**📚 Documentación Técnica - Fase 1 Completa**

**Sistema de Bots Inteligentes para MensajeroPro SaaS**

**🎯 Objetivo de Fase 1**

Implementar un sistema de bots configurables por empresa que puedan funcionar como agentes de ventas o agentes de citas, con integración completa a WhatsApp y OpenAI.

**📋 Fase 1.1 - Configuración Base del Bot**

**Archivos involucrados:**

* sistema/cliente/modulos/bot-config.php - Vista principal
* sistema/api/v1/bot/configurar.php - API configuración
* whatsapp-service/src/botHandler.js - Lógica del bot

**Base de datos - Modificaciones:**

sql

*-- Agregar a configuracion\_bot*

ALTER TABLE configuracion\_bot ADD COLUMN (

tipo\_bot ENUM('ventas', 'citas') DEFAULT 'ventas',

prompt\_ventas TEXT,

prompt\_citas TEXT,

templates\_activo BOOLEAN DEFAULT 1,

respuestas\_rapidas JSON,

escalamiento\_config JSON,

modo\_prueba BOOLEAN DEFAULT 0,

numero\_prueba VARCHAR(20)

);

*-- Nueva tabla para templates*

CREATE TABLE bot\_templates (

id INT PRIMARY KEY AUTO\_INCREMENT,

tipo\_negocio VARCHAR(50), *-- restaurante, tienda, clinica, salon*

tipo\_bot ENUM('ventas', 'citas'),

nombre\_template VARCHAR(100),

prompt\_template TEXT,

respuestas\_rapidas\_template JSON,

configuracion\_adicional JSON,

activo BOOLEAN DEFAULT 1

);

*-- Nueva tabla para métricas*

CREATE TABLE bot\_metricas (

id INT PRIMARY KEY AUTO\_INCREMENT,

empresa\_id INT,

fecha DATE,

conversaciones\_iniciadas INT DEFAULT 0,

conversaciones\_completadas INT DEFAULT 0,

escalamientos INT DEFAULT 0,

hora\_pico VARCHAR(5), *-- "14:00"*

preguntas\_frecuentes JSON,

INDEX idx\_empresa\_fecha (empresa\_id, fecha)

);

**Flujo de implementación:**

1. **Usuario selecciona tipo de bot** (ventas/citas)
2. **Elige template** según su tipo de negocio
3. **Personaliza prompt** y respuestas rápidas
4. **Activa modo prueba** para testing
5. **Activa bot** para clientes reales

**📋 Fase 1.2 - Bot de Ventas**

**Archivos involucrados:**

* sistema/cliente/modulos/catalogo-bot.php - Gestión catálogo
* sistema/api/v1/bot/subir-catalogo.php - Upload Excel/PDF
* sistema/api/v1/bot/descargar-catalogo.php - Download
* whatsapp-service/src/salesBot.js - Lógica específica ventas

**Base de datos:**

sql

CREATE TABLE catalogo\_bot (

id INT PRIMARY KEY AUTO\_INCREMENT,

empresa\_id INT UNIQUE,

archivo\_excel VARCHAR(255),

archivo\_pdf VARCHAR(255),

datos\_json LONGTEXT, *-- Productos, promociones, delivery*

fecha\_actualizacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

FOREIGN KEY (empresa\_id) REFERENCES empresas(id)

);

CREATE TABLE ventas\_bot (

id INT PRIMARY KEY AUTO\_INCREMENT,

empresa\_id INT,

numero\_cliente VARCHAR(20),

productos\_cotizados JSON,

total\_cotizado DECIMAL(10,2),

estado ENUM('cotizado', 'confirmado', 'cancelado'),

fecha\_cotizacion DATETIME,

metodo\_pago VARCHAR(50),

tipo\_entrega ENUM('tienda', 'delivery'),

direccion\_entrega TEXT,

INDEX idx\_empresa\_fecha (empresa\_id, fecha\_cotizacion)

);

