

Contexto Completo - Sistema ERP Psicología

Mi Rol

Soy un asistente experto en backend Node.js 20.13.1 + MariaDB. Mi objetivo es proporcionar soluciones eficientes siguiendo estrictamente los patrones establecidos del proyecto. El usuario es un desarrollador freelance backend especializado en Node.js.

Proyecto

Sistema ERP para gestión de clínica de psicología. Un psicólogo que trabaja en 4 ubicaciones:

- Clínica A, Clínica B, Clínica C, Privado
- Cada clínica tiene color distintivo para UI (#RRGGBB)
- Distribución típica: Privado 40%, A 25%, B 20%, C 15%

Stack Tecnológico

- **Backend:** Node.js 20.13.1, MariaDB
- **Arquitectura:** MVC estricto (routes → controllers → models → utils/config)
- **Frontend:** Angular ≥18 (colaborador externo, componentes standalone)
- **Documentación:** Swagger UI obligatoria

Estructura de Respuestas API (OBLIGATORIO)

```
json
{
  "success": boolean,
  "data": any,
  "message": string,
  "total": number (opcional para listas)
}
```

CONTEXTO FUNCIONAL COMPLETO

Dashboard (Página de Inicio)

La psicóloga NO inicia aquí, pero es útil para métricas generales. Componentes:

KPIs Principales (Cards rápidos):

- Sesiones del mes
- Ingresos del mes

- Pacientes activos
- Próximas citas hoy

Gráficos y Visualizaciones:

- **Sesiones por clínica** (donut chart)
- **Ingresos mensuales** (gráfico lineal)
- **Próximas citas hoy y mañana** (texto plano con formato)
- **Distribución por modalidad** (gráfico barras: Online vs Presencial)
- **Métodos de pago** (donut chart)
- **Sesiones semanales** (gráfico barras)
- **Distribución por edad** (donut chart)
- **Resultados de sesiones** (gráfico barras)
- **Rendimiento por clínica** (texto enriquecido)

Nota: Los KPIs muestran datos de toda la BD. Futuramente se añadirán selectores de año para filtrar.

Calendario (Módulo Principal)

La psicóloga **SIEMPRE EMPIEZA AQUÍ**. Es el corazón del sistema.

Características:

- **Vistas:** Semanal y Mensual (botones toggle)
- **Horario:** 9:00 AM - 10:00 PM
- **Colores:** Cada clínica tiene color distintivo (identificación visual)
- **Leyenda:** Muestra las 4 clínicas con sus colores

Creación de Citas:

Método 1: Botón "Nueva Cita" → Abre formulario modal **Método 2:** Click directo en hora del calendario (como Microsoft Teams)

Formulario "Nueva Cita" (campos):

- **Paciente:** Dropdown con búsqueda
- **Fecha:** Date picker (dd/mm/aaaa)
- **Hora:** Time picker
- **Clínica:** Dropdown de 4 clínicas
- **Tipo de Sesión:** Dropdown (individual, grupal, familiar, pareja)
- **Precio (€):** Input numérico (la psicóloga define tarifas manualmente)

- **Método de Pago:** Dropdown (efectivo, tarjeta, transferencia, seguro, bono)
- **Modalidad:** Dropdown (Presencial, Online)
- **Notas:** Textarea para observaciones

Gestión de Citas Existentes:

- **Click en cita** → Abre popup de gestión completa
- **Recordatorio WhatsApp:** Botón en cada cita que genera deep link
- **Edición:** La psicóloga programa semanas antes, luego edita según necesidad
- **Estados:** scheduled, completed, cancelled, no-show

Gestión de Pacientes (6 Tabs)

Tab 1: RESUMEN

- **Card "Datos del Cliente":** Número ficha, email, teléfono, tipo sesión, estado
- **Card "Sesiones":** Contadores (realizadas, programadas, canceladas) + filtro por año
- **Card "Histórico de Sesiones":** Tabla con filtro por rango de fechas
- **Card "Facturación":** Gráfico gastos mensuales + filtro por año

Tab 2: DATOS DEL PACIENTE

- Formulario completo con toda la información personal y médica

Tab 3: HISTORIA CLÍNICA

- **Interfaz tipo OneNote:** Secciones jerárquicas
- **Sin plantillas predefinidas:** La psicóloga crea libremente
- **Estructura:** Por temas/sesiones según criterio profesional
- **Editor:** Títulos, subtítulos, notas por sesión

Tab 4: SESIONES

- **Tabla detallada** de todas las sesiones del paciente
- **Filtros:** Por fecha, estado, tipo, clínica
- **Acción:** "Crear Factura" desde sesiones seleccionadas

Tab 5: FACTURAS

- **Crear nuevas facturas** (individual o desde sesiones)
- **Tabla facturas definitivas**
- **Tabla facturas rectificativas**

- **Generación:** Individual o masiva según periodicidad deseada

Tab 6: BONOS

- **Sistema de bonos promocionales** (ej: 10 sesiones con descuento)
- **Estados:** active, consumed, expired
- **Gestión:** Crear nuevos, tracking de uso, historial completo
- **Restricción:** NO transferibles entre pacientes, únicos por paciente

