

Diagnóstico de Módulos Faltantes - CitaPlanner

Fecha: 8 de octubre de 2025
Versión: v1.0.0
Estado: Análisis completado

Resumen Ejecutivo

Después de analizar el código fuente de CitaPlanner v1.0.0, se han identificado **tres módulos críticos** que están marcados como “en desarrollo” o tienen funcionalidad incompleta:

- 1. **✓ Módulo de Clientes:** FUNCIONAL (confirmado por usuario)
- 2. **✗ Módulo de Citas:** INCOMPLETO - Sin endpoints de creación/edición
- 3. **✗ Módulo de Ventas:** PARCIALMENTE FUNCIONAL - Falta integración con clientes
- 4. **✗ Módulo de Servicios:** INCOMPLETO - Solo tiene endpoints API, sin UI funcional

1. Módulo de Citas (Appointments)

Estado Actual

- ✗ **UI:** Página placeholder con mensaje “Módulo en Desarrollo”
- ✗ **API:** Solo existe endpoint `/api/appointments/recent` (GET)
- ✗ **Funcionalidad:** No se pueden crear, editar o eliminar citas

Archivos Afectados

<code>app/app/dashboard/appointments/page.tsx</code>	- Página placeholder
<code>app/components/modals/appointment-modal.tsx</code>	- Modal con datos mock
<code>app/app/api/appointments/recent/route.ts</code>	- Solo lectura de citas recientes

Problemas Identificados

1.1 Falta Endpoint de Creación

Archivo: `app/app/api/appointments/route.ts` (NO EXISTE)

Problema: No existe el endpoint principal para CRUD de citas.

Impacto:

- No se pueden crear citas nuevas
- No se pueden editar citas existentes
- No se pueden eliminar citas

1.2 Modal con Datos Mock

Archivo: `app/components/modals/appointment-modal.tsx`

Problema: El modal usa datos hardcodeados en lugar de llamadas API reales:

```
const mockClients = [
  { id: '1', name: 'María González', phone: '+52 55 1234 5678' },
  // ...
]

const mockServices = [
  { id: '1', name: 'Corte de Cabello', duration: 60, price: 350 },
  // ...
]
```

Impacto:

- Los datos no se guardan en la base de datos
- No se integra con clientes reales del sistema
- No se integra con servicios reales del catálogo

1.3 Página Sin Funcionalidad

Archivo: app/app/dashboard/appointments/page.tsx

Problema: La página solo muestra un mensaje de “Módulo en Desarrollo”.

Impacto:

- No hay calendario interactivo
- No hay lista de citas
- No hay gestión de estados

Solución Propuesta

Fase 1: Crear Endpoints API

1. Crear app/app/api/appointments/route.ts :
 - GET: Listar citas con filtros
 - POST: Crear nueva cita
2. Crear app/app/api/appointments/[id]/route.ts :
 - GET: Obtener cita específica
 - PUT: Actualizar cita
 - DELETE: Eliminar cita

Fase 2: Actualizar Modal





1. Reemplazar datos mock con llamadas API reales
2. Integrar con /api/clients para lista de clientes
3. Integrar con /api/services para lista de servicios
4. Integrar con /api/users?role=PROFESSIONAL para profesionales

Fase 3: Implementar UI Completa

1. Crear componente de calendario interactivo
 2. Crear lista de citas con filtros
 3. Implementar gestión de estados (Pendiente, Confirmada, etc.)
 4. Agregar validaciones de horarios disponibles
-

2. Módulo de Ventas (Sales)

Estado Actual

-  **API:** Endpoints completos y funcionales
-  **UI Dashboard:** Métricas y reportes funcionando
-  **UI POS:** Funcional pero con problemas de integración
-  **Integración:** Problemas al cargar clientes

Archivos Afectados

app/app/dashboard/sales/page.tsx	- Dashboard funcional
app/app/dashboard/sales/pos/page.tsx	- POS con problemas
app/app/api/sales/route.ts	- API funcional
app/lib/services/saleService.ts	- Servicio funcional

Problemas Identificados

2.1 Error al Cargar Clientes en POS

Archivo: app/app/dashboard/sales/pos/page.tsx (línea 38)

