

UNIVERSIDAD DE ALMERIA

ESCUELA SUPERIOR DE INGENIERÍA

# InmoTable

Aplicación de gestión de  
inmuebles para inmobiliarias

Curso: 2024/2025

**Alumno/a:**

Juan Diego Ortega Aranda

**Director/es:**

Francisco José García García  
Antonio Leopoldo Corral Liria

A mis padres, por darme la oportunidad de estudiar, por su apoyo incondicional y por siempre creer en mí. Vuestra confianza fue, sin duda, mi mayor impulso.

A mis compañeros de clase, por ser ese apoyo tan importante durante toda la carrera. Hicisteis que el camino fuera mucho más llevadero y divertido.

A mi tutor, por su guía y paciencia. Gracias a él pude hacer el mejor trabajo posible.

---

# ÍNDICE GENERAL

---

	Página
<b>Resumen y Abstract</b>	<b>xI</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Motivación . . . . .	1
1.2. Objetivos . . . . .	3
1.2.1. Objetivo principal . . . . .	3
1.2.2. Objetivos específicos . . . . .	3
1.3. Planificación Temporal . . . . .	4
1.4. Resumen del documento . . . . .	8
<b>2. Análisis de viabilidad</b>	<b>10</b>
2.1. Situación actual . . . . .	10
2.2. Ingeniería de Requisitos . . . . .	11
2.2.1. Actores . . . . .	11
2.2.2. Requisitos Funcionales . . . . .	12
2.2.2.1. Requisitos Funcionales de la Aplicación Web . . . . .	12
2.2.2.2. Requisitos Funcionales del Sistema de Gestión Interna (Airtable) . . . . .	14
2.2.3. Requisitos No Funcionales . . . . .	18
2.3. Casos de Uso . . . . .	23
2.3.1. Aplicación web . . . . .	23
2.3.2. Airtable . . . . .	24
2.4. Análisis de costes . . . . .	25
<b>3. Modelos de desarrollo y tecnologías</b>	<b>29</b>
3.1. Metodologías de desarrollo . . . . .	29
3.1.1. Comparativa con Otras Metodologías Comunes . . . . .	29
3.1.1.1. Metodologías Tradicionales . . . . .	29
3.1.1.2. Metodologías Ágiles . . . . .	31
3.1.1.3. DevOps . . . . .	33
3.1.2. Justificación y Razonamiento de la Elección Final (Scrum) . . . . .	34

3.2. Arquitecturas de Software . . . . .	35
3.2.1. Justificación de la Arquitectura Seleccionada (Híbrida de Tres Capas) . . . . .	36
3.2.2. Comparativa con Otras Arquitecturas Comunes . . . . .	37
3.2.3. Razonamiento de la Elección Final . . . . .	38
3.3. Tecnologías de Desarrollo . . . . .	38
3.3.1. Tecnologías Frontend . . . . .	38
3.3.2. Tecnologías Backend . . . . .	40
3.3.3. Base de Datos y Gestión de Contenido . . . . .	42
3.3.4. Conclusión sobre la Elección Tecnológica . . . . .	46
3.4. Herramientas de Desarrollo . . . . .	46
3.5. Despliegue . . . . .	49
3.5.1. Estrategia de Despliegue Componentizada . . . . .	49
3.5.2. Despliegue del Frontend (Aplicación Web Angular) en Vercel . . . . .	50
3.5.3. Despliegue del Backend (Laravel API) en Render . . . . .	51
3.5.4. Interacción con Airtable en Producción . . . . .	51
3.6. Conclusiones . . . . .	52
<b>4. InmoTable</b>	<b>54</b>
4.1. Descripción de la aplicación . . . . .	54
4.2. Diseño de la arquitectura . . . . .	54
4.3. Modelo de Datos . . . . .	57
4.4. Diseño e implementación . . . . .	59
4.4.1. Prototipado de la interfaz . . . . .	59
4.4.1.1. Mockup de la Página de Inicio (Landing Page) . . . . .	60
4.4.1.2. Mockup de la Página de Listado de Propiedades . . . . .	61
4.4.1.3. Mockup de la Página de Detalle de Propiedad . . . . .	62
4.4.1.4. Mockup de Formulario de Solicitud de Cita . . . . .	63
4.4.1.5. Mockup de Formulario de Contacto . . . . .	64
4.4.1.6. Justificación de las Desviaciones entre Prototipo e Implementación Real	64
4.4.2. Diseño de la interfaz . . . . .	65
4.4.2.1. Módulo: Componentes Globales de la Interfaz del Cliente . . . . .	65
4.4.2.2. Módulo: Gestión de Propiedades . . . . .	71
4.4.2.3. Módulo: Gestión de Clientes y Preferencias . . . . .	80
4.4.2.4. Módulo: Gestión de Citas . . . . .	90
4.4.2.5. Módulo: Análisis de Negocio . . . . .	94
4.4.2.6. Módulo: Automatizaciones (Visión General) . . . . .	104
4.4.2.7. Módulo: Gestión de Cuentas de Usuario y Roles de Acceso (Airtable) .	109
4.4.3. Flujo de la aplicación . . . . .	110
4.4.3.1. Flujo de cargar el Formulario de Cita . . . . .	110