**Plantilla Excel requerida:**

HOJA 1: PRODUCTOS

- Columnas: Categoría | Producto | Precio | Disponible

HOJA 2: PROMOCIONES

- Columnas: Producto | Tipo | Descripción | Precio Promo

HOJA 3: DELIVERY

- Columnas: Zona | Costo | Tiempo

- Fila especial: GRATIS DESDE: monto

**Flujo del bot de ventas:**

1. Cliente saluda → Bot responde con bienvenida
2. Cliente pide catálogo → Bot envía PDF
3. Cliente consulta precios → Bot responde del JSON
4. Cliente ordena → Bot cotiza con totales
5. Bot pregunta tipo entrega → Cliente elige
6. Bot informa pago → Cliente confirma
7. Bot registra venta

**📋 Fase 1.3 - Bot de Citas**

**Archivos involucrados:**

* sistema/cliente/modulos/horarios-bot.php - Gestión horarios
* sistema/api/v1/bot/configurar-horarios.php - API horarios
* sistema/api/v1/bot/agendar-cita.php - API citas
* whatsapp-service/src/appointmentBot.js - Lógica citas

**Base de datos:**

sql

CREATE TABLE horarios\_atencion (

id INT PRIMARY KEY AUTO\_INCREMENT,

empresa\_id INT,

dia\_semana INT, *-- 1=Lunes, 7=Domingo*

hora\_inicio TIME,

hora\_fin TIME,

duracion\_cita INT DEFAULT 30, *-- minutos*

activo BOOLEAN DEFAULT 1,

FOREIGN KEY (empresa\_id) REFERENCES empresas(id),

INDEX idx\_empresa\_dia (empresa\_id, dia\_semana)

);

CREATE TABLE citas\_bot (

id INT PRIMARY KEY AUTO\_INCREMENT,

empresa\_id INT,

numero\_cliente VARCHAR(20),

nombre\_cliente VARCHAR(100),

fecha\_cita DATE,

hora\_cita TIME,

tipo\_servicio VARCHAR(100),

estado ENUM('agendada', 'confirmada', 'cancelada', 'completada'),

recordatorio\_24h BOOLEAN DEFAULT 0,

recordatorio\_2h BOOLEAN DEFAULT 0,

notas TEXT,

fecha\_creacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

INDEX idx\_empresa\_fecha (empresa\_id, fecha\_cita),

INDEX idx\_fecha\_hora (fecha\_cita, hora\_cita)

);

CREATE TABLE servicios\_disponibles (

id INT PRIMARY KEY AUTO\_INCREMENT,

empresa\_id INT,

nombre\_servicio VARCHAR(100),

duracion\_minutos INT DEFAULT 30,

requiere\_preparacion TEXT, *-- "Venir en ayunas"*

activo BOOLEAN DEFAULT 1,

FOREIGN KEY (empresa\_id) REFERENCES empresas(id)

);  
  
**Agregando Google Calendar**

**Modificaciones necesarias:**

**En la base de datos:**

sql

ALTER TABLE configuracion\_bot ADD COLUMN (

google\_calendar\_activo BOOLEAN DEFAULT 0,

google\_client\_id VARCHAR(255),

google\_client\_secret VARCHAR(255),

google\_refresh\_token TEXT,

google\_calendar\_id VARCHAR(255), *-- ID del calendario a usar*

sincronizar\_citas BOOLEAN DEFAULT 1

);

**En la interfaz (bot-config.php):**

Sección: Integración Google Calendar

├── [ ] Activar sincronización con Google Calendar

├── Client ID: [\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_]

├── Client Secret: [\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_]

├── [Botón: Autorizar con Google]

├── Estado: ✓ Conectado / ✗ No conectado

└── Calendario: [Seleccionar calendario ▼]

**Flujo de configuración:**

1. **Usuario ingresa credenciales** de su proyecto Google Cloud
2. **Hace clic en "Autorizar"** → Redirige a Google OAuth
3. **Google devuelve token** → Sistema guarda refresh token
4. **Usuario selecciona** cuál calendario usar
5. **Bot sincroniza** citas en ambas direcciones

**Funcionalidades con Calendar:**

**Al agendar cita:**

* Bot verifica disponibilidad real en Google Calendar
* Al confirmar, crea evento en Calendar
* Incluye datos del paciente en la descripción

**Al cancelar:**

* Bot elimina evento de Calendar
* Libera el horario automáticamente

**Ventajas:**

* Doctor ve citas en su calendario habitual
* Evita doble agendamiento
* Notificaciones automáticas de Google

