

Especificaciones Técnicas - Proyecto Armonía

Sistema de Administración de Conjuntos Residenciales

1. Introducción

1.1 Descripción General

Armonía es una plataforma integral para la gestión de conjuntos residenciales, diseñada para facilitar la administración, comunicación y coordinación entre administradores, residentes y personal. La plataforma permite gestionar eficientemente todos los aspectos de la vida en comunidad, desde el inventario general del conjunto, administración de espacios comunes, hasta la gestión de finanzas, asambleas, recepción, seguridad y solicitudes.

Armonía: Transforma la Gestión Residencial a través de la Tecnología

Desarrolladores, tienen en sus manos la oportunidad de crear una solución integral que revolucionará la administración de conjuntos residenciales. Armonía no es solo otra aplicación web/móvil; es un ecosistema tecnológico completo que conectará comunidades enteras.

Su misión: Construir una plataforma robusta, escalable y de alto rendimiento que sirva simultáneamente a múltiples tipos de usuarios: empresas administradoras, administradores de conjuntos, residentes y personal de seguridad.

El desafío técnico: Desarrollar una arquitectura que soporte:

- **Portal Público Inteligente:** Landing page optimizada para SEO que convierta visitantes en usuarios, con UX/UI excepcional que comunique valor desde el primer segundo
- **Motor de Gestión Integral:** Sistema complejo de inventarios (propietarios, inmuebles, vehículos, mascotas) con relaciones de datos eficientes
- **Módulo de Democracia Digital:** Gestión de asambleas con votaciones en línea en tiempo real, cálculos automáticos de quórum y generación de reportes dinámicos
- **Sistema Financiero Avanzado:** Engine de facturación automatizada, gestión de presupuestos, generación de cuotas y integración con pasarelas de pago
- **Dashboard Personalizado para Residentes:** Portal intuitivo con gestión de pagos, registro de visitantes, reservas de espacios comunes y centro de notificaciones
- **Sistema de Comunicación Integrado:** Citofonía virtual vía WhatsApp/Telegram, correspondencia digital y alertas de seguridad
- **Portal de Seguridad IoT-Ready:** Integración con cámaras IP, gestión de accesos biométricos y minutas digitales

Modelo de Negocio Freemium: Implementar sistema de prueba gratuita (25 inmuebles/2 meses) con escalabilidad automática a planes básico, estándar y premium.

Este proyecto demanda excelencia técnica, arquitectura limpia, seguridad robusta y experiencia de usuario excepcional. Cada línea de código que escriban impactará directamente la calidad de vida de miles de familias.

1.2 Modelo de Negocio

La aplicación seguirá un modelo Freemium con los siguientes planes:

- **Plan Básico (Gratuito)**
 - Hasta 30 unidades residenciales
 - Gestión de propiedades y residentes
 - Portal básico de comunicaciones
 - Limitado a 1 año de históricos
- **Plan Estándar (\$USD 25/mes por conjunto)**
 - Hasta 30 unidades residenciales

- \$USD 1/mes por unidad residencial adicional.
- Todas las funcionalidades básicas
- Gestión completa de asambleas y votaciones
- Sistema de PQR avanzado
- Históricos de hasta 3 años
- **Plan Premium (\$USD 50/mes por conjunto)**
 - Hasta 80 Unidades residenciales
 - \$USD 1/mes por unidad residencial adicional.
 - Todas las funcionalidades estándar
 - Módulo financiero avanzado con generación automática de recibos
 - Personalización de la plataforma con logo y colores corporativos
 - Históricos completos e ilimitados
 - API para integración con otros sistemas
 - Soporte prioritario 24/7

1.3 Objetivos del Proyecto

- Proporcionar una plataforma moderna, intuitiva y completamente responsiva para la gestión eficiente de conjuntos residenciales.
- Mejorar la comunicación entre administradores, residentes y personal.
- Simplificar los procesos administrativos y reducir el trabajo manual.
- Integrar todos los aspectos de la administración de conjuntos en una única plataforma.
- Promover la transparencia en la gestión y facilitar el acceso a la información.
- Ofrecer una escalabilidad técnica para soportar desde pequeños conjuntos hasta grandes urbanizaciones.

2. Arquitectura del Sistema

2.1 Stack Tecnológico

- **Frontend:** Next.js 15 o superior, React 19 o superior, TypeScript, Tailwind CSS, Shadcn/UI
- **Backend:** Next.js API Routes, Serverless Functions
- **Base de Datos:** PostgreSQL17 o superior, con enfoque multi-tenant basado en esquemas
- **ORM:** Prisma 6.5.0 o superior
- **Autenticación:** JWT (JSON Web Tokens), bcrypt para el hash de contraseñas
- **Almacenamiento:** PostgreSQL para datos binarios (documentos) con posibilidad de migrar a almacenamiento en la nube
- **Validación:** Zod para validación de datos
- **Gráficos y Visualización:** Recharts para gráficos y estadísticas
- **Generación de PDFs:** pdfkit para reportes exportables
- **CI/CD:** GitHub Actions, Vercel o similares

2.2 Arquitectura Multi-tenant

Cada conjunto residencial tendrá un esquema dedicado en la base de datos PostgreSQL, siguiendo el formato `tenant_cjXXXX` donde XXXX es el ID del conjunto residencial con padding de ceros. La información de los conjuntos debe ser exportable e importable desde un archivo.