## ÍNDICE GENERAL

---

4.4.3.2. Flujo de Actualización de Datos de Usuario (Cliente) . . . . .	111
4.4.3.3. Flujo de Gestión de Propiedades Favoritas (Cliente) . . . . .	113
4.4.4. Implementación . . . . .	115
4.4.4.1. Estructura General del Proyecto . . . . .	115
4.4.4.2. Implementación del Frontend (Aplicación Web Angular) . . . . .	116
4.4.4.3. Implementación del Backend (Laravel API) . . . . .	120
4.4.4.4. Patrones de Integración con Airtable . . . . .	122
4.4.4.5. Entorno de Desarrollo y Despliegue (Docker) . . . . .	123
<b>5. Conclusiones y Trabajos Futuros</b>	<b>124</b>
5.1. Conclusiones . . . . .	124
5.2. Conocimientos Académicos Adquiridos . . . . .	124
5.3. Trabajos Futuros . . . . .	125
<b>BIBLIOGRAFÍA</b> . . . . .	<b>125</b>

---

## ÍNDICE DE FIGURAS

---

1.1. Cronograma del TFG . . . . .	7
2.1. Casos de uso de la aplicación web . . . . .	24
2.2. Casos de uso de AirTable . . . . .	24
3.1. Diagrama de la Metodología en Cascada (Waterfall Model) . . . . .	29
3.2. Diagrama del Modelo en V (V-Model) . . . . .	30
3.3. Diagrama del Modelo en Espiral (Spiral Model) . . . . .	31
3.4. Diagrama de la Metodología Scrum . . . . .	31
3.5. Diagrama de la Metodología Kanban . . . . .	32
3.6. Diagrama de la Metodología Extreme Programming (XP) . . . . .	33
3.7. Diagrama del Esquema tareas DevOps . . . . .	33
3.8. Logos de Angular, React y Vue.js . . . . .	39
3.9. Logos de Laravel, .NET, Spring y Node.js . . . . .	40
3.10. Logo de AirTable . . . . .	42
3.11. Logos de PostgreSQL, MySQL y phpMyAdmin . . . . .	43
3.12. Logo de Google Sheets . . . . .	44
3.13. Logos de Notion, Smartsheet, Firebase y Superbase . . . . .	44
3.14. Logos de Draw.io PlantUML y Online Gantt Chart . . . . .	46
3.15. Logo de Figma . . . . .	47
3.16. Logos de Herramientas de Desarrollo del Software . . . . .	48
3.17. Logos de LaTeX y Overleaf . . . . .	49
3.18. Resumen de Modelos de Desarrollo y Tecnologías . . . . .	53
4.1. Estructura General del Proyecto . . . . .	55
4.2. Diagrama del modelo de datos . . . . .	58
4.3. Mockup de la Página de Inicio (Landing Page) . . . . .	60
4.4. Mockup de la Página de Listado de Propiedades . . . . .	61
4.5. Mockup de la Página de Detalle de Propiedad . . . . .	62

4.6. Mockup del Mapa de la Página de Detalle de Propiedad . . . . .	63
4.7. Mockup de Formulario de Solicitud de Cita . . . . .	63
4.8. Mockup de Formulario de Contacto . . . . .	64
4.9. Página de Inicio . . . . .	66
4.10. Página de Inicio dinámica . . . . .	68
4.11. Barra de navegación . . . . .	69
4.12. Barra de navegación dinámica . . . . .	69
4.13. Pie de página . . . . .	69
4.14. Pie de página dinámica . . . . .	70
4.15. Vista de Cuadrícula (Grid View) de Empresa en AirTable . . . . .	70
4.16. Vista de Cuadrícula (Grid View) de Empresa en AirTable . . . . .	70
4.17. Listado de propiedades en formato Grid (por defecto) . . . . .	72
4.18. Listado de propiedades en formato List . . . . .	73
4.19. Filtros del listado de propiedades . . . . .	74
4.20. Filtros del listado de propiedades . . . . .	75
4.21. Vista de Cuadrícula (Grid View) de Propiedades en AirTable . . . . .	77
4.22. Vista de Listado Compacto de Propiedades en AirTable . . . . .	77
4.23. Añadir Propiedad en Vista de Listado Compacto de Propiedades en AirTable . . . . .	78
4.24. Vista Kanban por Estado de Disponibilidad de Propiedades en AirTable . . . . .	78
4.25. Vista Kanban por Tipo de Propiedades en AirTable . . . . .	79
4.26. Vista de Galería Visual de Propiedades en AirTable . . . . .	79
4.27. Inicio de sesión en Aplicación Web . . . . .	80
4.28. Registro en Aplicación Web . . . . .	81
4.29. Menú Desplegable (Dropdown) de Usuario Logueado en Aplicación Web . . . . .	82
4.30. Mi Perfil en Aplicación Web . . . . .	82
4.31. Contacto en Aplicación Web . . . . .	83
4.32. Botón Agregar Propiedad Interés (Favoritos) en Aplicación Web . . . . .	84
4.33. Mensaje Agregar Propiedad Interés (Favoritos) en Aplicación Web . . . . .	84
4.34. Botón Quitar Propiedad Interés (Favoritos) en Aplicación Web . . . . .	84
4.35. Mensaje Quitar Propiedad Interés (Favoritos) en Aplicación Web . . . . .	84
4.36. Mis Propiedad Interés (Favoritos) en formato Grid en Aplicación Web . . . . .	85
4.37. Mis Propiedad Interés (Favoritos) en formato List en Aplicación Web . . . . .	86
4.38. Vista de Cuadrícula (Grid View) de Clientes en AirTable . . . . .	87
4.39. Vista Kanban de Clientes en AirTable . . . . .	87
4.40. Vista de Cuadrícula (Grid View) de Agentes en AirTable . . . . .	88