**Flujo del bot de citas:**

1. Cliente pide cita → Bot pregunta servicio
2. Cliente elige → Bot muestra días disponibles
3. Cliente elige día → Bot muestra horarios
4. Cliente elige hora → Bot pide nombre
5. Bot confirma → Registra cita
6. Bot envía recordatorios automáticos

**📋 Fase 1.4 - Sistema de Templates**

**Archivos involucrados:**

* sistema/api/v1/bot/templates/ - Carpeta con templates
* sistema/api/v1/bot/cargar-template.php - API

**Templates incluidos:**

**Para Bot de Ventas:**

restaurante\_template.json

├── prompt: "Eres un mesero virtual amigable..."

├── respuestas\_rapidas: {

│ "ubicacion": "Estamos en Av. Ejemplo 123",

│ "horario": "Lun-Dom 12pm-10pm",

│ "metodos\_pago": "Efectivo, Yape, Plin, Visa"

│ }

└── configuracion: {

"tiene\_delivery": true,

"tiempo\_preparacion": 30

}

tienda\_template.json

├── prompt: "Eres un vendedor experto..."

├── respuestas\_rapidas: {...}

└── configuracion: {...}

**Para Bot de Citas:**

clinica\_template.json

├── prompt: "Eres una recepcionista médica..."

├── respuestas\_rapidas: {

│ "preparacion": "Traer DNI y seguro",

│ "cancelacion": "Con 24h anticipación"

│ }

└── configuracion: {

"duracion\_cita\_default": 30,

"dias\_anticipacion": 30

}

**📋 Fase 1.5 - Sistema de Métricas**

**Archivos involucrados:**

* sistema/cliente/modulos/bot-reportes.php - Dashboard
* sistema/api/v1/bot/metricas/ - APIs métricas
* whatsapp-service/src/metricsCollector.js - Recolector

**Métricas recolectadas:**

1. **Por conversación:**
   * Hora inicio/fin
   * Mensajes totales
   * Si completó objetivo (venta/cita)
   * Si escaló a humano
2. **Diarias agregadas:**
   * Total conversaciones
   * Tasa de éxito
   * Horarios pico
   * Preguntas más frecuentes
3. **KPIs principales:**
   * Conversiones (ventas/citas agendadas)
   * Tiempo promedio respuesta
   * Satisfacción (si no escaló)

**📋 Fase 1.6 - Modo Prueba y Escalamiento**

**Archivos involucrados:**

* sistema/api/v1/bot/modo-prueba.php - Toggle modo
* whatsapp-service/src/testMode.js - Lógica prueba
* whatsapp-service/src/escalationHandler.js - Manejo escalamiento

**Configuración de escalamiento:**

json

{

"max\_mensajes\_sin\_resolver": 5,

"palabras\_clave": ["hablar humano", "operador", "ayuda real"],

"horario\_escalamiento\_auto": {

"inicio": "22:00",

"fin": "08:00"

},

"mensaje\_escalamiento": "Te estoy transfiriendo con un asesor humano..."

}

**🚀 Guía de Implementación Paso a Paso**

**Orden de desarrollo:**

1. **Fase 1.1** - Base (3 días)
2. **Fase 1.4** - Templates (2 días)
3. **Fase 1.2** - Bot Ventas (5 días)
4. **Fase 1.3** - Bot Citas (5 días)
5. **Fase 1.5** - Métricas (3 días)
6. **Fase 1.6** - Prueba/Escalamiento (2 días)

**Total estimado: 20 días (1 desarrollador)**

**Testing requerido:**

* Carga de Excel con 500+ productos
* Cambio mensual de catálogo
* Citas concurrentes sin duplicar
* Escalamiento automático
* Templates para cada tipo negocio
* Métricas en tiempo real

**📝 Notas para desarrolladores:**

1. **TODO está en la BD** - No hardcodear nada
2. **Validar empresa\_id** en todas las consultas
3. **Logs detallados** para debugging
4. **Límites ya implementados** en documento principal
5. **JSON para flexibilidad** en configuraciones
6. **Reutilizar código** entre bot ventas y citas