2.3 Componentes del Sistema

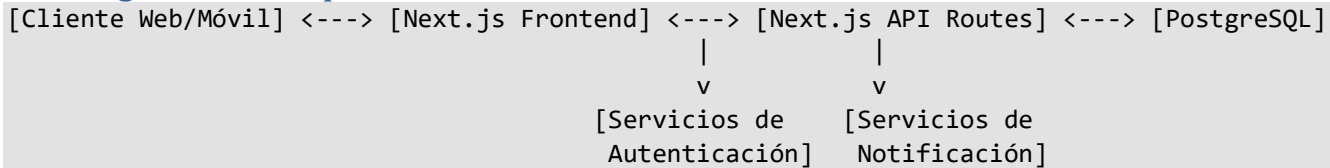
1. **Capa de Presentación (Frontend)**
 - Interfaces de usuario para diferentes roles (administrador, residente, staff)
 - Componentes reutilizables para garantizar consistencia
 - Diseño totalmente responsive (mobile-first)
2. **Capa de Aplicación (Backend)**
 - API RESTful para todas las operaciones
 - Lógica de negocio modular y extensible
 - Servicios para autenticación, autorización y gestión de sesiones

- Servicios para notificaciones y comunicaciones

3. Capa de Datos

- Modelo de datos relacional optimizado
- Estrategia multi-tenant por esquemas
- Índices y optimizaciones para consultas frecuentes
- Validación y consistencia de datos

2.4 Diagrama de Arquitectura



3. Funcionalidades Principales

3.1 Landing Page Comercial

- Presentación del producto con enfoque en beneficios
- Explicación clara de los tres planes disponibles (Básico, Estándar, Premium)
- Formulario básico de registro del conjunto
- Blog con contenido útil sobre administración de conjuntos
- Testimonios y casos de éxito
- Diseño atractivo con animaciones sutiles y microinteracciones
- Optimizada para SEO

3.2 Sistema de Autenticación y Autorización

- Registro y login multi-rol (administrador, residente, personal de recepción/Vigilancia)
- Recuperación y cambio de contraseña seguro
- Autenticación basada en JWT
- Autorización granular basada en roles y permisos
- Protección contra ataques comunes (CSRF, XSS, inyección SQL)
- Sesiones con tiempo de expiración configurable
- Historial de inicios de sesión

3.3 Panel de Control Global

- Dashboard personalizado según el rol del usuario
- Selector de idioma (español, inglés)
- Selector de moneda (COP, USD, EUR)
- Selector de rol (administrador, Residente, Recepción/Vigilancia)
- Modo oscuro/claro
- Notificaciones en tiempo real
- Menú lateral colapsable
- Barra de búsqueda global
- Accesos rápidos personalizables

3.4 Panel Administrador del Conjunto

- Dashboard personalizado del administrador: KPIs principales
 - Total de propiedades (casas/apartamentos)
 - Total residentes registrados
 - Estado de cartera general y por propiedad
 - Ejecución del presupuesto del conjunto
 - Ejecución de proyectos aprobados
 - Porcentaje de uso de servicios comunes
 - Gráficas de tendencias y comparativas

- PQRs, total, atendidos, pendientes

3.4.1 Gestión de Inventario

- Registro y seguimiento de propiedades
- Registro de propietarios y residentes
- Gestión de vehículos y parqueaderos
- Registro y control de mascotas
- Inventario de bienes comunes

3.4.2 Gestión de Asambleas

- Programación y convocatoria de asambleas
- Verificación de quórum y asistencia
- Sistema de votaciones con resultados en tiempo real
- Elaboración y firma digital de actas
- Repositorio de actas y documentos

3.4.3 Gestión de Servicios Comunes

- Reserva de áreas comunes (salón comunal, BBQ, zonas deportivas)
- Calendario de disponibilidad de servicios
- Estadísticas de uso de servicios
- Reglas y requisitos para cada servicio
- Sistema de reservas con confirmación automática

3.4.4 Gestión Financiera

- Presupuestos anuales y seguimiento
- Generación de cuotas ordinarias y extraordinarias
- Registro y seguimiento de pagos
- Generación de recibos y facturas
- Reportes financieros personalizables
- Integración con pasarelas de pago (próxima versión)
- Recordatorios automáticos de pagos pendientes

