Resumen del Merge - PR #97

Estado del Merge

- **PR:** #97
- **Estado:** Mergeado exitosamente
- **Método:** Squash merge
- **Fecha:** 13 de Octubre, 2025
- **Commit SHA:** `04c6c6592561425707b62ff048c1996a8b0a13a5`
- **Branch origen:** `fix/service-delete-validation` (eliminado)
- **Branch destino:** `main`

Título del PR

feat: Validación de eliminación de servicios con citas asociadas (Fix P2003)

Descripción del Problema

Error Original

```
Foreign key constraint violated on the constraint: `appointments_serviceId_fkey`
Código Prisma: P2003
```

Causa Raíz

El sistema intentaba eliminar servicios que tenían citas asociadas, violando la restricción de integridad referencial de la base de datos. PostgreSQL rechazaba la operación debido a la clave foránea `appointments.serviceId` que referencia a `services.id`.

Solución Implementada

1. Validación Previa en ServiceManager

- Verificación de citas asociadas antes de intentar eliminación
- Uso de `prisma.appointment.count()` para contar citas
- Error personalizado con formato `APPOINTMENTS_EXIST:{count}`
- Logging detallado del número de citas encontradas
- Prevención del error P2003 antes de llegar a la base de datos

2. Manejo Mejorado de Errores en API

- Manejo específico del error `APPOINTMENTS_EXIST`
- Manejo del error P2003 como fallback
- Mensajes en español claros y descriptivos
- Respuestas estructuradas con detalles:
  - `reason`: Código de la razón del error
  - `appointmentCount`: Número de citas asociadas
  - `suggestion`: Sugerencia de acción alternativa
- Códigos HTTP apropiados (400, 404, 500)
- Logging exhaustivo en cada paso

Archivos Modificados

1. app/lib/services/serviceManager.ts

Cambios: +11 líneas **Tipo:** Validación de negocio

Funcionalidad agregada:

```
// Verificar si el servicio tiene citas asociadas
const appointmentCount = await prisma.appointment.count({
  where: { serviceId: id },
});

console.log(`[ServiceManager] Checking appointments for service ${id}: ${appointmentCount} found`);

if (appointmentCount > 0) {
  throw new Error(`APPOINTMENTS_EXIST:${appointmentCount}`);
}
```

Impacto: - Previene eliminación de servicios con citas - Proporciona información precisa del número de citas - Mejora la experiencia del usuario con mensajes claros

---

2. app/app/api/services/[id]/route.ts

Cambios: +60 líneas, -5 líneas (65 cambios totales) **Tipo:** Manejo de errores y respuestas

Mejoras implementadas:

a) Logging Detallado

```
console.log('[Services API] Attempting to delete service:', { serviceId: params.id, tenantId });
console.log('[Services API] Service deleted successfully:', params.id);
console.error('[Services API] DELETE error:', error);
console.error('[Services API] Error details:', {
  message: error.message,
  code: error.code,
  meta: error.meta,
  stack: error.stack
});
```

b) Manejo de Error APPOINTMENTS_EXIST

```
if (error.message && error.message.startsWith('APPOINTMENTS_EXIST:')) {
  const appointmentCount = error.message.split(':')[1];
  return NextResponse.json({
    success: false,
    error: `No se puede eliminar el servicio porque tiene ${appointmentCount} cita(s) asociada(s)`,
    details: {
      reason: 'APPOINTMENTS_EXIST',
      appointmentCount: parseInt(appointmentCount),
      suggestion: 'Puede desactivar el servicio en lugar de eliminarlo para mantener el historial de ci
    }
  }, { status: 400 });
}
```

c) Manejo de Error P2003 (Fallback)

```
if (error.code === 'P2003') {
  return NextResponse.json({
    success: false,
    error: 'No se puede eliminar el servicio porque tiene registros asociados (citas, ventas, etc.)',
    details: {
      reason: 'FOREIGN_KEY_CONSTRAINT',
      suggestion: 'Puede desactivar el servicio en lugar de eliminarlo para mantener la integridad de l
    }
  }, { status: 400 });
}
```

