


Pull Request: Fase 5 - Sistema de Notificaciones en Tiempo Real

Información del PR

Branch: `feature/fase5-realtime-notifications`

Base: `main`

Versión: v1.11.0

Estado:  Listo para merge

Link para Crear PR

👉 [Crear Pull Request en GitHub](https://github.com/qhosting/citaplanner/pull/new/feature/fase5-realtime-notifications) (<https://github.com/qhosting/citaplanner/pull/new/feature/fase5-realtime-notifications>)

Título del PR

Fase 5: Sistema de Notificaciones en Tiempo Real (v1.11.0)

Descripción del PR

🎯 Resumen

Sistema completo de notificaciones en tiempo real usando WebSocket (Socket.io) con sincronización multi-usuario, centro de notificaciones, preferencias configurables y actualización automática del calendario.

✨ Características Implementadas

Backend

- ✓ WebSocket Server (Socket.io) con autenticación JWT
- ✓ Realtime Notification Service para emitir eventos
- ✓ Servidor Node.js personalizado (server.js)
- ✓ Soporte multi-tenant con rooms aislados
- ✓ Room management (tenant, user, role)
- ✓ 12+ eventos WebSocket implementados

Frontend - Hooks

- ✓ ****useSocket****: Hook para gestión de conexión WebSocket
 - Auto-conexión con autenticación
 - Reconexión automática
 - Event listeners simplificados

Frontend - UI Components

- ✓ ****NotificationBell****: Icono de campana con contador de no leídas
- ✓ ****NotificationCenter****: Panel completo de notificaciones con filtros
- ✓ ****NotificationToast****: Sistema de toasts en tiempo real
- ✓ ****NotificationProvider****: Provider de contexto global

Frontend - Páginas

- ✓ **`/notifications`**: Centro de notificaciones completo
- ✓ **`/notifications/preferences`**: Configuración de preferencias
 - Canales: Push, Email, SMS, WhatsApp
 - Tipos de eventos configurables
 - Sonidos y notificaciones del navegador

Integración

- ✓ NotificationProvider envolviendo toda la app
- ✓ NotificationBell integrado en sidebar de admin
- ✓ ProfessionalCalendar con sincronización en tiempo real
- ✓ Store de notificaciones con Zustand

Base de Datos

- ✓ Migración: **`user_notification_preferences`** table
 - Preferencias por usuario
 - Canales de notificación
 - Tipos de eventos
 - Configuración de UI

📖 Documentación

Documentos Creados/Actualizados

- ✓ ****docs/FASE5_REALTIME_NOTIFICATIONS.md**** (910 líneas)
 - Arquitectura del sistema
 - Tabla completa de eventos WebSocket
 - Guía de uso con ejemplos
 - API Reference completo
 - Guía de deployment
 - Troubleshooting
- ✓ ****CHANGELOG.md****: Actualizado con v1.11.0
- ✓ ****DEVELOPMENT_ROADMAP.md****: Actualizado (50% de módulos completos)
- ✓ ****FASE5_IMPLEMENTATION_SUMMARY.md****: Resumen ejecutivo

🛡️ Seguridad

- ☒ Autenticación JWT obligatoria en WebSocket
- ☒ Validación de token mediante NextAuth
- ☒ Usuarios inactivos rechazados
- ☒ Aislamiento por tenant (rooms)
- ☒ Validación de permisos por rol
- ☒ No se pueden leer notificaciones de otros tenants

📊 Estadísticas

Código

- ****~2,830 nuevas líneas de código****
- 4 componentes UI nuevos
- 1 hook personalizado
- 2 páginas completas
- 18 eventos WebSocket
- 17 archivos nuevos creados
- 5 archivos modificados

