

Universidad Tecnológica Nacional  
Regional La Plata

## **MKalpin - Sistema de Negocios Inmobiliarios**

**Carrera:** Ingeniería en Sistemas de Información

**Asignatura:** Desarrollo de Software Cloud

### **Integrantes:**

Capra, Valentina	32519
Kalpin, Sofía	32311
Gardiner, Ariadna	31017
Ransan, Magali	31513

**Fecha de Entrega:** 01 de Diciembre de 2025

# Índice

<b>MKalpin - Sistema de Gestión de Negocios Inmobiliarios</b>	<b>2</b>
<b>Enlaces del Proyecto</b>	<b>2</b>
<b>Descripción del Proyecto</b>	<b>2</b>
Objetivos principal:	2
Áreas de Gestión:	2
Funcionalidad	2
<b>Gestión del Proyecto</b>	<b>2</b>
<b>Arquitectura Tecnológica</b>	<b>3</b>
Stack Tecnológico	3
<b>Características Técnicas</b>	<b>4</b>
Backend (API REST)	4
Capas de Protección (Middleware)	4
Manejo de Errores	4
Base de Datos	4
Modelo de Propiedades (Property.js)	4
Modelo de Usuarios (User.js)	5
Frontend	5
Gestión de Archivos Multimedia	5
Servicio de Emails (Notificaciones)	6
<b>Infraestructura y Despliegue</b>	<b>6</b>
Estrategia de Despliegue en Render	6
<b>Funcionalidades Detalladas</b>	<b>6</b>
Para el Usuario Final (Cliente)	6
Para el Administrador (Back-office)	7
<b>Galería del Proyecto</b>	<b>7</b>
<b>Usuario Final</b>	<b>8</b>
1. Página de Inicio	8
2. Detalle de Propiedad	9
3. Tasación	10
<b>Administrador</b>	<b>11</b>
1. Inicio	11
2. Propiedades	11
3. Configuración	12
<b>Conclusión</b>	<b>12</b>

# MKalpin - Sistema de Gestión de Negocios Inmobiliarios

## Enlaces del Proyecto

- **Aplicación Desplegada (Web):** <https://mkalpin-frontend-web.onrender.com/>
- **Demostración en Video:**  
[https://drive.google.com/file/d/1p1Xwf3YGSORXEEguS5rb\\_BivRxsNw5Le/view?usp=drive\\_link](https://drive.google.com/file/d/1p1Xwf3YGSORXEEguS5rb_BivRxsNw5Le/view?usp=drive_link)
- **Repositorio:** <https://github.com/MaguiRansan/Desarrollo-de-Software-Cloud>

## Descripción del Proyecto

**MKalpin** es una plataforma web integral para la Gestión de Negocios Inmobiliarios. La aplicación centraliza operaciones de alquiler, venta y administración de propiedades, facilitando la comunicación entre la inmobiliaria y los clientes mediante notificaciones automatizadas.

### Objetivos principal:

- Resolver una necesidad de negocio real en el sector inmobiliario.
- Centralizar y optimizar la gestión de propiedades.
- Demostrar la capacidad de construir una solución cloud, garantizando escalabilidad y alta disponibilidad.

### Áreas de Gestión:

El sistema maneja de forma eficiente dos operaciones clave del negocio:

- **Alquileres:** alquileres de propiedades tanto a largo como a corto plazo/temporales (con gestión de fechas).
- **Ventas:** gestión de catálogo de propiedades para la venta.

### Funcionalidad

La aplicación ofrece paneles personalizados para la gestión de tareas, destacando las siguientes para el usuario Administrador:

- Registro y gestión completa de propiedades.
- Acceso centralizado a las propiedades.
- Control sobre la información institucional y perfil de acceso.