---

4.41. Vista de Cuadrícula (Grid View) de Usuarios en AirTable . . . . .	89
4.42. Vista de Cuadrícula (Grid View) de Clientes en AirTable . . . . .	89
4.43. Solicitud de Cita en Aplicación Web . . . . .	90
4.44. Vista de Cuadrícula (Grid View) de Citas en AirTable . . . . .	91
4.45. Calendario de Citas en AirTable . . . . .	92
4.46. Vista de Calendario de Citas en AirTable . . . . .	92
4.47. Vista de Cuadrícula (Grid View) de Citas en AirTable . . . . .	93
4.48. Vista de Calendario de Citas en AirTable . . . . .	94
4.49. Interfaz de Análisis de Stock en AirTable . . . . .	95
4.50. Detalle de Interfaz de Análisis de Stock en AirTable . . . . .	95
4.51. Interfaz de Media de Precios por Tipo de Inmueble en AirTable . . . . .	96
4.52. Detalle de Interfaz de Media de Precios por Tipo de Inmueble en AirTable . . . . .	96
4.53. Filtro de Interfaz de Media de Precios por Tipo de Inmueble en AirTable . . . . .	97
4.54. Interfaz de Visitas por Propiedad en AirTable . . . . .	97
4.55. Detalle de Interfaz de Visitas por Propiedad en AirTable . . . . .	98
4.56. Interfaz de Visitas Programadas en AirTable . . . . .	98
4.57. Detalle por semana de Interfaz de Visitas Programadas en AirTable . . . . .	99
4.58. Detalle por mes de Interfaz de Visitas Programadas en AirTable . . . . .	99
4.59. Interfaz de Tiempo Medio de Venta en AirTable . . . . .	100
4.60. Detalle de Interfaz de Tiempo Medio de Venta en AirTable . . . . .	100
4.61. Detalle de Interfaz de Tiempo Medio de Venta en AirTable . . . . .	101
4.62. Interfaz de Propiedades con Baja Rotación en AirTable . . . . .	101
4.63. Propiedades Disponibles con Baja Rotación en AirTable . . . . .	102
4.64. Interfaz de Propiedades en AirTable . . . . .	102
4.65. Detalle de Interfaz de Propiedades Disponibles en AirTable . . . . .	103
4.66. Interfaz de Tipo de Propiedades en AirTable . . . . .	103
4.67. Detalle de Interfaz de Tipo de Propiedades en AirTable . . . . .	104
4.68. Automatización Notificar Agente Propiedad Vendida en AirTable . . . . .	105
4.69. Email de Automatización Notificar Agente Propiedad Vendida en AirTable . . . . .	105
4.70. Automatización Notificar Agente Cambio Precio Propiedad en AirTable . . . . .	106
4.71. Email de Automatización Notificar Agente Cambio Precio Propiedad en AirTable . . . . .	106
4.72. Automatización Asignar Agente al Crear Nuevo Cliente en AirTable . . . . .	107
4.73. Automatización Asigna nuevo Cliente al Crear nuevo Usuario en AirTable . . . . .	108
4.74. Email de Automatización Asigna nuevo Cliente al Crear nuevo Usuario en AirTable . . . . .	109
4.75. Gestión de Cuentas de Usuario y Roles de Acceso en Airtable . . . . .	109

---

## ÍNDICE DE FIGURAS

---

4.76. Diagrama Secuencias - Formulario Pedir Cita AirTable . . . . .	111
4.77. Diagrama Secuencias - Formulario Editar Perfil Usuario . . . . .	113
4.78. Diagrama Secuencias - Formulario Pedir Cita AirTable . . . . .	114
4.79. Estructura General del Proyecto . . . . .	115
4.80. Estructura del Frontend . . . . .	116
4.81. Estructura de los módulos lógicos del Frontend . . . . .	117
4.82. Estructura de Componentes del Frontend . . . . .	117
4.83. Estructura de Servicios del Frontend . . . . .	118
4.84. Estructura de Rutas del Frontend . . . . .	118
4.85. Estructura de Rutas Auth del Frontend . . . . .	119
4.86. Estructura de Guards del Frontend . . . . .	119
4.87. Estructura del Backend . . . . .	120
4.88. Estructura de los Controladores del Backend . . . . .	121
4.89. Estructura de los Servicios del Backend . . . . .	121
4.90. Estructura de las Rutas del Backend . . . . .	121
4.91. Estructura del Middleware del Backend . . . . .	122

