# **📘 DOCUMENTO MAESTRO DE PROMPTS – DESARROLLO DE EXTENSIÓN WHATSAPP WEB (VERSIÓN PRO)**

**🎯 Objetivo General:** Desarrollar y escalar una extensión de WhatsApp Web tipo Segsmart que impacte a 200.000 usuarios, con arquitectura modular, escalabilidad en la nube, automatizaciones y control total de calidad.

## **🧩 ESTRUCTURA GENERAL**

**Duración total estimada (modular):**

* Fase 1: MVP funcional
* Fase 2: Escalamiento y módulos avanzados
* Fase 3: QA, optimización, seguridad y publicación

**Roles involucrados:**

* **Tú:** PM, QA, Visión estratégica
* **Dev Junior:** Implementación + ejecución diaria
* **ChatGPT (yo):** Copiloto técnico, generador de código, verificador

## **🗂 ÍNDICE GENERAL DE MÓDULOS**

| **Módulo** | **Fase** | **Descripción** |
| --- | --- | --- |
| 🧠 Sidebar | 1 | Panel UI sobre WhatsApp Web |
| 🏷️ Etiquetas | 1 | Crear, guardar y asignar etiquetas a chats |
| 📄 Plantillas | 1 | Enviar mensajes prediseñados al chat activo |
| 👤 Usuarios (Auth) | 1 | Login/logout con Supabase |
| ☁️ Sync Cloud | 1 | Guardar y cargar datos desde Supabase |
| 📦 Publicación | 1 | Empaque y publicación inicial |
| 📊 Dashboard | 2 | Visualización de actividad y métricas |
| 🔁 Automatizaciones | 2 | Mensajes automáticos, recordatorios |
| 🛠️ Configuración | 2 | Centro de control y preferencias del usuario |
| 🔐 Seguridad | 2 | Prevención de ataques, manejo de sesiones |
| 🧩 API Externa | 2 | Conexión con CRM / ERPs |
| 🧪 QA y Logs | 3 | Testing, reportería y auditoría |
| 🎛️ Control de Planes | 3 | Roles: Free, Pro, Enterprise |

## **🔧 FASE 1 – MVP (20 HORAS)**

### **🧠 MÓDULO 1 – SIDEBAR**

**✅ Prompt de Desarrollo:**

css

CopiarEditar

Crea una barra lateral fija tipo CRM, en el lado izquierdo de WhatsApp Web.

Debe tener 10 íconos verticales con tooltips, construida con Tailwind CSS.

Debe inyectarse desde content.js al detectar que estamos en web.whatsapp.com.

**🔍 Verificación:**

css

CopiarEditar

¿Hay conflictos entre el CSS del sidebar y el CSS nativo de WhatsApp Web?

Sugiere cómo encapsular estilos si es necesario.

**🧠 Debug:**

mathematica

CopiarEditar

La barra desaparece al cambiar de chat. ¿Cómo puedo mantenerla anclada de forma permanente en el DOM?

### **🏷️ MÓDULO 2 – ETIQUETAS**

**✅ Prompt de Desarrollo:**

kotlin

CopiarEditar

Crea un sistema de etiquetas para WhatsApp Web.

Permite al usuario: crear etiquetas con color, asignarlas al contacto actual, guardarlas en LocalStorage, y verlas al volver al chat.

**🔍 Verificación:**

nginx

CopiarEditar

Valida que las etiquetas se carguen correctamente al cambiar de chat.

¿Qué clave en LocalStorage debo usar para evitar colisiones?

**🧠 Debug:**

arduino

CopiarEditar

Las etiquetas se pierden tras reload. ¿Cómo persistirlas sin duplicar datos?

### **📄 MÓDULO 3 – PLANTILLAS**

**✅ Prompt de Desarrollo:**

css

CopiarEditar

Permite al usuario crear mensajes tipo plantilla (título + mensaje).

Debe poder seleccionarlas y enviarlas al input de WhatsApp Web con un clic.

**🔍 Verificación:**

nginx

CopiarEditar

Valida que el mensaje se inserta correctamente en el campo de texto de WhatsApp.

¿Qué selectores debo usar para no romper compatibilidad?

**🧠 Debug:**

mathematica

CopiarEditar

El mensaje se inserta pero no se envía. ¿Cómo simular correctamente la tecla Enter con JS?

### **👤 MÓDULO 4 – AUTENTICACIÓN**

**✅ Prompt de Desarrollo:**

bash

CopiarEditar

Crea un sistema de login/logout con Supabase Auth.