Impacto: - Mensajes de error claros y accionables - Mejor debugging en producción - Experiencia de usuario mejorada

---

3. `docs/SERVICE_DELETE_VALIDATION.md`

Cambios: +582 líneas (nuevo archivo) **Tipo:** Documentación técnica

Contenido: - Descripción detallada del problema y solución - Diagramas de flujo de validación - Casos de uso con ejemplos de request/response - Guía de testing manual - Recomendaciones para usuarios - Referencias técnicas

---

Casos de Uso

Caso 1: Eliminación Exitosa (Sin Citas)

Request:

```
DELETE /api/services/clxyz123
Authorization: Bearer {token}
```

Response:

```
{
  "success": true,
  "message": "Servicio eliminado exitosamente"
}
```

Status: 200 OK

Caso 2: Eliminación Bloqueada (Con Citas)

Request:

```
DELETE /api/services/clxyz456
Authorization: Bearer {token}
```

Response:

```
{
  "success": false,
  "error": "No se puede eliminar el servicio porque tiene 5 cita(s) asociada(s)",
  "details": {
    "reason": "APPOINTMENTS_EXIST",
```

```
    "appointmentCount": 5,
    "suggestion": "Puede desactivar el servicio en lugar de eliminarlo para mantener el historial de ci
  }
}
```

Status: 400 Bad Request

Logs del servidor:

```
[ServiceManager] Checking appointments for service clxyz456: 5 found
[Services API] Cannot delete service - 5 appointments exist
```

Recomendación para Usuarios

Opción Recomendada: Desactivar en lugar de Eliminar

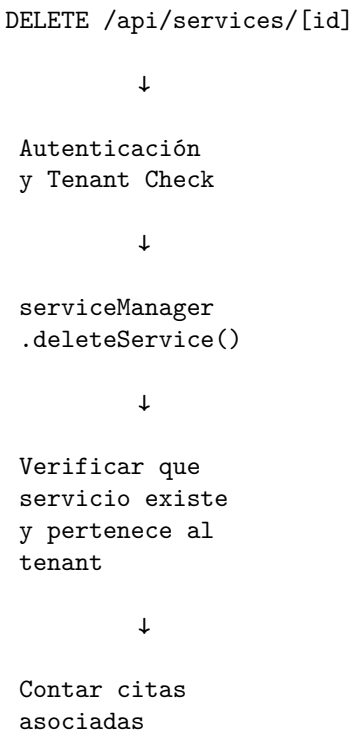
En lugar de eliminar servicios con historial, se recomienda **desactivarlos**:

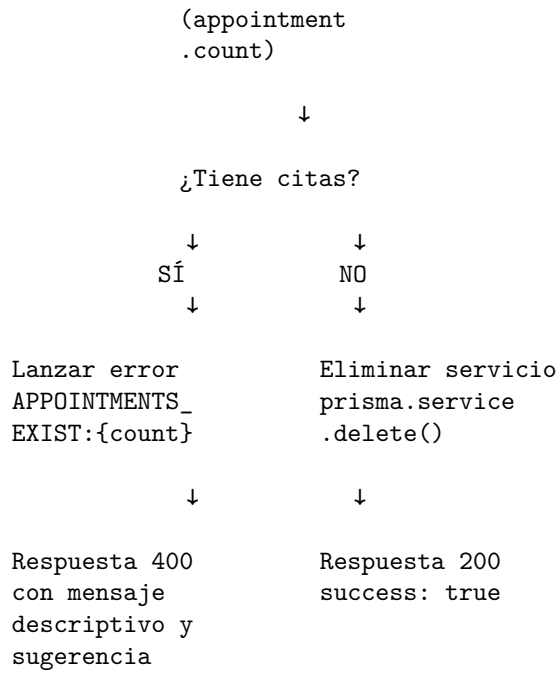
```
PUT /api/services/[id]
Content-Type: application/json

{
  "isActive": false
}
```