---

## ÍNDICE DE TABLAS

---

2.1. Actor ACT-01: Sin Usuario (Web App) . . . . .	11
2.2. Actor ACT-02: Cliente (Web App) . . . . .	11
2.3. Actor ACT-03: Agente (AirTable) . . . . .	12
2.4. Actor ACT-04: Administrador (AirTable) . . . . .	12
2.5. RF 01.1: Buscar Propiedades por Criterios . . . . .	12
2.6. RF 01.2: Visualizar Listado de Propiedades . . . . .	12
2.7. RF 01.3: Visualizar Detalle de Propiedad . . . . .	13
2.8. RF 01.4: Registrarse como Cliente . . . . .	13
2.9. RF 01.5: Iniciar Sesión (Cliente) . . . . .	13
2.10. RF 01.6: Enviar Mensaje de Contacto . . . . .	13
2.11. RF 01.7: Solicitar Cita para Propiedad . . . . .	13
2.12. RF 02.1: Crear Nueva Propiedad . . . . .	14
2.13. RF 02.2: Consultar Propiedades (en Airtable) . . . . .	14
2.14. RF 02.3: Actualizar Información de Propiedad . . . . .	14
2.15. RF 02.4: Eliminar Propiedad . . . . .	14
2.16. RF 03.1: Programar y Modificar Citas . . . . .	15
2.17. RF 03.2: Visualizar Calendario de Citas . . . . .	15
2.18. RF 03.3: Actualizar Estado de Cita . . . . .	15
2.19. RF 04.1: Registrar Nuevo Cliente . . . . .	15
2.20. RF 04.2: Consultar y Actualizar Cliente . . . . .	16
2.21. RF 05.1: Visualizar Análisis de Stock . . . . .	16
2.22. RF 05.2: Visualizar Media de Precios por Tipo de Inmueble . . . . .	16
2.23. RF 05.3: Visualizar Visitas por Propiedad . . . . .	16
2.24. RF 05.4: Visualizar Tiempo Medio de Venta/Rotación . . . . .	17
2.25. RF 05.5: Visualizar Propiedades con Baja Rotación . . . . .	17
2.26. RF 05.6: Visualizar Distribución por Tipo de Propiedad . . . . .	17
2.27. RF 06.1: Administrar Usuarios Internos y Roles . . . . .	17
2.28. RF 06.2: Configurar y Supervisar Automatizaciones . . . . .	18

## ÍNDICE DE TABLAS

---

2.29.RF 06.3: Personalizar Vistas y Formularios (Airtable) . . . . .	18
2.30.Requisito RNF-01.1 . . . . .	18
2.31.Requisito RNF-01.2 . . . . .	18
2.32.Requisito RNF-01.3 . . . . .	19
2.33.Requisito RNF-02.1 . . . . .	19
2.34.Requisito RNF-02.2 . . . . .	19
2.35.Requisito RNF-02.3 . . . . .	19
2.36.Requisito RNF-03.1 . . . . .	19
2.37.Requisito RNF-03.2 . . . . .	20
2.38.Requisito RNF-04.1 . . . . .	20
2.39.Requisito RNF-04.2 . . . . .	20
2.40.Requisito RNF-04.3 . . . . .	20
2.41.Requisito RNF-05.1 . . . . .	20
2.42.Requisito RNF-05.2 . . . . .	21
2.43.Requisito RNF-06.1 . . . . .	21
2.44.Requisito RNF-06.2 . . . . .	21
2.45.Requisito RNF-07.1 . . . . .	21
2.46.Requisito RNF-07.2 . . . . .	21
2.47.Requisito RNF-08.1 . . . . .	22
2.48.Requisito RNF-08.2 . . . . .	22
2.49.Requisito RNF-09.1 . . . . .	22
2.50.Requisito RNF-09.2 . . . . .	22
2.51.Requisito RNF-10.1 . . . . .	22
2.52.Requisito RNF-10.2 . . . . .	23
2.53.Comparativa de Costes y Enfoque de Airtable con Competidores Clave . . . . .	26

---

## RESUMEN Y ABSTRACT

---

El presente Trabajo de Fin de Grado describe el desarrollo de **InmoTable**, una innovadora aplicación web para la gestión inmobiliaria. La solución implementa una arquitectura híbrida que combina un **frontend en Angular** para una óptima experiencia del cliente y un **backend en Laravel** para la lógica de negocio y la integración API. Su principal diferenciador es el uso de **Airtable como base de datos central y potente plataforma low-code** para la gestión interna de agentes y administradores, incluyendo propiedades, clientes, citas y análisis de negocio. Este enfoque permite una alta eficiencia, un bajo coste operativo y una gran portabilidad, haciendo de InmoTable una solución replicable y ágil para diversas inmobiliarias.

This Final Degree Project describes the development of **InmoTable**, an innovative web application for real estate management. The solution implements a hybrid architecture combining an **Angular frontend** for an optimal client experience and a **Laravel backend** for business logic and API integration. Its main differentiator is the use of **Airtable as a central database and powerful low-code platform** for internal management by agents and administrators, including properties, clients, appointments, and business analytics. This approach enables high efficiency, low operational cost, and great portability, making InmoTable a replicable and agile solution for various real estate agencies.

# 1 INTRODUCCIÓN

En este capítulo se presenta la motivación del proyecto, sus objetivos principales y la planificación general

## 1.1 MOTIVACIÓN

El presente Trabajo de Fin de Grado se enmarca en la creciente necesidad del sector inmobiliario de optimizar y modernizar sus procesos de gestión de propiedades y clientes. En un mercado cada vez más dinámico y competitivo, las herramientas tradicionales a menudo presentan limitaciones en términos de eficiencia, colaboración y acceso a la información en tiempo real.