## Gestión del Proyecto

Para la organización y seguimiento del desarrollo, implementamos una metodología de trabajo ágil, gestionada a través de Jira. Esto nos permitió:

- **División de tareas:** Crear historias de usuario y tareas técnicas para cada funcionalidad (Backend, Frontend, Despliegue)

- **Planificación de Sprints:** Organizar el trabajo en ciclos de desarrollo cortos y enfocados, facilitando la entrega incremental de valor.
- **Seguimiento transparente:** Utilizar tableros (Scrum) para visualizar el progreso del equipo, permitiendo identificar cuellos de botella y gestionar la carga de trabajo de cada miembro del equipo.
- **Colaboración:** Mantener una comunicación centralizada sobre el estado y los detalles de cada tarea.

## Arquitectura Tecnológica

El sistema fue desarrollado bajo un enfoque Cloud - Native y distribuido, utilizando tecnologías modernas y servicios en la nube para garantizar un rendimiento óptimo, escalabilidad y una experiencia de usuario fluida.

**MKalpin** es una aplicación moderna **Full-Stack JavaScript (MERN: MongoDB, Express, React, Node.js)** que destaca por:

1. **Seguridad:** Implementada en múltiples niveles (Base de datos, API, Servidor).
2. **Rendimiento:** Uso de índices en base de datos, compresión Gzip y CDN para imágenes (Cloudinary).
3. **Experiencia de Usuario:** Mapas interactivos, calendarios dinámicos y login social con Google.
4. **Escalabilidad:** Lista para crecer gracias a su arquitectura en la nube (Render + MongoDB Atlas).

## Stack Tecnológico

Componente	Tecnología	Rol en la Arquitectura
Frontend	<b>React + Tailwind</b>	Interfaz de usuario dinámica, responsiva y moderna (SPA - Single Page Application)
Backend (API)	<b><u>Node.js</u> + Express</b>	Servidor de aplicación para gestionar la lógica de negocio, la API REST, la seguridad y la autenticación.
Base de Datos	<b>MongoDB Atlas</b>	Base de Datos NoSQL alojada en la nube, utilizada para almacenar de manera flexible toda la información transaccional y de propiedades.
Almacenamiento de Media	<b>Cloudinary</b>	Servicio en la nube para el almacenamiento, optimización y entrega de imágenes y otros archivos multimedia de las

		propiedades.
--	--	--------------

## Características Técnicas

### Backend (API REST)

El backend de la aplicación fue desarrollado en **Node.js** y **Express**, el servidor gestiona la lógica de negocio, la seguridad y la comunicación con la base de datos.

El archivo `server.js` revela que la aplicación no es un servidor básico, sino que implementa varias capas de Middleware para seguridad y rendimiento antes de procesar cualquier solicitud.

#### Capas de Protección (Middleware)

1. **Protección contra Ataques (Helmet):** Se utiliza `helmet` para configurar cabeceras HTTP seguras, protegiendo la app de vulnerabilidades web comunes.
2. **Control de Acceso (CORS Dinámico):** El sistema verifica el origen de cada petición. Solo permite peticiones desde `localhost` (para desarrollo) y desde la URL de producción (`process.env.FRONTEND_URL`). Si otro sitio web intenta usar tu API, será bloqueado automáticamente.
3. **Prevención de DDoS (Rate Limiting):** Se utiliza `express-rate-limit`.
  - En producción, limita a 100 solicitudes cada 15 minutos por IP. Esto evita que un usuario malintencionado o un bot sature el servidor.
4. **Optimización (Compression):** Usa Gzip para comprimir las respuestas JSON, haciendo que la aplicación cargue más rápido en redes móviles lentas.

#### Manejo de Errores

El servidor tiene un sistema robusto para que no colapse ante fallos inesperados. Utiliza `process.on('uncaughtException')` y `process.on('unhandledRejection')` para capturar errores graves y cerrarse ordenadamente si es necesario, evitando estados corruptos en la memoria.

### Base de Datos

El diseño de la base de datos está en MongoDB, presentando una conexión con **MongoDB Atlas** mediante Mongoose (ODM) para su modelado.

#### Modelo de Propiedades (Property.js)

Este es el núcleo del negocio. El esquema está diseñado para soportar Venta, Alquiler y Alquiler Temporal en una sola colección.

- **Geolocalización:** Almacena `latitud` y `longitud` y crea un virtual llamado `coordenadas` para facilitar su uso en los mapas del frontend.