3.4.5 Sistema de PQR (Peticiónes, Quejas y Reclamos)

- Creación y seguimiento de solicitudes
- Categorización por tipo y prioridad
- Asignación de responsables
- Notificaciones de estado
- Historial de comunicaciones
- Indicadores de tiempo de respuesta
- Encuestas de satisfacción

3.4.6 Comunicaciones

- Tablón de anuncios
- Mensajería interna (Uso de WhatsApp o Telegram)
- Notificaciones por correo electrónico
- Comunicados oficiales
- Calendario de eventos comunitarios

3.4.7 Configuración

- Datos básicos del conjunto
- Logotipo y personalización visual
- Datos bancarios y medios de pago
- Certificaciones y documentos legales
- Configuración de módulos y permisos

3.5 Panel de Usuario Residente

- Dashboard con información relevante: KPIs personalizados
 - o Total residentes en la propiedad
 - o Estado de cartera actual
 - o Resumen de pagos del año en curso
 - Cuotas de administración
 - Cuotas Extraordinarias
 - Servicios adicionales
 - o Gráfico de gastos mensuales
 - o Próximos vencimientos
 - o Uso de servicios comunes
 - o PQRs de los residentes de su propiedad, total, resuelto y pendientes.
- Consulta de estado de cuenta
- Historial de pagos realizados
- Visualización de cuotas pendientes
- Reserva de servicios comunes
- Participación en asambleas virtuales
- Creación y seguimiento de PQR
- Directorio de residentes (opcional)
- Notificaciones de eventos y comunicados

3.6 Panel de Recepción y Vigilancia

- Dashboard con información relevante: KPIs personalizados con vista rápida de información crítica
 - o Visitantes actuales en el conjunto
 - o Servicios comunes en uso
 - o Alertas pendientes
 - o Estado de cámaras
 - o Novedades del turno anterior en minuta virtual.
 - o PQRs asignados a recepción/y vigilancia
- Registro de visitantes y proveedores
- Control de correspondencia y paquetes
- Citofonía virtual para verificación de visitas (Uso de WhatsApp o Telegram)
- Registro de incidentes de seguridad
- Minuta de novedades diarias
- Alertas y notificaciones
- Control de acceso a zonas restringidas
- Reportes de vigilancia

3.7 Panel Administrador de la Aplicación

- Gestión de conjuntos residenciales
- Monitoreo de uso y rendimiento
- Estadísticas y métricas de uso
- Gestión de licencias y planes
- Soporte y asistencia técnica
- Configuración global del sistema

4. Modelo de Datos

4.1 Entidades Principales

- **Usuario:** Datos de acceso y permisos
- **Conjunto Residencial:** Información básica del conjunto
- **Propiedad:** Unidades habitacionales y comerciales

- **Residente:** Habitantes del conjunto
- **Propietario:** Dueños de propiedades
- **Asamblea:** Reuniones de copropietarios
- **Servicio:** Áreas y servicios comunes
- **Cuota:** Pagos ordinarios y extraordinarios
- **Pago:** Registro de transacciones
- **PQR:** Peticiones, quejas y reclamos
- **Documento:** Archivos y documentos del conjunto
- **Visita:** Registro de visitantes
- **Personal:** Empleados y contratistas

Mas las necesarias para el buen funcionamiento de la solución Armonía.

4.2 Relaciones Clave

- Un conjunto tiene muchas propiedades
- Una propiedad pertenece a uno o varios propietarios
- Una propiedad puede tener muchos residentes
- Un usuario puede estar asociado a uno o más conjuntos
- Un residente puede crear múltiples PQR
- Un conjunto puede programar múltiples asambleas
- Una propiedad tiene asociadas múltiples cuotas
- Un servicio pertenece a un conjunto

5. Interfaces de Usuario

5.1 Diseño General

- Diseño limpio y moderno con enfoque en usabilidad
- Paleta de colores principal: Indigo (#4f46e5) y blanco (ffffff)
- Soporte para modo oscuro/claro
- Interfaz responsive para todos los dispositivos
- Tiempo de carga optimizado (LCP < 2.5s)
- Animaciones sutiles para mejorar la experiencia

5.2 Componentes UI Reutilizables

- Sistema de diseño basado en Shadcn/UI
- Componentes altamente personalizables
- Consistencia visual en toda la plataforma
- Accesibilidad (WCAG 2.1 AA)
- Soporte para RTL (próxima versión)

5.3 Plantillas de Páginas

- Layouts específicos para cada tipo de panel
- Patrones de navegación intuitivos
- Breadcrumbs para facilitar la navegación
- Organización jerárquica de la información
- Componentes de carga y estados vacíos diseñados

