

RESUMEN: Archivos Afectados por Error “User not found”

Fecha: 8 de octubre de 2025


Repositorio: qhosting/citaplanner

Error: “User not found” al crear perfiles de cliente

ARCHIVOS CRÍTICOS (Requieren Fix Inmediato)


1. Backend - Lógica de Negocio

`/app/lib/clients/clientManager.ts`

- **Línea:** 48-54
- **Función:** `createClientProfile()`
- **Problema:** Valida existencia de User antes de crear ClientProfile
- **Impacto:**  CRÍTICO - Bloquea creación de clientes
- **Código problemático:**

```
const user = await prisma.user.findUnique({
  where: { id: data.userId },
});
if (!user) {
  throw new Error('User not found');
}
```

`/app/lib/clients/historyService.ts`

- **Líneas:** 38-45, 160-167
- **Funciones:** `getClientHistory()` , `addHistoryEntry()`
- **Problema:** Valida existencia de User para operaciones de historial
- **Impacto:**  ALTO - Bloquea historial de clientes
- **Código problemático:**

```
// Línea 38-45
const user = await prisma.user.findUnique({
  where: { id: targetUserId },
  select: { tenantId: true, email: true, phone: true },
});
if (!user) {
  throw new Error('User not found');
}

// Línea 160-167
const user = await prisma.user.findUnique({
  where: { id: data.userId },
});
if (!user) {
  throw new Error('User not found');
}
```

/app/lib/clients/noteManager.ts

- **Línea:** 44-50
- **Función:** createClientNote()
- **Problema:** Valida existencia de User al crear notas
- **Impacto:** 🟡 MEDIO - Bloquea creación de notas (pero createdByUserId debería ser válido)
- **Código problemático:**

```
const user = await prisma.user.findUnique({
  where: { id: data.createdByUserId },
});
if (!user) {
  throw new Error('User not found');
}
```

2. Frontend - Interfaz de Usuario

/app/app/dashboard/clients/new/page.tsx

- **Línea:** 23-25
- **Función:** handleSubmit()
- **Problema:** Envía userId del usuario autenticado (staff) en lugar del cliente
- **Impacto:** 🔴 CRÍTICO - Causa el error en backend
- **Código problemático:**

```
if (!data.userId && session?.user?.id) {
  data.userId = session.user.id; // ❌ Incorrecto
}
```

/app/app/dashboard/clients/page.tsx

- **Línea:** 54
- **Función:** fetchClients()
- **Problema:** Llama a /api/clients/profiles que puede no tener datos
- **Impacto:** 🟢 BAJO - Solo afecta visualización

- **Código:**

```
const response = await fetch('/api/clients/profiles');
```

/app/app/dashboard/clients/[id]/page.tsx

- **Función:** Vista de detalle de cliente
- **Problema:** Depende de ClientProfile que puede no existir
- **Impacto:** 🟡 MEDIO - Puede fallar al ver detalles

/app/app/dashboard/clients/[id]/edit/page.tsx

- **Función:** Edición de cliente
- **Problema:** Depende de ClientProfile existente
- **Impacto:** 🟡 MEDIO - Puede fallar al editar

3. API Endpoints

/app/app/api/clients/profiles/route.ts

- **Método:** POST
- **Línea:** 47-75
- **Problema:** Llama a `createClientProfile()` que falla
- **Impacto:** 🔴 CRÍTICO - Endpoint principal de creación

/app/app/api/clients/profiles/[id]/history/route.ts

- **Método:** GET
- **Problema:** Llama a `getClientHistory()` que puede fallar
- **Impacto:** 🟡 ALTO - Endpoint de historial

/app/app/api/clients/notes/route.ts

- **Método:** POST
- **Problema:** Llama a `createClientNote()` que valida usuario
- **Impacto:** 🟡 MEDIO - Endpoint de notas

4. Componentes Reutilizables

/app/components/clients/ClientProfileForm.tsx

- **Línea:** 35
- **Problema:** Schema de validación requiere userId
- **Impacto:** 🟡 MEDIO - Formulario valida campo incorrecto
- **Código:**

```
userId: z.string().min(1, 'El ID de usuario es requerido'),
```

/app/components/clients/ClientHistory.tsx

- **Problema:** Puede depender de datos de ClientProfile
- **Impacto:** 🟢 BAJO - Solo visualización

✓ ARCHIVOS NO AFECTADOS

Módulos que NO tienen el patrón problemático:

- `/app/lib/services/productService.ts` - ✓ OK
 - `/app/lib/services/inventoryService.ts` - ✓ OK
 - `/app/lib/services/saleService.ts` - ✓ OK
 - `/app/lib/services/serviceManager.ts` - ✓ OK
 - `/app/lib/services/reportService.ts` - ✓ OK
 - `/app/lib/services/commissionService.ts` - ✓ OK
 - `/app/lib/notifications/*.ts` - ✓ OK
-