- **Validación de Fechas:** Implementa lógica de negocio directamente en la base de datos. Por ejemplo, al guardar una disponibilidad, valida automáticamente que la `endDate` (fecha fin) sea posterior a la `startDate` (fecha inicio).
- **Índices de Búsqueda:** Se han creado índices (`index: true`) en campos clave como `precio`, `barrio`, `habitaciones` y `transaccionTipo`. Esto hace que los filtros de búsqueda en la web sean casi instantáneos, incluso con miles de propiedades.
- **Búsqueda de Texto:** Tiene configurado un índice de texto completo (`text index`) sobre título, descripción y dirección, permitiendo búsquedas tipo Google dentro de la inmobiliaria.

## Modelo de Usuarios (User.js)

- **Roles y Permisos:** Utiliza un sistema numérico (`idrol`) para definir permisos:
  - 1: Propietario
  - 2: Inquilino (por defecto)
  - 3: Administrador
- **Seguridad de Contraseñas:**
  - Nunca guarda la contraseña real. Usa `bcryptjs` con un "salt" de 12 rondas para encriptarla (`hash`) antes de guardarla.

## Frontend

El frontend fue desarrollado en React, es una aplicación rica e interactiva que combina varias librerías:

- **Diseño Híbrido:** Utiliza Bootstrap (`react-bootstrap`) para la estructura de rejillas clásica y Tailwind CSS para estilos modernos y personalizados de forma rápida.
- **Mapas Interactivos:** Implementa `react-leaflet` y `leaflet`. Esto es fundamental para una inmobiliaria, permitiendo mostrar pines en el mapa, zonas, etc., sin depender de la API paga de Google Maps.
- **Gestión de Calendarios:** Usa `react-big-calendar` y `react-datepicker`, esenciales para el módulo de Alquileres Temporales, permitiendo seleccionar fechas de entrada y salida visualmente.
- **Multimedia:**
  - `react-image-gallery` y `slick-carousel`: Para crear galerías de fotos atractivas de las casas.
  - `framer-motion`: Para añadir animaciones suaves a la interfaz, mejorando la experiencia de usuario (UX).
- **Conexión Real-time:** La presencia de `@microsoft/signalr` sugiere que la aplicación tiene (o tendrá) capacidades de tiempo real, como notificaciones instantáneas o un chat de soporte.

## Gestión de Archivos Multimedia

Se utiliza **Cloudinary** para el almacenamiento y optimización de imágenes en la nube.

- **Implementación:** Se usa la librería `multer-storage-cloudinary` junto con `multer` para la
 

Hogar

mkalpin

propiedades

tasaciones

usuarios

carga directa de archivos desde la API hacia Cloudinary, gestionando eficientemente las fotos de las propiedades.

## Servicio de Emails (Notificaciones)

La aplicación cuenta con un sistema de notificaciones por correo electrónico para confirmar consultas y contactos.

- **Motor de Envío:** Aunque se utiliza la API de SendGrid como proveedor de servicio (SMTP), la implementación en código se realiza a través de la librería **nodemailer**.
- **Configuración:** Se utiliza un **transporter** configurado con variables de entorno, lo que permite cambiar de proveedor sin reescribir el código, manteniendo la seguridad de las credenciales.
- **Casos de Uso:**
  1. **Formulario de Contacto General:** Envío de consultas generales.
  2. **Consultas por Propiedad:** Correos específicos que incluyen el título de la propiedad de interés.
  3. **Alquiler Temporal:** Solicitudes complejas que incluyen fechas de entrada/salida y cantidad de personas.

## Infraestructura y Despliegue

La aplicación utiliza una arquitectura de **"Monorepo"**, lo que significa que tanto el código del Frontend como el del Backend viven en el mismo repositorio, pero se despliegan como servicios independientes en la nube de **Render**.