La motivación principal de este proyecto surge de la identificación de estas deficiencias y de la oportunidad de aplicar tecnologías web modernas para crear una solución que impulse la productividad de los agentes inmobiliarios y mejore la experiencia del cliente. Específicamente, este TFG busca abordar los siguientes puntos clave:

- Necesidad de Flexibilidad y Adaptabilidad en la Gestión de Datos:** Las bases de datos relacionales tradicionales, aunque robustas, a menudo imponen esquemas rígidos que dificultan la adaptación a los cambios constantes en las necesidades del negocio inmobiliario, como la inclusión de nuevos tipos de propiedades, atributos o métricas. Se observa una creciente tendencia hacia herramientas más ágiles en la gestión de datos.
- Demandas de Colaboración y Acceso en Tiempo Real:** En el sector inmobiliario, la colaboración entre agentes, la actualización instantánea del estado de las propiedades y la programación de citas son vitales. Las soluciones existentes no siempre facilitan un entorno de trabajo colaborativo y accesible desde cualquier lugar.
- Optimización de Procesos y Toma de Decisiones:** Los agentes inmobiliarios requieren de información clara y concisa para tomar decisiones estratégicas, como la valoración de propiedades o la identificación de inmuebles con baja rotación. La recopilación manual de datos o el uso de herramientas dispersas ralentiza este proceso.

Para dar respuesta a estas motivaciones, se propone el desarrollo de la aplicación web **InmoTable**. Su justificación se basa en una serie de ventajas estratégicas y operativas:

- Eficiencia en el Desarrollo y Ahorro de Costes (Modelo No-Code/Low-Code):** La utilización de una base de datos tipo spreadsheet como Airtable es una decisión fun-

damental que aporta una eficiencia significativa. Permite externalizar gran parte de la lógica de gestión de la empresa, reduciendo drásticamente la necesidad de programar módulos complejos de administración de datos en el backend y frontend. Esto se traduce en un ahorro considerable de tiempo y recursos de desarrollo y mantenimiento. La opción de un plan básico gratuito de Airtable y su escalabilidad a bajo coste en el futuro la convierten en una solución muy rentable desde el punto de vista económico. Además, su interfaz es altamente intuitiva y fácil de usar para los agentes inmobiliarios.

- **Flexibilidad y Adaptabilidad del Esquema de Datos:** A diferencia de las bases de datos tradicionales que requieren un esquema fijo, Airtable permite diseñar tablas y campos de forma dinámica, adaptándose rápidamente a las necesidades cambiantes del sector inmobiliario. Esta versatilidad es crucial cuando se manejan diferentes tipos de datos y se busca evitar las limitaciones de una estructura predefinida.
- **Colaboración en Tiempo Real y Acceso Remoto:** Airtable facilita que múltiples usuarios trabajen simultáneamente en los datos, con cambios instantáneos y accesibilidad remota, lo cual es esencial para equipos de agentes distribuidos o en constante movimiento.
- **Potencial de Automatización de Flujos de Trabajo:** La capacidad de crear flujos de trabajo automáticos directamente en Airtable (como el envío de notificaciones por email al agente al recibir un contacto o al cliente tras confirmar una cita) ahorra tiempo en tareas repetitivas y mejora la comunicación.
- **Modelo de Negocio Replicable (Franquiable):** Una ventaja estratégica clave de esta arquitectura es su facilidad de replicación y adaptación para múltiples inmobiliarias. Una vez desarrollada la base de Airtable para una empresa, su implementación en una nueva inmobiliaria se simplifica enormemente: solo requeriría copiar la base de Airtable existente a un nuevo proyecto y actualizar las claves de API y nombres de configuración en el archivo environment de la aplicación web (Angular y Laravel). Esto transforma el proyecto en una solución "plug-and-play" para futuras implementaciones, lo que representa una propuesta de valor significativa para el mercado.
- **Experiencia de Usuario Diferenciada para el Cliente:** La implementación de un frontend en Angular ofrece una interfaz moderna, dinámica y atractiva para los clientes finales. Esto les permitirá buscar propiedades de manera eficiente, visualizar detalles completos y gestionar sus interacciones (como la programación de citas) de forma intuitiva, mejorando su experiencia general con la inmobiliaria.
- **Backend Robusto y Seguro con Laravel:** La capa de Laravel no solo facilita la comunicación entre Angular y Airtable, sino que también garantiza la seguridad y la fiabilidad del sistema mediante un control de autenticación y autorización (Laravel Passport), protegiendo la integridad de los datos y las operaciones.
- **Herramientas Analíticas Integradas para Agentes Inmobiliarios:** Mediante las interfaces y automatizaciones configuradas directamente en Airtable (Análisis de stock, Media de precios por tipo de inmueble, Visitas por propiedad, Tiempo medio de venta, Propiedades con baja rotación, entre otras), el proyecto empodera a los agentes con

## 1.2. OBJETIVOS

información valiosa para una toma de decisiones informada, mejorando la eficiencia operativa y la estrategia de ventas.

En síntesis, **InmoTable** no solo busca modernizar la gestión inmobiliaria a través de la digitalización, sino que también establece un modelo de trabajo dual innovador donde la flexibilidad y las capacidades de gestión de un sistema *no-code/low-code* se combinan con la robustez y personalización de un desarrollo web a medida, creando una sinergia que beneficia tanto a los clientes como a los profesionales del sector.

