

| Criterio                 | Objetivo                     | ¿Funciona? |
|--------------------------|------------------------------|------------|
| 🕒 Setup Time             | < 1 hora                     | ✓ / ✗      |
| ⬇️ Ingesta correcta      | ≥ 6 listas, sin intervención | ✓ / ✗      |
| ☐ Cotizaciones completas | ≥ 15 cotizaciones            | ✓ / ✗      |
| 🔒 RLS efectiva           | Ventas sin acceso a costos   | ✓ / ✗      |
| ✉️ Envío por n8n         | 100% de cotizaciones "sent"  | ✓ / ✗      |



| Criterio                                       | Objetivo               | ¿Funciona? |
|------------------------------------------------|------------------------|------------|
| <input type="checkbox"/> Tiempo por cotización | < 5 minutos end-to-end | ✓ / ✗      |

**Criterio global de éxito:** 5/6 ✓

### Definición de “Terminado”

#### 1. Usuario puede:

- Compras: subir PDF/Excel, ver confirmación de filas insertadas.
- Ventas: crear cotización con desglose y enviarla por WhatsApp/Email.
- Ver su lista de cotizaciones y estatus.

#### 2. Técnicamente:

- n8n/ingest parsea Excel (directo) y PDF (IA) y guarda costos.
- n8n/send entrega mensajes y responde OK.
- RLS bloquea costos a usuarios de ventas.
- No hay errores visibles en happy path.

#### 3. Dev:

- Código compila y despliega en Vercel.
- Variables de entorno configuradas.
- Logs suficientes para depurar.

### 💡 Lecciones (para el agente)

#### Qué SÍ funciona

- Un solo endpoint por función (/upload, /quote, /send).
- Excel sin IA, PDF con IA controlada (prompts claros).
- Cálculo en server-side (usa costos, pero **no** los expone a ventas).
- RLS estricta desde el día 1.

#### Qué evitar



Sistema de Innovación  
e Inteligencia



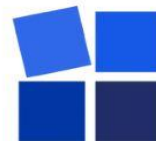
**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.



Grupo Almatodo  
SOLUTIONS FOR BUSINESS

- Parsers genéricos “mágicos” sin validación humana en PMV.
- Multiplicar endpoints innecesariamente.
- Dashboards complejos antes de tener datos.

### Principio clave

**“Un flujo perfecto > diez a medias.”**

### Resultado Final

**CotizAI PMV** donde:

- Compras alimenta costos desde PDF/Excel (con IA para PDF).
- Ventas cotiza con reglas claras (marca/volumen/zona).
- n8n envía la cotización y se registra el estatus.
- Seguridad garantizada: ventas no ve costos.

Grupo Almatodo  
SOLUTIONS FOR BUSINESS



Sistema de Innovación  
e Inteligencia



**Telefono.**  
55 7585 4438



**Whatsapp.**  
55 4029 7707



**Dirección.**  
San Miguel Jacalones,  
56604, Chalco de Días  
Covarrubias, Estado  
de México.