

PLATAFORMA INMUEBLES

Guía de Referencia para Claude.ai

SISTEMA INTEGRAL DE GESTIÓN

- Plataforma completa de inversiones inmobiliarias
 - Integración bancaria automatizada (Bankinter)
- Análisis financiero avanzado y gestión de contratos

STACK TECNOLÓGICO

Backend:

- FastAPI (Python) - API REST principal
- SQLAlchemy + SQLite - ORM y base de datos
- Selenium WebDriver - Scraping bancario
- Pandas/NumPy - Análisis financiero

Frontend:

- Next.js 14 (React) - App web moderna
- TypeScript - Tipado estático
- Tailwind CSS - Sistema de diseño
- PWA - Aplicación instalable

Deployment:

- Frontend: Vercel (inmuebles-web.vercel.app)
- Backend: Render (inmuebles-backend-api.onrender.com)

MODELOS DE DATOS PRINCIPALES

ENTIDADES CORE:

- **User:** Usuarios con autenticación JWT
- **Property:** Propiedades con datos financieros completos
- **FinancialMovements:** Movimientos bancarios clasificados
- **RentalContract:** Contratos de alquiler y gestión inquilinos
- **MortgageDetails:** Hipotecas variables con revisiones
- **ClassificationRules:** Reglas automáticas de categorización
- **BankConnection:** Conexiones bancarias (GoCardless)
- **EuriborRate:** Tasas históricas para hipotecas
- **PaymentRule:** Configuración de ventanas de pago

RELACIONES CLAVE:

- User → Properties (1:N)
- Property → FinancialMovements (1:N)
- Property → RentalContracts (1:N)
- Property → MortgageDetails (1:1)
- MortgageDetails → MortgageRevisions (1:N)
- RentalContract → TenantDocuments (1:N)

ENDPOINTS PRINCIPALES DE LA API

AUTENTICACIÓN:

- POST /auth/login - Login usuario
- POST /auth/register - Registro usuario

PROPIEDADES:

- GET /properties/ - Listar propiedades
- POST /properties/ - Crear propiedad
- PUT /properties/{id} - Actualizar propiedad

ANÁLISIS FINANCIERO:

- GET /financial-movements/ - Movimientos
- POST /financial-movements/upload - Importar
- GET /analytics/ - Métricas y análisis
- GET /cashflow/ - Flujos de caja

CONTRATOS E HIPOTECAS:

- GET /rental-contracts/ - Contratos alquiler
- GET /mortgage-details/ - Detalles hipotecas
- POST /euribor-rates/ - Tasas Euribor

INTEGRACIÓN BANCARIA:

- POST /bankinter-real/scrape - Scraping
- GET /bank-integration/ - Estado conexiones
- POST /bankinter-upload/ - Upload datos

UTILIDADES:

- GET /notifications/ - Notificaciones
- POST /classification-rules/ - Reglas
- GET /mortgage-calculator/ - Calculadora

GUÍA DE DESARROLLO

COMANDOS PRINCIPALES:

Backend Development:

- `uvicorn app.main:app --reload --port 8000`
- `python bankinter_auto_download.py`
- `pip install -r requirements.txt`

Frontend Development:

- `cd inmuebles-web && npm run dev`
- `npm run build && npm run lint`
- `npm install`

Deployment:

- `./deploy-auto.bat`
 - `./deploy-inmuebles-frontend.bat`
 - `./deploy-backend-fixed.bat`
- `BANKINTER_USERNAME=75867185`
 - `BANKINTER_PASSWORD=Motoreta123$`
 - `DATABASE_URL=sqlite:///./app.db`
 - `SECRET_KEY=your-secret-key`

CARACTERÍSTICAS ESPECIALES:

- Scraping automatizado de Bankinter
- Clasificación inteligente de transacciones
- Cálculos ROI y rentabilidad automáticos
- Proyecciones hipotecarias con Euribor
- Gestión completa de contratos PDF
- Análisis comparativo de inversiones
- PWA instalable en dispositivos móviles

TEMPLATES DE PROMPTS PARA CLAUDE.AI

NUEVO ENDPOINT:

Crear endpoint POST /[recurso]/ en FastAPI para [funcionalidad]. Usar modelo [Modelo] de models.py. Incluir validación y autenticación JWT.

COMPONENTE REACT:

Crear componente React TypeScript para [funcionalidad] usando Tailwind CSS. Integrar con API endpoint /[ruta]/. Responsive design.

ANÁLISIS FINANCIERO:

Implementar cálculo de [métrica] en analytics.py. Usar datos de FinancialMovement y Property. Devolver JSON con métricas.

SCRAPING BANCARIO:

Mejorar bankinter_client.py para extraer [datos]. Usar Selenium WebDriver. Clasificar automáticamente con ClassificationRule.

BASE DE DATOS:

Añadir campo [campo] al modelo [Modelo] en models.py. Crear migración y actualizar endpoints relacionados.

FRONTEND DASHBOARD:

Crear página dashboard para [funcionalidad] en app/(protected)/[ruta]/. Usar contextos y hooks existentes.

PROYECTO ESTRUCTURADO PARA DESARROLLO EFICIENTE

Usa estos templates como base para solicitudes específicas en Claude.ai