### 1.2 OBJETIVOS

El objetivo principal de este Trabajo de Fin de Grado es desarrollar una aplicación web de gestión inmobiliaria robusta, eficiente y escalable, que satisfaga las necesidades tanto de los clientes finales como de los agentes inmobiliarios, utilizando un enfoque híbrido que combine un desarrollo web a medida con las capacidades de una base de datos tipo spreadsheet.

Para alcanzar este objetivo general, se establecen los siguientes objetivos específicos:

#### 1.2.1 Objetivo principal

Desarrollar una aplicación web integral para la gestión inmobiliaria, **InmoTable**, que proporcione una plataforma dual: una interfaz de usuario optimizada para clientes finales y un sistema de gestión y análisis de negocio eficiente para agentes inmobiliarios, logrando un equilibrio entre personalización y las ventajas de las herramientas *no-code/low-code* [1].

#### 1.2.2 Objetivos específicos

##### 1. Diseñar y Desarrollar un Frontend Intuitivo para el Cliente:

- Crear una interfaz de usuario atractiva y fácil de usar con Angular que permita a los clientes buscar, filtrar y visualizar propiedades de manera eficiente (listado y detalle).
- Implementar funcionalidades de autenticación y registro de usuarios para clientes, asegurando un acceso seguro y personalizado.
- Permitir a los clientes la interacción con la inmobiliaria mediante funcionalidades como la programación de citas para visitas y un formulario de contacto.

##### 2. Desarrollar un Backend Robusto y Seguro con Laravel:

- Implementar una API RESTful en Laravel que actúe como una capa intermedia entre el frontend de Angular y la API de Airtable.
- Asegurar la comunicación y el acceso a los datos mediante la implementación de un sistema de autenticación y autorización basado en tokens con Laravel Passport.
- Gestionar las operaciones CRUD sobre las entidades (propiedades, clientes, agentes, citas) de forma segura y eficiente, traduciendo las solicitudes de la aplicación a interacciones con la API de Airtable.

### 1.3. PLANIFICACIÓN TEMPORAL

#### 3. Implementar y Configurar Airtable como Plataforma de Gestión Central para Agentes:

- Definir y estructurar las tablas de Airtable (Propiedades, Agentes, Clientes, Citas, Empresas, Usuarios, Contacto) para almacenar toda la información inmobiliaria de manera organizada y relacional.
- Configurar y personalizar las vistas de Airtable (ej. calendario de citas, galerías de propiedades, vistas de mapa) para facilitar a los agentes inmobiliarios la visualización y el seguimiento de los datos.
- Crear formularios de entrada de datos en Airtable para la fácil adición y modificación de registros por parte de los agentes (ej. Nueva Propiedad, Nueva Cita, Nuevo Cliente).
- Configurar automatizaciones en Airtable para gestionar flujos de trabajo clave, como el envío de notificaciones por correo electrónico a agentes y clientes, y la categorización automática de propiedades.

#### 4. Proveer Herramientas Analíticas y de Inteligencia de Negocio para Agentes (a través de Interfaces de Airtable): Desarrollar dashboards interactivos en Airtable para ofrecer a los agentes una visión clara del rendimiento y estado del negocio. Esto incluye:

- Análisis de stock (métricas generales, propiedades disponibles, precio medio).
- Media de precios por tipo de inmueble (análisis de tendencias de mercado).
- Visitas por propiedad (seguimiento del interés en cada inmueble).
- Visitas programadas (gestión del calendario de citas).
- Tiempo medio de venta (análisis de eficiencia de rotación de propiedades).
- Propiedades con baja rotación (identificación de inmuebles que requieren atención).
- Propiedades (General) y Tipo de Propiedades (visión completa y categorizada del inventario).

#### 5. Demostrar la Viabilidad y Replicabilidad del Modelo de Negocio Híbrido:

- Validar la eficiencia en el desarrollo y el ahorro de costes al aprovechar las funcionalidades *no-code/low-code* de Airtable para la gestión interna de la empresa.
- Establecer la metodología de implementación del proyecto en nuevas inmobiliarias [2], demostrando la facilidad de replicación de la base de Airtable y la adaptación de la aplicación web mediante la configuración de variables de entorno.

### 1.3 PLANIFICACIÓN TEMPORAL

La planificación temporal de este Trabajo de Fin de Grado se estableció con una duración estimada inicial de 300 horas, distribuidas a lo largo de un curso académico. Esta estimación se basó en el alcance inicial del proyecto y en la complejidad de las tecnologías a integrar.

Sin embargo, la implementación real del proyecto, en un ejercicio de transparencia y precisión, demandó un total de 395 horas, una cifra que, aunque supera la estimación inicial, se mantiene dentro de un rango justificado por la naturaleza innovadora y la profundización en las funcionalidades.

Esta sección detalla las fases clave del proyecto, la asignación de tiempo estimado y el tiempo real dedicado a cada una de sus sub-fases, proporcionando una base sólida para la creación de un cronograma visual detallado.