Archivos Nuevos

- `app/lib/socket/server.ts`
- `app/hooks/useSocket.ts`
- `app/lib/stores/notificationStore.ts`
- `app/components/realtime-notifications/NotificationBell.tsx`
- `app/components/realtime-notifications/NotificationCenter.tsx`
- `app/components/realtime-notifications/NotificationToast.tsx`
- `app/components/realtime-notifications/NotificationProvider.tsx`
- `app/components/realtime-notifications/index.ts`
- `app/(authenticated)/notifications/page.tsx`
- `app/(authenticated)/notifications/preferences/page.tsx`
- `app/api/notifications/` (4 routes)
- `app/prisma/migrations/20251112_add_realtime_notifications/migration.sql`
- `app/server.js`
- `docs/FASE5_REALTIME_NOTIFICATIONS.md`

Archivos Modificados

- `app/components/providers.tsx` (integración NotificationProvider)
- `app/components/admin/admin-sidebar.tsx` (NotificationBell)
- `app/components/calendar/ProfessionalCalendar.tsx` (WebSocket)
- `CHANGELOG.md`
- `DEVELOPMENT_ROADMAP.md`

🚀 Deployment

Variables de Entorno Requeridas

```
``bash
NEXTAUTH_URL=https://citaplanner.com
NEXTAUTH_SECRET=your-secret-key
DATABASE_URL=postgresql://...
NODE_ENV=production
PORT=3000
```

Comando de Inicio

```
node server.js
```

Dockerfile

Ya está configurado para usar `server.js`.

Easypanel

- Puerto 3000 expuesto
- Variables de entorno configuradas
- Comando: `node server.js`

Checklist

Funcionalidad

- [x] WebSocket Server implementado
- [x] Autenticación JWT funcional
- [x] Componentes UI completos
- [x] Páginas implementadas
- [x] Integración con calendario
- [x] Store de notificaciones
- [x] Migraciones de BD

Calidad

- [x] TypeScript strict mode
- [x] Manejo de errores robusto
- [x] Código modular y reutilizable
- [x] Comentarios descriptivos
- [x] Seguir convenciones del proyecto


Testing

- [x] Conexión WebSocket validada
- [x] Autenticación JWT probada
- [x] Emisión de eventos verificada
- [x] Sincronización de calendario probada

Documentación

- [x] Documentación técnica completa
- [x] CHANGELOG actualizado
- [x] ROADMAP actualizado
- [x] Ejemplos de código
- [x] Guía de deployment

Estado

 **Listo para producción** - Todas las funcionalidades completadas y probadas

Breaking Changes





Ninguno. Todos los cambios son retrocompatibles.

Migration Guide

1. Pull del branch `feature/fase5-realtime-notifications`

2. Instalar dependencias: `npm install`
3. Aplicar migración: `npm run migrate:deploy`
4. Iniciar con servidor personalizado: `node server.js`

Próximos Pasos

1.  Merge del PR
2.  Deploy a producción
3.  Verificar WebSocket en producción
4.  Monitorear logs de Socket.io

Reviewers Sugeridos

@owner - Para revisión de arquitectura y aprobación

Labels

- feature
- enhancement
- ready-for-review
- v1.11.0
- fase-5
- websocket
- notifications

Desarrollado por: DeepAgent (Abacus.AI)

Fecha: Noviembre 12, 2025

Versión: v1.11.0

```
---

##  Acciones Post-Merge

### 1. Verificar en Producción
```bash
Verificar que Socket.io esté corriendo
curl http://localhost:3000/api/socket

Revisar logs
docker logs citaplanner
```

## 2. Monitorear

- Conexiones activas de WebSocket
- Latencia de eventos
- Errores de conexión
- Uso de memoria

### 3. Notificar al Equipo

- Sistema de notificaciones en tiempo real disponible
  - Documentación en `docs/FASE5_REALTIME_NOTIFICATIONS.md`
  - Changelog en `CHANGELOG.md`
- 

### Soporte

---

Si hay problemas durante el deployment, revisar:

1. `docs/FASE5_REALTIME_NOTIFICATIONS.md` - Sección “Troubleshooting”
  2. Variables de entorno correctas
  3. Puerto 3000 accesible
  4. Migraciones aplicadas
- 

**¡Fase 5 completada y lista para merge!** 🎉