**🏠 Prompt Maestro - Novaterra Lifestyle**

**📋 Contexto del Proyecto**

**Proyecto:** Novaterra Lifestyle - Plataforma inmobiliaria premium en Cantabria, España  
**Tipo:** Aplicación web React con autenticación Firebase  
**Estado:** ✅ MVP funcional desplegado en Netlify  
**URL:** https://novaterra-lifestyle.netlify.app

**🏗️ Arquitectura Técnica**

**Stack Principal**

* **Frontend:** React 18 + Vite
* **Routing:** React Router DOM v7
* **Estilos:** CSS vanilla + variables CSS + Sass
* **Autenticación:** Firebase Auth
* **Deploy:** Netlify (automático desde GitHub)
* **i18n:** react-i18next (5 idiomas: ES, EN, DE, RU, MK)

**Estructura de Carpetas**

src/

├── components/ # Header, Footer, Dashboard

├── pages/ # Home, Properties, Services, etc.

├── hooks/ # useAuth.js

├── config/ # firebase.js, imageConfig.js

├── i18n/ # config.js + locales/

├── utils/ # roles.js

├── App.jsx # Router principal

└── main.jsx # Entry point

**Sistema de Autenticación**

* **Usuarios NO logueados:** Ven páginas preview (limitadas) + CTA de registro
* **Usuarios logueados:** Acceso completo + Dashboard + servicios premium
* **Roles:** USER, ADMIN, CREATOR (definidos en utils/roles.js)

**🎨 Sistema de Diseño**

**Paleta de Colores**

* **Primario:** Negro premium (#1a1a1a, #2a2a2a)
* **Acento:** Dorado elegante (#d4af37, #b8941f)
* **Neutros:** Escalas de grises con contraste WCAG AA
* **Estados:** Verde (#28a745), Amarillo (#ffc107), Rojo (#dc3545)

**Tipografía**

* **Títulos:** Playfair Display (serif elegante)
* **Cuerpo:** Inter (sans-serif moderna)
* **Escala:** Responsiva 12px-60px

**Componentes UI**

* **Cards:** preview-card (con indicador "🔒 VISTA PREVIA")
* **Botones:** preview-button (gradiente naranja), cta-button (negro/dorado)
* **Modales:** Sistema auth completo con validaciones
* **Navegación:** Diferente para logueados/no-logueados

**🌐 Funcionalidades Implementadas**

**✅ Completadas**

* ✅ Header con carrusel de imágenes de Cantabria
* ✅ Sistema de navegación dual (preview vs completo)
* ✅ Autenticación Firebase completa
* ✅ Páginas preview con CTAs de conversión
* ✅ Dashboard para usuarios registrados
* ✅ Sistema de roles y permisos
* ✅ Internacionalización (5 idiomas)
* ✅ Footer con información completa
* ✅ Deploy automático en Netlify
* ✅ CSS responsive y accesible

**🚧 En Desarrollo**

* [ ] Gestión completa de propiedades
* [ ] Sistema de reservas
* [ ] Panel de administración
* [ ] Integración con APIs externas
* [ ] Sistema de pagos
* [ ] Chat en vivo

**📝 Convenciones de Código**

**React**

* Componentes funcionales con hooks
* Props destructuring
* useCallback para funciones
* useMemo para cálculos costosos
* Manejo de errores robusto

**CSS**

* Variables CSS en :root
* BEM methodology para clases
* Mobile-first responsive
* Transiciones suaves (300ms)
* Focus rings para accesibilidad

**Naming**

* Archivos: PascalCase.jsx
* Componentes: PascalCase
* Funciones: camelCase
* Variables CSS: --kebab-case
* Clases CSS: kebab-case

**🔧 Comandos Útiles**

# Desarrollo

npm run dev # Servidor local

npm run build # Build producción

npm run preview # Preview del build

# Deploy

git add . && git commit -m "feat: descripción" && git push origin main

# Netlify despliega automáticamente

# Idiomas

# Editar: src/i18n/locales/[idioma].json

# Usar: {t('key.subkey')} en componentes

**🎯 Instrucciones para Claude**

**Cuando trabajo en Novaterra:**

1. **Comportarse como experto** en el tema específico que estemos tratando
2. **Código siempre en el chat**, no en artifacts separados
3. **Mantener consistencia** con la arquitectura existente
4. **Usar las convenciones** establecidas
5. **Considerar i18n** en nuevas funcionalidades
6. **Priorizar accesibilidad** y responsive design
7. **Seguir el sistema de colores** negro/dorado
8. **Integrar con Firebase** cuando sea necesario

**Tipos de ayuda frecuentes:**

* Nuevos componentes React
* Fixes de bugs y optimizaciones
* Nuevas páginas con rutas
* Mejoras de UX/UI
* Integración de APIs
* Funcionalidades de admin

**📌 Notas Importantes**

* **Firebase:** Configuración en src/config/firebase.js
* **Imágenes header:** public/images/header/ (configurar en imageConfig.js)
* **Rutas protegidas:** Verificar isAuthenticated en App.jsx
* **Estilos preview:** Ya incluidos en App.css
* **Deploy:** Automático, no requiere configuración adicional

**🚀 Próximos Pasos Sugeridos**

1. **Panel Admin:** Gestión de propiedades y usuarios
2. **Sistema de Reservas:** Calendario y pagos
3. **CRM Básico:** Leads y seguimiento
4. **Blog/Noticias:** Content marketing
5. **SEO/Analytics:** Optimización y métricas

*Última actualización: Enero 2025 | Estado: MVP Funcional*