El usuario debe ingresar email y contraseña, recibir token y mantenerse logueado.

**🔍 Verificación:**

arduino

CopiarEditar

¿Dónde guardar la sesión para que persista al recargar WhatsApp Web sin exponer tokens?

**🧠 Debug:**

java

CopiarEditar

Error 401 al iniciar sesión. ¿Qué pasos revisar en Supabase Auth (clave, CORS, redirección)?

### **☁️ MÓDULO 5 – SINCRONIZACIÓN EN LA NUBE**

**✅ Prompt de Desarrollo:**

sql

CopiarEditar

Cuando el usuario inicia sesión, sincroniza sus etiquetas y plantillas desde Supabase.

Guarda cada dato con user\_id como foreign key.

**🔍 Verificación:**

r

CopiarEditar

Valida si los datos guardados en Supabase aparecen correctamente tras login.

Sugiere cómo optimizar la carga para que sea rápida.

**🧠 Debug:**

nginx

CopiarEditar

Los datos no aparecen al cargar el chat. ¿Dónde inyectar el código de recuperación de datos?

### **📦 MÓDULO 6 – EMPAQUE Y PUBLICACIÓN**

**✅ Prompt de Desarrollo:**

scss

CopiarEditar

Prepara el empaquetado de la extensión con manifest.json (v3), íconos, scripts y políticas.

Exporta en ZIP y prepara assets para Chrome Web Store.

**🔍 Verificación:**

CopiarEditar

¿El archivo manifest tiene los permisos correctos? Hazme una checklist de publicación Chrome.

**🧠 Debug:**

css

CopiarEditar

La extensión no se ejecuta automáticamente al abrir WhatsApp Web. ¿Qué revisar en background o content script?

## **📈 FASE 2 – ESCALAMIENTO Y FUNCIONALIDADES AVANZADAS**

### **📊 MÓDULO 7 – DASHBOARD**

**✅ Prompt de Desarrollo:**

mathematica

CopiarEditar

Crea un dashboard dentro del popup o en una URL externa (Next.js).

Debe mostrar: número de etiquetas usadas, plantillas enviadas, chats activos.

**🔍 Verificación:**

CopiarEditar

¿Puedes auditar los datos usados y verificar que el tracking por usuario sea confiable?

**🧠 Debug:**

nginx

CopiarEditar

Los datos del dashboard no se actualizan en tiempo real. ¿Cómo integrar un listener reactivo?

### **🔁 MÓDULO 8 – AUTOMATIZACIONES**

**✅ Prompt de Desarrollo:**

java

CopiarEditar

Permite al usuario configurar respuestas automáticas por palabra clave o por horario (tipo bot).

Guardar reglas en Supabase y aplicarlas sobre el DOM.

**🔍 Verificación:**

arduino

CopiarEditar

¿Hay forma de detectar eventos entrantes en los chats de WhatsApp Web desde JS sin WebSocket?

**🧠 Debug:**

mathematica

CopiarEditar

La respuesta automática se repite varias veces. ¿Cómo evitar loops o duplicaciones?

### **🛠️ MÓDULO 9 – CONFIGURACIÓN PERSONAL**

**✅ Prompt de Desarrollo:**

less

CopiarEditar

Permite que cada usuario configure: idioma, color del panel, tipo de etiquetas favoritas, orden del sidebar.

### **🔐 MÓDULO 10 – SEGURIDAD**

**Incluye:**

* Validación de tokens
* Protección CSRF / CORS
* Limpieza del DOM para evitar exposición

### **🧩 MÓDULO 11 – API EXTERNA**

**✅ Prompt de Desarrollo:**

css

CopiarEditar

Permite conectar con un CRM externo a través de una API REST.

Debe sincronizar contactos, etiquetas y plantillas de forma programada.

### **🧪 MÓDULO 12 – QA Y LOGS**

**Incluye:**

* Logs de consola avanzados
* Exportación de fallos
* Alertas por error

### **🎛️ MÓDULO 13 – PLANES Y ROLES**

**✅ Prompt de Desarrollo:**

yaml

CopiarEditar

Controla acceso a funciones según plan: Free, Pro, Enterprise.

Desactiva opciones si el usuario no está en el plan adecuado.

## **✅ BONUS: PROMPTS DE REFACTORIZACIÓN**

**🧽 Limpieza de código**

nginx

CopiarEditar

Reestructura este bloque para que sea más legible, mantenible y modular.

**🧪 Logging útil**

nginx

CopiarEditar

Agrega logs que indiquen eventos clave: carga, error, ejecución de función, resultado.