Ventajas: - Mantiene el historial de citas - Preserva la integridad de los datos - Permite reactivar el servicio en el futuro - No afecta reportes históricos - Evita pérdida de información valiosa

---

Flujo de Validación





Testing Recomendado

Pruebas Manuales

1. **Crear un servicio de prueba**
2. **Crear una cita con ese servicio**
3. **Intentar eliminar el servicio** → Debe mostrar error descriptivo
4. **Eliminar la cita**
5. **Intentar eliminar el servicio nuevamente** → Debe eliminarse exitosamente

---

Deployment

Información de Deployment

- **No se requieren migraciones de base de datos**
- **No hay breaking changes**
- **Compatible con versión actual**
- **Solo mejora el manejo de errores**
- **Deployment automático en Easypanel activado**

Pasos Post-Deployment

1. **Monitorear logs del servidor**
   - Buscar logs con prefijo `[Services API]` y `[ServiceManager]`
   - Verificar que la validación funciona correctamente
2. **Verificar funcionalidad**
   - Probar eliminación de servicios sin citas (debe funcionar)
   - Probar eliminación de servicios con citas (debe bloquearse con mensaje claro)
3. **Validar mensajes de error**

- Confirmar que los mensajes están en español
- Verificar que incluyen sugerencias útiles

Estadísticas del PR

- **Archivos modificados:** 3
- **Líneas agregadas:** 653
- **Líneas eliminadas:** 5
- **Cambios totales:** 658
- **Commits:** 1 (squash merge)

Enlaces Relacionados

- **PR en GitHub:** https://github.com/qhosting/citaplanner/pull/97
- **Commit en main:** https://github.com/qhosting/citaplanner/commit/04c6c6592561425707b62ff048c1996a8b0a13a5
- **Documentación técnica:** `docs/SERVICE_DELETE_VALIDATION.md`

Checklist de Verificación

☒ PR mergeado exitosamente
☒ Branch feature eliminado
☒ Commit verificado en main
☒ Repositorio local actualizado
☒ Documentación generada
☐ Testing manual en producción
☐ Verificación de logs en producción
☐ Confirmación de funcionalidad con usuarios

Lecciones Aprendidas

1. **Validación Previa es Esencial** - Siempre validar restricciones de integridad antes de operaciones destructivas
2. **Mensajes de Error Claros** - Los usuarios necesitan entender qué salió mal e incluir sugerencias
3. **Logging Detallado** - Facilita el debugging en producción
4. **Alternativas al Usuario** - Ofrecer opciones (desactivar) en lugar de solo bloquear
5. **Documentación Completa** - Documentar problema, solución y casos de uso

---

Próximos Pasos

Inmediatos

1. Monitorear el deployment automático en Easypanel
2. Verificar que no hay errores en el build
3. Confirmar que la aplicación inicia correctamente

Corto Plazo

1. Realizar testing manual en producción
2. Verificar logs del servidor
3. Confirmar que los mensajes de error son claros

Conclusión

Este PR resuelve exitosamente el error P2003 al eliminar servicios con citas asociadas, implementando:

Validación proactiva que previene el error antes de llegar a la base de datos
Mensajes claros en español con sugerencias útiles
Logging detallado para facilitar debugging
Mejor experiencia de usuario con alternativas (desactivar)
Documentación completa para referencia futura

El sistema ahora maneja correctamente la eliminación de servicios, protegiendo la integridad de los datos y proporcionando feedback claro a los usuarios.

Versión del Documento: 1.0
Última Actualización: 13 de Octubre, 2025
Autor: CitaPlanner Development Team
Commit SHA: `04c6c6592561425707b62ff048c1996a8b0a13a5`
Versión Sugerida: v1.3.2