Problema: Intenta cargar clientes con endpoint incorrecto:

```
const clientsRes = await fetch('/api/users?role=CLIENT');
```

Causa Raíz:

- Los clientes están en la tabla `Client`, no en `User`
- El endpoint correcto es `/api/clients`, no `/api/users?role=CLIENT`

Impacto:

- No se pueden seleccionar clientes al crear ventas
- El POS no puede completar transacciones sin cliente

2.2 Carga de Profesionales

Archivo: app/app/dashboard/sales/pos/page.tsx (línea 39)

Problema: Endpoint incorrecto para profesionales:

```
const professionalsRes = await fetch('/api/users?role=STAFF');
```

Causa Raíz:

- El rol correcto es `PROFESSIONAL`, no `STAFF`
- Debería ser `/api/users?role=PROFESSIONAL`

Impacto:

- No se pueden asignar profesionales a servicios vendidos
- Las comisiones no se calculan correctamente

Solución Propuesta

Fix Inmediato

Actualizar `app/app/dashboard/sales/pos/page.tsx` :

```
// Línea 38 - Cambiar de:
const clientsRes = await fetch('/api/users?role=CLIENT');
// A:
const clientsRes = await fetch('/api/clients');

// Línea 39 - Cambiar de:
const professionalsRes = await fetch('/api/users?role=STAFF');
// A:
const professionalsRes = await fetch('/api/users?role=PROFESSIONAL');
```

3. Módulo de Servicios (Services)

Estado Actual

- **✓ API:** Endpoints completos (/api/services)
- **✓ Schema:** Modelo `Service` en Prisma
- **✗ UI:** Página placeholder con mensaje “Módulo en Desarrollo”
- **✗ Funcionalidad:** No hay interfaz para gestionar servicios

Archivos Afectados

app/app/dashboard/services/page.tsx	- Página placeholder
app/app/api/services/route.ts	- API funcional
app/app/api/services/[id]/route.ts	- API funcional
app/app/api/services/categories/route.ts	- API funcional
app/components/modals/service-modal.tsx	- Modal existente

Problemas Identificados

3.1 Página Sin Funcionalidad

Archivo: app/app/dashboard/services/page.tsx

Problema: La página solo muestra un mensaje de “Módulo en Desarrollo”.

Impacto:

- No se pueden crear servicios nuevos
- No se pueden editar servicios existentes
- No se pueden ver servicios disponibles
- No se pueden asignar profesionales a servicios

3.2 Desconexión API-UI

Problema: Existe la API completa pero no hay UI que la consuma.

Endpoints Disponibles:

- **✓ GET** /api/services - Listar servicios
- **✓ POST** /api/services - Crear servicio
- **✓ GET** /api/services/[id] - Obtener servicio
- **✓ PUT** /api/services/[id] - Actualizar servicio
- **✓ DELETE** /api/services/[id] - Eliminar servicio
- **✓ GET** /api/services/categories - Listar categorías
- **✓ POST** /api/services/categories - Crear categoría

Impacto:

- Los servicios solo se pueden gestionar directamente en la base de datos
- No hay forma visual de administrar el catálogo
- Los usuarios no pueden usar la funcionalidad completa

Solución Propuesta

Fase 1: Implementar UI Básica

1. Crear tabla de servicios con datos reales
2. Integrar modal existente con API
3. Implementar CRUD completo desde la UI

Fase 2: Funcionalidades Avanzadas

1. Gestión de categorías de servicios
2. Asignación de profesionales a servicios
3. Configuración de comisiones por servicio
4. Vista de servicios más populares

4. Análisis de Dependencias

Módulos Interdependientes

```
graph TD
    S[Servicios (Services)] --> C[Citas (Appointments)]
    C --> V[Ventas (Sales)]
    S --> C
    C --> V
```

Servicios (Services)
↓
Citas (Appointments) ← Requiere servicios para funcionar
↓
Ventas (Sales) ← Puede vender servicios