RESUMEN DE IMPACTO

Por Severidad:

- **CRÍTICO (2 archivos):** Bloquean funcionalidad principal
 - `clientManager.ts`
 - `clients/new/page.tsx`
- **ALTO (3 archivos):** Bloquean funcionalidades secundarias
 - `historyService.ts` (2 funciones)
 - `noteManager.ts`
- **BAJO (5 archivos):** Afectan solo visualización
 - Páginas de listado y detalle
 - Componentes de UI

Por Tipo:

- **Backend (3 archivos):** Lógica de negocio
 - **Frontend (4 archivos):** Interfaz de usuario
 - **API (3 archivos):** Endpoints REST
 - **Componentes (2 archivos):** UI reutilizable
-

ARCHIVOS QUE NECESITAN MODIFICACIÓN

Solución Opción A (Usar modelo Client):

1. Crear nuevo:

- `/app/app/api/clients/route.ts` (nuevo endpoint)

2. Modificar:

- `/app/app/dashboard/clients/new/page.tsx` (cambiar endpoint)
- `/app/app/dashboard/clients/page.tsx` (cambiar endpoint)
- `/app/app/dashboard/clients/[id]/page.tsx` (adaptar a Client)

- `/app/app/dashboard/clients/[id]/edit/page.tsx` (adaptar a Client)
- `/app/components/clients/ClientProfileForm.tsx` (adaptar campos)

3. Opcional (deprecar):

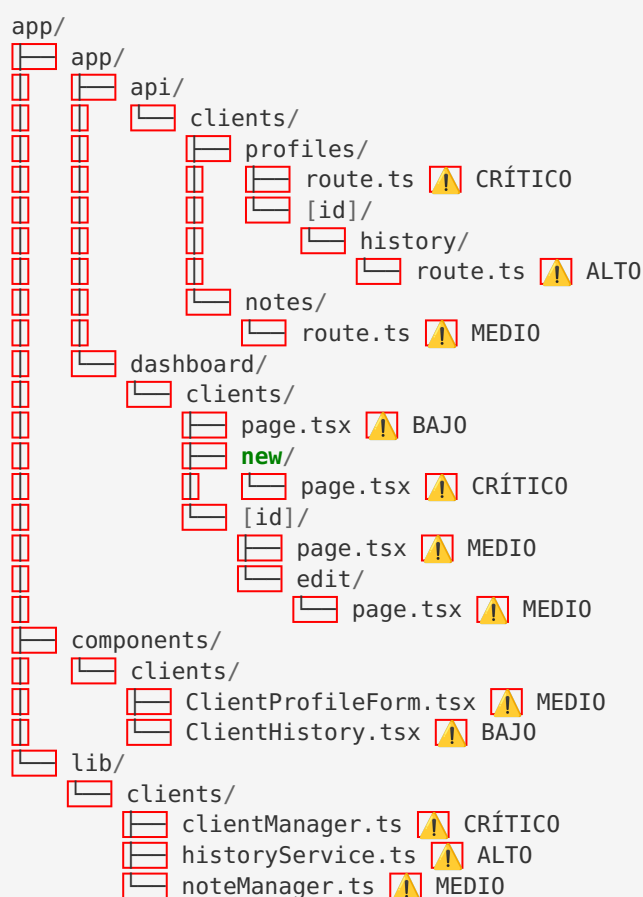
- `/app/lib/clients/clientManager.ts`
- `/app/app/api/clients/profiles/route.ts`

Solución Opción B (Crear User automáticamente):

1. Modificar:

- `/app/lib/clients/clientManager.ts` (crear User si no existe)
- `/app/app/dashboard/clients/new/page.tsx` (no enviar userId)
- `/app/components/clients/ClientProfileForm.tsx` (hacer userId opcional)

ESTRUCTURA DE DIRECTORIOS AFECTADA



🎯 PRIORIDAD DE CORRECCIÓN

Fase 1 (Inmediata):

1. `clientManager.ts` - Función `createClientProfile()`
2. `clients/new/page.tsx` - Lógica de envío de datos

Fase 2 (Alta prioridad):

1. `historyService.ts` - Funciones de historial
2. Endpoints API relacionados

Fase 3 (Media prioridad):

1. `noteManager.ts` - Creación de notas
2. Componentes de UI
3. Páginas de visualización

CONTACTO Y REFERENCIAS

- **Reporte completo:** `DIAGNOSTICO_ERROR_USER_NOT_FOUND.md`
- **Repositorio:** <https://github.com/qhosting/citaplanner>
- **Commit:** `ba4c75b`

Última actualización: 8 de octubre de 2025