Sistema de Sesiones

Estados del Ciclo de Vida:

1. **Scheduled:** Cita programada (puede ser semanas antes)
2. **Completed:** Sesión realizada
3. **Cancelled:** Cancelada por cualquier motivo
4. **No-show:** Paciente no asistió

Modalidades:

- **Presencial:** En cualquiera de las 4 clínicas
- **Online:** Terapia virtual

Tipos de Terapia:

- **Individual:** 1 paciente
- **Grupal:** Múltiples pacientes
- **Familiar:** Terapia familiar
- **Pareja:** Terapia de pareja

Métodos de Pago:

- **Efectivo, Tarjeta, Transferencia, Seguro:** Pago directo
- **Bono:** Descuenta de bono promocional del paciente

Facturación

Características:

- **Periodicidad:** Flexible, según criterio de la psicóloga
- **Tipos:** Individual (por sesión) o Masiva (múltiples pacientes/sesiones)
- **Tarifas:** Variables, definidas manualmente en cada sesión
- **Sin descuentos:** Solo bonos promocionales

- **Generación PDF:** Diseño moderno y profesional

Recordatorios WhatsApp

Funcionalidad:

- **Botón en cada cita** del calendario
- **Proceso:** Backend genera deep link → Frontend abre WhatsApp → Mensaje pre-rellenado
- **Control de calidad:** La psicóloga envía manualmente (no automático)
- **URL Format:** `whatsapp://send?phone=X&text=Y`
- **Contenido:** Nombre paciente, fecha, hora, tipo de cita

Flujo de Trabajo Diario

Secuencia Típica:

1. **Inicio:** Siempre en Calendario (módulo principal)
2. **Revisión:** Citas del día, próximas citas
3. **Gestión:** Editar citas, enviar recordatorios WhatsApp
4. **Durante/Después sesión:** Actualizar estado, notas, historia clínica
5. **Administrativo:** Facturación periódica, gestión de bonos

Sin Automatizaciones:

- **No hay reportes automáticos**
- **No hay métricas de seguimiento**
- **No hay notificaciones automáticas** (solo WhatsApp manual)
- **No hay plantillas predefinidas**

Estado Actual de Desarrollo

APIs Implementadas

- **Sessions:** GET (filtros dinámicos), POST (crear), PUT (actualizar)
- **Patients:** GET (filtros dinámicos)
- **Auth:** Sistema JWT implementado con tabla users

Estructura de Base de Datos

Tabla patients

```
sql
```

```
CREATE TABLE `patients` (  
  `id` bigint(20) NOT NULL AUTO_INCREMENT,  
  `name` varchar(255) NOT NULL,  
  `email` varchar(255) NOT NULL,  
  `phone` varchar(20) NOT NULL,  
  `dni` varchar(15) NOT NULL,  
  `status` enum('active','inactive','discharged','on-hold') DEFAULT 'active',  
  `session_type` enum('individual','group','family','couples') DEFAULT 'individual',  
  `address` text DEFAULT NULL,  
  `birth_date` date DEFAULT NULL,  
  `emergency_contact_name` varchar(255) DEFAULT NULL,  
  `emergency_contact_phone` varchar(20) DEFAULT NULL,  
  `medical_history` text DEFAULT NULL,  
  `current_medication` text DEFAULT NULL,  
  `allergies` varchar(500) DEFAULT NULL,  
  `referred_by` varchar(255) DEFAULT NULL,  
  `insurance_provider` varchar(255) DEFAULT NULL,  
  `insurance_number` varchar(50) DEFAULT NULL,  
  `notes` text DEFAULT NULL,  
  `is_active` boolean NOT NULL DEFAULT true,  
  `created_at` timestamp NULL DEFAULT current_timestamp(),  
  `updated_at` timestamp NULL DEFAULT current_timestamp() ON UPDATE current_timestamp()  
);
```

Tabla sessions

sql

```
CREATE TABLE `sessions` (  
  `id` bigint(20) NOT NULL AUTO_INCREMENT,  
  `patient_id` bigint(20) NOT NULL,  
  `clinic_id` bigint(20) NOT NULL,  
  `session_date` date NOT NULL,  
  `start_time` time NOT NULL,  
  `end_time` time NOT NULL,  
  `mode` varchar(100) NOT NULL,  
  `type` varchar(100) NOT NULL,  
  `status` enum('scheduled','completed','cancelled','no-show') DEFAULT 'scheduled',  
  `price` decimal(10,2) NOT NULL DEFAULT 0.00,  
  `payment_method` enum('cash','card','transfer','insurance','bonus') DEFAULT 'cash',  
  `payment_status` enum('pending','paid','partially_paid') DEFAULT 'pending',  
  `notes` text DEFAULT NULL,  
  `is_active` boolean NOT NULL DEFAULT true,  
  `created_at` timestamp NULL DEFAULT current_timestamp(),  
  `updated_at` timestamp NULL DEFAULT current_timestamp() ON UPDATE current_timestamp()  
);
```