### Estrategia de Despliegue en Render

Según el archivo de configuración **render.yaml**, la infraestructura se divide en dos servicios web que se comunican entre sí:

- **Servicio Backend (mkalpin-backend):**
  - **Entorno:** Node.js en modo producción.
  - **Variables de Entorno Críticas:** No se guardan en el código (por seguridad), sino que se inyectan en el servidor: **MONGODB\_URI** (base de datos), **JWT\_SECRET** (seguridad de sesiones) y credenciales de **CLOUDINARY** (imágenes) y **GOOGLE** (login social).
  - **Conexión:** Expone la API a través de <https://mkalpin-backend.onrender.com>.
- **Servicio Frontend (mkalpin-frontend):**
  - **Construcción:** Ejecuta **npm run build** para transformar el código React en archivos estáticos (HTML/CSS/JS) optimizados.
  - **Servidor Web:** Utiliza el paquete **serve** para entregar estos archivos estáticos al navegador del usuario de manera eficiente.

## Funcionalidades Detalladas

### Para el Usuario Final (Cliente)

- **Búsqueda Avanzada:** Filtros en tiempo real por ubicación, tipo de operación y comodidades, permitiendo encontrar inmuebles específicos al instante.
- **Visualización:** Galerías de imágenes optimizadas y mapas interactivos para ubicar propiedades.
- **Gestión de Contacto:** Formularios específicos para "Alquiler Temporal" que validan fechas y disponibilidad automáticamente.

### Para el Administrador (Back-office)

- **Dashboard de Propiedades:** CRUD completo (Crear, Leer, Actualizar, Borrar) de propiedades con subida de imágenes integrada a Cloudinary.
- **Gestión Multimedia:** Subida y optimización automática de imágenes integrada con Cloudinary.
- **Administración Institucional:** Control total sobre la información pública de la empresa y gestión de perfiles de acceso.

## Galería del Proyecto


### Usuario Final

#### 1. Página de Inicio

*Vista principal con integración de mapas y barra de filtros avanzada.*







**\$150,000**

### Hermoso Apartamento en Playa del Carmen

Playa del Carmen, Playa del Carmen, Quintana Roo

<b>3</b>	<b>2</b>	<b>65m<sup>2</sup></b>
Dorm.	Baños	Área

[Ver Detalles](#)



MARCELO KALPIN NEGOCIOS INMOBILIARIOS

Mat. C.U.C.I.C.B.A 7025



Enlaces rápidos

Inicio

Venta

Alquiler

Alquiler Temporario

Tasaciones

Contacto

Contacto

 Florida 142, oficina: 8° i., CABA

 (011) 5669-0002

 mkalpinri@gmail.com

© 2025 Mitalpin Negocios Inmobiliarios. Todos los derechos reservados.

[Términos](#) [Privacidad](#)

## 2. Detalle de Propiedad






Visualización de galería de imágenes y formulario de contacto.

### Hermoso Apartamento en Playa del Carmen

Av. 10 Norte 245

\$150000 / noche

Disponible para alquiler temporario



Características

Descripción

Servicios

Ubicación

Estadía

65 m<sup>2</sup> construidos

N/A m<sup>2</sup> de terreno

3 Habitaciones

2 Baños

#### ¿Te interesa este alquiler temporario?

Nombre completo

Email

Teléfono

Fecha de ingreso

dd/mm/aaaa

Fecha de salida

dd/mm/aaaa

Cantidad de personas

Mensaje

Me interesa alquilar esta propiedad...

Consultar disponibilidad y precio

## 3. Tasación

Interfaz del formulario para solicitar tasación de inmueble.

Propiedad

Características

Contacto

### Tasa tu propiedad con Nosotros

Información Básica

Dirección de la propiedad:

Ingresa la dirección completa

Barrio:

Ej: Centro

Localidad:

Ej: Rosario

Provincia:

Ej: Santa Fe

Tipo de propiedad:

Casa

Ubicación del inmueble:

Urbano

Continuar

Ventajas de Nuestra Tasación

Resultados Inmediatos

Precisión Garantizada

Asesoría Personalizada

# Administrador

## 1. Inicio

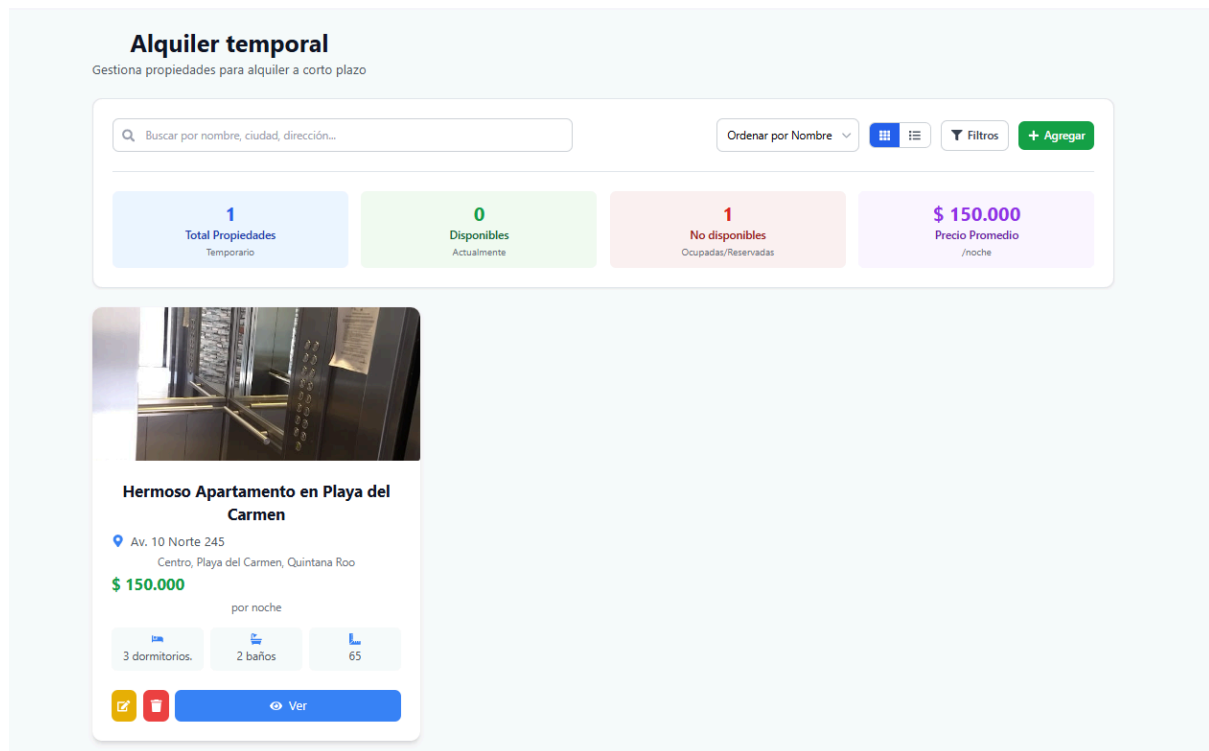
*Tablero de control general con menú lateral de navegación y resumen de actividad reciente del sistema.*



## 2. Propiedades

*Panel de inventario que permite visualizar el estado de los inmuebles y acceder a las funciones de edición y eliminación.*





### 3. Configuración

*Panel para la actualización de datos general de la empresa.*



## Conclusión

**MKalpin** es una solución robusta que integra tecnologías líderes en la industria. La combinación de **Node.js** y **MongoDB** proporciona un backend caracterizado por su alto rendimiento asincrónico y flexibilidad estructural, esencial para gestionar con eficiencia el volumen y la variabilidad de datos propios del mercado inmobiliario. La integración de **Cloudinary** optimiza, transforma y distribuye el contenido multimedia a través de una red de

distribución global (CDN), garantizando una experiencia de usuario fluida, mientras que **SendGrid** (vía Nodemailer) asegura la fiabilidad de todas las comunicaciones transaccionales, manteniendo la imagen corporativa.

Finalmente, la elección de **Render** para el despliegue garantiza que la infraestructura posea la escalabilidad y resiliencia necesarias para absorber el crecimiento sostenido y los picos de demanda del negocio, asegurando la continuidad operativa y una inversión tecnológica a prueba de futuro.