# Prompt Maestro para Cursor - Automatización COP

Eres un experto desarrollador fullstack. Vamos a construir un prototipo funcional de una aplicación web para automatizar el flujo del Centro de Órdenes de Pago (COP).  
  
La aplicación debe ser construida en:  
- Frontend: Next.js (React + TypeScript + TailwindCSS)  
- Backend & DB: Supabase (Postgres + Auth + RLS)  
- Autenticación: Supabase Auth con email/contraseña y OAuth con Google  
  
---  
  
## Contexto del negocio  
El área Tributaria valida y liquida cuentas de cobro de proveedores. Si todo está correcto, solicita al área COP que genere la orden de pago y la apruebe. Una vez autorizada, la orden pasa a Pagaduría para el desembolso y contabilización.  
  
El flujo a automatizar es el de COP, donde las órdenes de pago se deben generar y autorizar automáticamente según reglas de negocio.  
  
---  
  
## Funcionalidades iniciales  
  
### 1. Acceso y perfiles  
- El usuario accede mediante una URL.  
- Login con correo+contraseña o con Google OAuth.  
- Perfiles con permisos diferenciados: AdminCOP, ConsultaCOP, OperacionCOP, OperacionTRIB.  
  
### 2. Login  
- Botón "Ingresar" inactivo hasta que se llenen credenciales válidas.  
- Si credenciales incorrectas → mostrar mensaje de error.  
  
### 3. Recuperación de contraseña  
Flujo requerido:  
1. Mostrar al usuario el correo y celular registrados en user\_metadata.  
2. Permitir elegir medio (correo o SMS).  
3. Enviar OTP y validar en Supabase (tabla otp\_codes).  
4. Generar contraseña temporal y obligar al usuario a cambiarla.  
  
Incluye ejemplos de SQL y Next.js para OTP.  
  
### 4. Perfil y Cerrar Sesión  
- Avatar superior derecho estilo Google.  
- Menú con opción "Cerrar sesión".  
- Implementación con supabase.auth.signOut().  
  
### 5. Reglas automáticas del COP  
- Tabla ordenes\_pago con estados (pendiente, aprobada, rechazada).  
- Generación automática según reglas (ejemplo: montos bajos se aprueban).  
- API CRUD para registrar, evaluar reglas y actualizar estados.  
- Dashboard inicial con listado de órdenes.  
  
---  
  
## Estructura recomendada  
  
/app  
 ├── /auth  
 │ ├── login.tsx  
 │ ├── register.tsx  
 │ ├── reset-password.tsx  
 │ ├── verify-otp.tsx  
 │ └── update-password.tsx  
 │  
 ├── /dashboard  
 │ ├── index.tsx  
 │ └── profile.tsx  
 │  
 └── layout.tsx  
 └── page.tsx  
   
/pages/api  
 ├── otp.ts  
 ├── send-temp-password.ts  
 ├── twilio.ts  
 └── ordenes.ts  
  
/lib  
 ├── supabase.ts  
 ├── auth.ts  
 └── utils.ts  
  
/components  
 ├── AuthForm.tsx  
 ├── OTPInput.tsx  
 ├── ProtectedRoute.tsx  
 ├── Navbar.tsx  
 └── UserAvatar.tsx  
  
/styles  
 └── globals.css  
  
---  
  
## Objetivo del prompt  
Construir en Next.js + Supabase un prototipo con:  
- Login/Registro con Auth y Google.  
- Recuperación de contraseña con OTP y contraseña temporal.  
- Roles y permisos.  
- Navbar con avatar y logout.  
- Dashboard con reglas automáticas de órdenes de pago.  
  
---  
  
⚡ Cursor debe:  
1. Generar la estructura vacía.  
2. Implementar auth, OTP, logout y dashboard.  
3. Comentar integración Twilio.  
4. Usar TailwindCSS para UI.  
  
---