Tabla clinics

sql

```
CREATE TABLE `clinics` (  
  `id` bigint(20) NOT NULL AUTO_INCREMENT,  
  `name` varchar(255) NOT NULL,  
  `address` text DEFAULT NULL,  
  `clinic_color` varchar(7) DEFAULT NULL COMMENT 'Color en formato hexadecimal #RRGGBB',  
  `is_active` boolean NOT NULL DEFAULT true,  
  `created_at` timestamp NULL DEFAULT current_timestamp(),  
  `updated_at` timestamp NULL DEFAULT current_timestamp() ON UPDATE current_timestamp()  
);
```

Tabla bonuses

sql

```
CREATE TABLE `bonuses` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT,
  `patient_id` bigint(20) NOT NULL,
  `total_sessions` int(11) NOT NULL,
  `price_per_session` decimal(10,2) NOT NULL,
  `total_price` decimal(10,2) NOT NULL,
  `used_sessions` int(11) NOT NULL DEFAULT 0,
  `status` enum('active','consumed','expired') DEFAULT 'active',
  `purchase_date` date NOT NULL,
  `expiry_date` date NOT NULL,
  `notes` text DEFAULT NULL,
  `created_at` timestamp NULL DEFAULT current_timestamp(),
  `updated_at` timestamp NULL DEFAULT current_timestamp() ON UPDATE current_timestamp()
);
```

Tabla bonus_usage_history

sql

```
CREATE TABLE `bonus_usage_history` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT,
  `bonus_id` bigint(20) NOT NULL,
  `session_id` bigint(20) DEFAULT NULL,
  `used_date` date NOT NULL,
  `sessions_consumed` int(11) NOT NULL DEFAULT 1,
  `notes` varchar(500) DEFAULT NULL,
  `created_by` varchar(255) DEFAULT NULL,
  `created_at` timestamp NULL DEFAULT current_timestamp()
);
```

Tabla users

sql

```
CREATE TABLE `users` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT,
  `email` varchar(255) NOT NULL,
  `password_hash` varchar(255) NOT NULL,
  `name` varchar(255) NOT NULL,
  `last_login` timestamp NULL DEFAULT NULL,
  `is_active` boolean NOT NULL DEFAULT true,
  `created_at` timestamp NULL DEFAULT current_timestamp(),
  `updated_at` timestamp NULL DEFAULT current_timestamp() ON UPDATE current_timestamp()
);
```


Soft Delete Implementado

- Todas las tablas principales tienen `is_active` boolean DEFAULT true
- Controllers filtran por `WHERE is_active = true`
- DELETE = UPDATE SET is_active = false

Sistema de Autenticación JWT

- JWT_SECRET: Clave segura de 64 caracteres generada aleatoriamente
- JWT_EXPIRATION: 8-12 horas recomendado
- Usuario único: psicologa@psicologia.es / password123
- Funciones: generateToken(), verifyToken() simplificadas

Datos de Prueba Actuales

- Dashboard KPIs: 45 sesiones/mes, €6750 ingresos, +12.5% crecimiento
- 8 pacientes con diferentes estados
- 5 citas programadas para hoy
- 5 bonos con historial completo (25 registros de uso)

Workflow COMPLETO para Endpoints (SIEMPRE incluir los 4)

1. **Controller:** Extraer parámetros, validaciones, try/catch, respuesta estándar
2. **Model:** Queries SQL dinámicas con parámetros preparados
3. **Route:** Express router con importación de funciones del controller
4. **Swagger Documentation:** Paths, schemas, responses completas

Convenciones de Código

Patrón Controller

```
javascript
```

```

const obtenerEntidad = async (req, res) => {
  try {
    const { param1, param2 } = req.query;
    const filters = {};
    if (param1) filters.param1 = param1;

    const datos = await getEntidad(filters);

    res.json({
      success: true,
      total: datos.length,
      data: datos,
    });
  } catch (err) {
    console.error("Error:", err.message);
    res.status(500).json({
      success: false,
      error: "Error al obtener datos",
    });
  }
};

```

Patrón Model

javascript

```

const getEntidad = async (filters = {}) => {
  let query = "SELECT * FROM tabla WHERE is_active = true";
  const params = [];
  const conditions = [];

  if (filters.campo) {
    conditions.push("campo = ?");
    params.push(filters.campo);
  }

  if (conditions.length > 0) {
    query += " AND " + conditions.join(" AND ");
  }

  const [rows] = await db.execute(query, params);
  return rows;
};

```

Instrucciones de Trabajo

- Todas las tablas tienen CREATE sin ENGINE= al final
- Siempre incluir inserts de prueba (5 por tabla principal)
- Filtros dinámicos en todos los GET
- Soft delete obligatorio: `WHERE is_active = true`
- Validaciones en controllers, queries en models
- Try/catch obligatorio en controllers
- Documentación Swagger completa para cada endpoint

Configuración de Claude Agent

- Archivo .claude_rules en raíz del proyecto
- Usar `@rules` en cada prompt para mantener contexto
- Patrones y convenciones definidos claramente

El usuario busca mantener consistencia absoluta en todo el desarrollo, siguiendo los patrones MVC establecidos y las convenciones de respuestas API.