**Fases del Proyecto y Detalle Horario:**

**1. Fase 1: Investigación y Diseño**

■ **Horas Estimadas: 45 horas**

■ **Horas Reales Invertidas: 55 horas**

- a) **1.1. Definición de requisitos y alcance:** 15 horas estimadas / 20 horas reales. Esta sub-fase requirió una dedicación extra para una comprensión exhaustiva de las necesidades específicas de la gestión inmobiliaria para clientes y agentes, y para delinear cómo el enfoque híbrido (Angular/Laravel/Airtable) optimizaría estos procesos.
- b) **1.2. Análisis y selección de tecnologías principales:** 15 horas estimadas / 15 horas reales. La evaluación y justificación de la arquitectura se ajustó a lo previsto.
- c) **1.3. Diseño inicial de la estructura de datos en Airtable (tablas y relaciones):** 15 horas estimadas / 20 horas reales. La conceptualización y diseño inicial de las tablas de Airtable (Agentes, Citas, Clientes, Empresas, Propiedades, Usuarios, Contacto), considerando las relaciones y tipos de campos óptimos para una base de datos spreadsheet, demandó un poco más de tiempo.

**2. Fase 2: Desarrollo del Backend y Conexión con Airtable**

■ **Horas Estimadas: 65 horas**

■ **Horas Reales Invertidas: 70 horas**

- a) **2.1. Implementación de la API RESTful en Laravel:** 20 horas estimadas / 25 horas reales. La creación de controladores y rutas que sirvieran como capa de abstracción para Airtable, incluyendo la validación y transformación de datos, requirió un esfuerzo adicional.
- b) **2.2. Desarrollo del AirtableService para interacción (CRUD):** 25 horas estimadas / 25 horas reales. La implementación de la lógica para interactuar con la API externa de Airtable, optimizando las llamadas y el manejo de respuestas, se gestionó dentro de lo previsto.
- c) **2.3. Configuración de autenticación (Laravel Passport):** 10 horas estimadas / 10 horas reales. La implementación del sistema de autenticación seguro basado en OAuth2 se mantuvo en línea con la estimación.

- d) **2.4. Configuración inicial de formularios y automatizaciones en Airtable (simples):** 10 horas estimadas / 10 horas reales. Creación de los primeros formularios (ej. Nueva Cita, Nueva Propiedad) y automatizaciones básicas para validar el flujo de datos.

### 3. Fase 3: Desarrollo del Frontend (Aplicación Cliente)

- **Horas Estimadas:** 70 horas
- **Horas Reales Invertidas:** 80 horas
  - a) **3.1. Implementación de componentes de interfaz de usuario (UI/UX) con Angular:** 45 horas estimadas / 55 horas reales. El desarrollo de interfaces para el cliente (listado y detalle de propiedades, formularios de citas y contacto, autenticación), con un fuerte enfoque en una experiencia de usuario fluida y un diseño receptivo, lo cual siempre consume más tiempo para asegurar la calidad.
  - b) **3.2. Integración con servicios del backend de Laravel:** 20 horas estimadas / 20 horas reales. Conectar los componentes de Angular con los endpoints de la API de Laravel, manejando la comunicación asíncrona y la visualización correcta de los datos provenientes de Airtable.
  - c) **3.3. Implementación de guardias de ruta y lógica de roles:** 5 horas estimadas / 5 horas reales. Asegurar que los diferentes perfiles de usuario accedan solo a las secciones correspondientes.

### 4. Fase 4: Configuración y Personalización de Airtable (Herramienta para Agentes)

- **Horas Estimadas:** 30 horas
- **Horas Reales Invertidas:** 70 horas
  - a) **4.1. Configuración avanzada de vistas de datos:** 10 horas estimadas / 20 horas reales. Diseño y optimización de las múltiples vistas de Airtable (calendario de citas, galerías, mapas, etc.) para ofrecer a los agentes herramientas de visualización de datos internas de alta calidad, lo cual fue un componente más extenso de lo previsto.
  - b) **4.2. Desarrollo de interfaces analíticas (Dashboards) en Airtable:** 10 horas estimadas / 25 horas reales. La creación de los dashboards de análisis de negocio (Análisis de stock, Media de precios, Visitas por propiedad, Tiempo medio de venta, Propiedades con baja rotación, etc.) es un diferenciador clave que implicó una curva de aprendizaje y configuración más prolongada.
  - c) **4.3. Optimización de formularios y automatizaciones de Airtable (avanzadas):** 10 horas estimadas / 25 horas reales. La creación y el ajuste fino de flujos de trabajo automatizados, incluyendo envíos de correos electrónicos condicionales y la gestión del ciclo de vida de las propiedades, representó un trabajo significativo en la plataforma Airtable.