Orden de Implementación Recomendado

1. **PRIORIDAD ALTA:** Módulo de Servicios
 - Es la base para citas y ventas
 - Sin servicios, no se pueden crear citas
 - Sin servicios, el POS no puede vender servicios
 2. **PRIORIDAD ALTA:** Fix de Ventas (POS)
 - Fix rápido (2 líneas de código)
 - Desbloquea funcionalidad crítica
 - Permite usar el sistema inmediatamente
 3. **PRIORIDAD MEDIA:** Módulo de Citas
 - Depende de servicios
 - Requiere más desarrollo
 - Es el core del negocio
-

5. Estimación de Esfuerzo

Fix Inmediato (1-2 horas)

- ☒ Corregir carga de clientes en POS
- ☒ Corregir carga de profesionales en POS
- ☒ Probar flujo completo de ventas

Módulo de Servicios (4-6 horas)

- Implementar tabla de servicios con datos reales
- Conectar modal con API
- Implementar CRUD completo
- Agregar gestión de categorías
- Probar integración con POS

Módulo de Citas (8-12 horas)

- Crear endpoints API completos
- Implementar calendario interactivo
- Conectar modal con API real
- Implementar gestión de estados
- Agregar validaciones de horarios
- Probar flujo completo

6. Plan de Acción Inmediato

Paso 1: Fix Crítico de Ventas (AHORA)

```
# Editar archivo POS  
# Cambiar líneas 38-39  
# Probar creación de venta
```

Paso 2: Implementar Módulo de Servicios (HOY)

```
# Crear página funcional de servicios  
# Conectar con API existente  
# Probar CRUD completo
```

Paso 3: Implementar Módulo de Citas (MAÑANA)

```
# Crear endpoints API  
# Implementar UI completa  
# Probar integración
```

7. Riesgos y Consideraciones

Riesgos Técnicos




1. **Migración de Datos:** Si hay citas o servicios en la BD, verificar compatibilidad
2. **Validaciones:** Implementar validaciones de horarios para evitar conflictos
3. **Permisos:** Verificar que los roles tengan acceso correcto a cada módulo

Consideraciones de Negocio

1. **Prioridad:** El usuario necesita crear citas y ventas YA
2. **Datos Existentes:** Verificar si hay datos de prueba que limpiar
3. **Capacitación:** El usuario necesitará documentación de los nuevos módulos

8. Conclusiones







Problemas Críticos Identificados

1.  **Módulo de Citas:** Completamente no funcional
2.  **Módulo de Ventas:** Funcional pero con bug crítico
3.  **Módulo de Servicios:** API lista pero sin UI

Causa Raíz

- Los módulos fueron marcados como “en desarrollo” en versiones anteriores
- La migración de `ClientProfile` a `Client` no afectó estos módulos
- Falta integración entre componentes UI y APIs existentes

Próximos Pasos

1.  Aplicar fix inmediato de ventas
2.  Implementar módulo de servicios
3.  Implementar módulo de citas
4.  Crear PR con todos los cambios
5.  Probar flujo completo end-to-end
6.  Desplegar en producción

Anexos

A. Estructura de Archivos a Crear

```

app/app/api/appointments/
├── route.ts                # GET, POST
├── [id]/
│   └── route.ts            # GET, PUT, DELETE
└──
app/app/dashboard/services/
├── page.tsx                # UI completa con tabla y modal
└──

```

B. Archivos a Modificar

```
app/app/dashboard/sales/pos/page.tsx      # Líneas 38-39
app/components/modals/appointment-modal.tsx # Reemplazar mock data
app/app/dashboard/appointments/page.tsx    # Implementar UI completa
```

C. Endpoints API Disponibles

Servicios (Funcionales)

-  GET /api/services
-  POST /api/services
-  GET /api/services/[id]
-  PUT /api/services/[id]
-  DELETE /api/services/[id]

Citas (Faltantes)

-  GET /api/appointments
-  POST /api/appointments
-  GET /api/appointments/[id]
-  PUT /api/appointments/[id]
-  DELETE /api/appointments/[id]

Ventas (Funcionales)

-  GET /api/sales
-  POST /api/sales
-  GET /api/sales/[id]
-  PUT /api/sales/[id]

Documento generado automáticamente por el sistema de diagnóstico de CitaPlanner

Versión del documento: 1.0

Última actualización: 8 de octubre de 2025