6. Seguridad y Privacidad

6.1 Prácticas de Seguridad

- Encriptación de datos sensibles
- Protección contra ataques CSRF, XSS y SQL Injection
- Rate limiting para prevenir ataques de fuerza bruta
- Validación de datos en cliente y servidor
- Auditoría de accesos y cambios

- Backups automáticos diarios

6.2 Privacidad de Datos

- Cumplimiento con regulaciones de protección de datos
- Políticas claras de privacidad y términos de uso
- Opción de anonimización de datos al eliminar cuentas
- Exportación de datos personales
- Consentimiento explícito para uso de cookies

7. Pruebas y Calidad

7.1 Estrategia de Pruebas

- Pruebas unitarias para componentes y servicios críticos
- Pruebas de integración para flujos principales
- Pruebas e2e con Cypress
- Pruebas de rendimiento y carga
- Pruebas de compatibilidad con navegadores

7.2 Control de Calidad

- Revisión de código mediante pull requests
- Análisis estático de código
- Monitoreo de errores en producción
- Retroalimentación continua de usuarios
- Métricas de calidad (cobertura de pruebas, complejidad ciclomática)

8. Despliegue y Operaciones

8.1 Infraestructura

- Despliegue en la nube (AWS, Azure o GCP)
- Arquitectura serverless donde sea posible
- Base de datos PostgreSQL gestionada
- CDN para activos estáticos
- Balanceador de carga para alta disponibilidad

8.2 CI/CD

- Integración continua con GitHub Actions
- Despliegue continuo en entornos de desarrollo y pruebas
- Despliegue controlado en producción
- Rollbacks automatizados en caso de fallos
- Entornos de desarrollo, staging y producción

8.3 Monitoreo

- Monitoreo de disponibilidad y tiempos de respuesta
- Alertas automáticas para incidentes
- Dashboards de métricas operativas
- Análisis de logs centralizados
- Monitoreo de experiencia del usuario real
-

8.4 Mejoras Continuas

- Optimización de rendimiento
- Mejoras de usabilidad basadas en feedback
- Nuevas funcionalidades según necesidades del mercado
- Expansión internacional con soporte multilenguaje
- Integraciones con sistemas externos (contabilidad, bancos, etc.)

9. Consideraciones Especiales

9.1 Accesibilidad

- Conformidad con WCAG 2.1 nivel AA
- Soporte para lectores de pantalla
- Navegación por teclado
- Alto contraste y fuentes ajustables
- Textos alternativos para imágenes

9.2 Internacionalización

- Soporte inicial para español e inglés
- Estructura preparada para fácil expansión a otros idiomas
- Formatos de fecha y número según localización
- Soporte para múltiples monedas

9.3 Escalabilidad

- Arquitectura diseñada para escalar horizontalmente
- Optimización de consultas a base de datos
- Caché estratégica para datos frecuentemente accedidos
- Lazy loading de componentes y datos
- Particionamiento de base de datos para grandes volúmenes

10. Conclusión

El proyecto Armonía tiene un enfoque en usabilidad, rendimiento y escalabilidad. La plataforma está diseñada para satisfacer las necesidades de diversos tipos de conjuntos residenciales, desde pequeñas comunidades hasta grandes urbanizaciones, ofreciendo una solución integral para la gestión eficiente de todos los aspectos relacionados con la administración de propiedades horizontales.

Anexo: Convenciones de Codificación

Nombrado

- **Archivos:** PascalCase para componentes React (.tsx), camelCase para utilidades (.ts)
- **Funciones:** camelCase (ej. getUserData)
- **Componentes React:** PascalCase (ej. UserProfile)
- **Variables:** camelCase (ej. userData)
- **Constantes:** UPPER_SNAKE_CASE (ej. API_BASE_URL)
- **Interfaces/Types:** PascalCase con prefijo I para interfaces (ej. IUserData)
- **Endpoints API:** kebab-case (ej. /api/user-profile)

Estructura de Directorios

```
src/  
  app/                # Next.js App Router  
    (auth)/           # Rutas protegidas  
      dashboard/      # Panel principal  
      resident/       # Panel de residente  
    (public)/         # Rutas públicas  
    api/              # API Routes  
  components/         # Componentes React  
    ui/               # Componentes UI básicos  
    admin/            # Componentes de administrador  
    auth/             # Componentes de autenticación  
  context/            # Context API  
  hooks/              # Custom hooks  
  lib/                # Utilidades y servicios  
  models/             # Tipos y modelos  
  styles/             # Estilos globales  
  utils/              # Funciones utilitarias
```

Estilo de Código

- Usar TypeScript con tipos estrictos
- Preferir funciones arrow para componentes
- Usar React Hooks para gestión de estado
- Documentar funciones y componentes complejos
- Mantener componentes pequeños y con responsabilidad única
- Usar Prettier y ESLint para formateo y calidad de código