

# TodoList - VML Perito

---

## Completado

### Sistema de Notificaciones - Arquitectura Base

- Modelos de datos expandidos (NotificationConfig, Notification, NotificationQueue)
- Servicio central de notificaciones (NotificationService)
- Servicios de canales (InApp, Email, WhatsApp, SMS, Push)
- Controlador de APIs (NotificationController)
- Integración con WebSocket existente
- Cola de procesamiento con reintentos
- Sistema de plantillas con variables
- Targeting por roles y usuarios específicos

### Terminología Contact Center

- Actualización de terminología en frontend
- Actualización de terminología en backend
- Actualización de base de datos (seedRBAC.js)
- Consistencia en UI y mensajes
- Documentación de reglas de terminología

## En Progreso

### Sistema de Notificaciones - Implementación de Canales

#### Email Service

- Configurar proveedor de email (SendGrid/NodeMailer)
- Implementar plantillas HTML responsivas
- Configurar variables de entorno para API keys
- Implementar sistema de bounce/complaint handling
- Testing de entrega de emails

#### WhatsApp Service

- Integrar con Meta Business API o Twilio
- Configurar webhook para estados de entrega
- Implementar validación de números colombianos
- Configurar plantillas de mensajes aprobadas
- Testing de envío de mensajes

#### SMS Service

- Integrar con proveedor SMS (Twilio/AWS SNS)

- Implementar cálculo de costos por segmento
- Configurar límites de envío diario/mensual
- Implementar validación de números móviles
- Testing de entrega SMS

## Push Service

- Configurar Firebase Cloud Messaging (FCM)
- Implementar Service Worker para notificaciones web
- Crear sistema de registro de tokens push
- Implementar deep linking por tipo de notificación
- Testing de notificaciones push

## Integración con Sistema Existente

- Integrar NotificationService en coordinadorContactoController
- Integrar NotificationService en inspectionOrderController
- Configurar tipos de notificación estándar en base de datos
- Implementar cron job para procesamiento de cola
- Configurar webhooks para estados de entrega

## Frontend - Componentes de Notificación

- Actualizar NotificationMenu para usar nueva API
- Implementar marcado de notificaciones como leídas
- Crear componente de estadísticas de notificaciones
- Implementar filtros por tipo y estado
- Mejorar UI de notificaciones con tiempo relativo

## Pendiente

## Configuración de Proveedores

- Obtener credenciales de SendGrid/NodeMailer
- Configurar cuenta de Meta Business API para WhatsApp
- Configurar cuenta de Twilio para SMS
- Configurar proyecto Firebase para Push notifications
- Documentar proceso de configuración de proveedores

## Base de Datos

- Ejecutar migraciones para nuevos campos en notification tables
- Seed inicial de tipos de notificación estándar
- Seed inicial de configuraciones de notificación
- Configurar índices para optimización de consultas
- Implementar cleanup de notificaciones antiguas

## Monitoreo y Logs

- Implementar dashboard de métricas de notificaciones
- Configurar alertas para fallos de envío
- Implementar logs estructurados para debugging
- Crear reportes de entrega por canal
- Implementar rate limiting por usuario/canal

## Testing

- Unit tests para NotificationService
- Integration tests para cada canal
- End-to-end tests para flujos completos
- Load testing para cola de procesamiento
- Testing de failover y recuperación

## Documentación

- Manual de configuración de proveedores
- Guía de troubleshooting
- Documentación de APIs para desarrolladores
- Ejemplos de uso para cada tipo de notificación
- Guía de mejores prácticas

## Optimización

- Implementar cache para configuraciones frecuentes
- Optimizar consultas de base de datos
- Implementar batch processing para envíos masivos
- Configurar CDN para assets de email templates
- Implementar compression para payloads grandes

## Seguridad

- Implementar rate limiting por IP/usuario
- Configurar encriptación para datos sensibles
- Implementar audit log para cambios críticos
- Configurar validación de webhooks
- Implementar sanitización de contenido

## Futuras Mejoras

### Features Avanzadas

- Notificaciones programadas recurrentes
- A/B testing para plantillas de notificación
- Segmentación avanzada de usuarios
- Personalización por preferencias de usuario
- Integración con CRM externo

## Analytics

- Dashboard de métricas en tiempo real
- Reportes de engagement por canal
- Análisis de patrones de lectura
- Optimización automática de horarios de envío
- Predicción de mejores canales por usuario

## Escalabilidad

- Implementar procesamiento distribuido
- Configurar auto-scaling para colas
- Implementar sharding de base de datos
- Configurar multi-region deployment
- Implementar circuit breakers para proveedores

## Notas Técnicas

### Prioridades de Implementación

1. **Alta:** Email Service (crítico para notificaciones formales)
2. **Alta:** Integración con controladores existentes
3. **Media:** WhatsApp Service (importante para clientes)
4. **Media:** SMS Service (notificaciones urgentes)
5. **Baja:** Push Service (mejora UX pero no crítico)

### Dependencias Externas

- **SendGrid:** Para email service
- **Meta Business API:** Para WhatsApp service
- **Twilio:** Para SMS service (alternativa a AWS SNS)
- **Firebase:** Para push notifications
- **Cron Jobs:** Para procesamiento de cola programada

### Consideraciones de Costos

- **Email:** ~\$0.001 por email (SendGrid)
- **WhatsApp:** ~\$0.05 por mensaje (Meta API)
- **SMS:** ~\$0.10 por mensaje (Twilio Colombia)
- **Push:** Gratuito hasta cierto límite (Firebase)

### Estimación de Tiempo

- **Configuración de proveedores:** 2-3 días
- **Implementación de canales:** 1 semana
- **Integración con sistema:** 2-3 días
- **Testing completo:** 1 semana
- **Documentación:** 2-3 días

**Total estimado:** 3-4 semanas para implementación completa