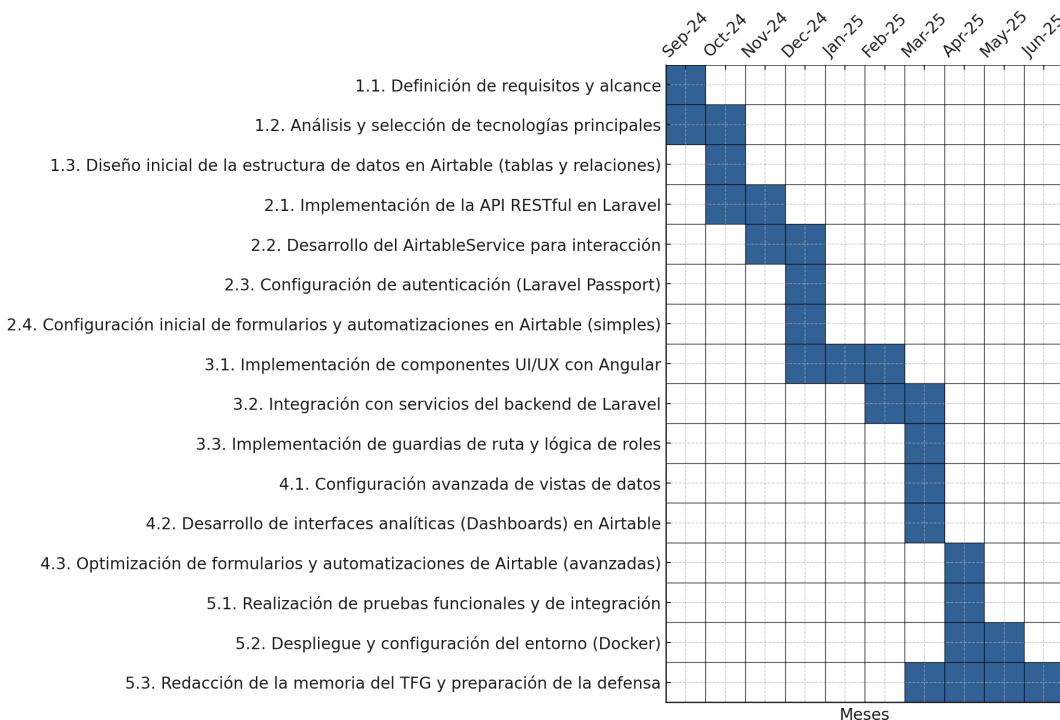
### 5. Fase 5: Pruebas, Despliegue, Ajustes y Documentación

- **Horas Estimadas:** 75 horas
-

■ **Horas Reales Invertidas: 110 horas**

- a) **5.1. Realización de pruebas funcionales y de integración:** 15 horas estimadas / 20 horas reales. Verificación de la correcta funcionalidad e integración entre el frontend de Angular, el backend de Laravel y la API de Airtable, así como la gestión de la autenticación y la lógica de negocio, consumió más tiempo de depuración.
- b) **5.2. Despliegue y configuración del entorno (Docker):** 10 horas estimadas / 10 horas reales. Optimización de los contenedores Docker para producción y configuración de variables de entorno para la replicabilidad.
- c) **5.3. Redacción de la memoria del TFG y preparación de la defensa:** 50 horas estimadas / 80 horas reales. Esta es una de las fases más intensivas en cualquier TFG, implicando la estructuración, redacción detallada de la metodología, resultados, conclusiones, discusión de tecnologías, y la preparación exhaustiva de la presentación y su defensa, abarcando un rango significativo de horas en la realidad de este tipo de proyectos académicos. Se ha ajustado para reflejar este considerable esfuerzo intelectual y de síntesis.

**Tabla Resumen de Horas (para Cronograma Visual):**



**Figura 1.1: Cronograma del TFG**

**Justificación de la Desviación Temporal Global:**

La inversión total de 395 horas (frente a las 300 estimadas inicialmente) se justifica por la ambición y el nivel de detalle alcanzado en el proyecto, especialmente en áreas donde

#### 1.4. RESUMEN DEL DOCUMENTO

---

se maximizó el potencial de la plataforma Airtable y en la fase de documentación final. Las principales razones de esta extensión de tiempo son:

- **Profundización en las Capacidades de Airtable como Plataforma de Negocio (Fase 4: +45 horas):** Esta fase experimentó una desviación significativa. La decisión de no solo usar Airtable como una base de datos, sino de explotar a fondo sus funcionalidades *low-code* para crear interfaces analíticas avanzadas (dashboards de stock, precios, visitas, rotación) y automatizaciones complejas (envíos de emails, categorización de propiedades), requirió un tiempo de diseño, configuración y optimización significativamente mayor. Este esfuerzo es clave para la justificación del ahorro de desarrollo personalizado que ofrece el enfoque híbrido.
- **Complejidad de la Integración Híbrida (Fase 1, 2 y 3):** La interconexión fluida entre el frontend de Angular, el backend de Laravel y una API de terceros como Airtable implicó desafíos adicionales en la sincronización de datos, el manejo de errores y la optimización del rendimiento de las solicitudes API. Esto resultó en un ligero aumento acumulado en las fases de diseño y frontend (+10 y +10 horas respectivamente).
- **Redacción de la Memoria y Preparación de la Defensa (Fase 5.3: +65 horas):** La fase final de documentación y preparación de la defensa ha sido ajustada para reflejar un tiempo realista y sustancial (de 15 horas estimadas a 40 horas estimadas y de 40 horas reales a 65 horas reales para el total de la fase 5). La redacción de una memoria completa, que detalla la arquitectura, la implementación, los resultados y las reflexiones sobre la viabilidad del modelo de negocio, es un proceso intensivo que requiere múltiples revisiones y ajustes, justificando ampliamente el tiempo dedicado.

La inversión de estas horas adicionales ha permitido entregar una solución más completa, escalable y con un valor de negocio superior, validando la eficacia del enfoque híbrido propuesto para la gestión inmobiliaria digital.

#### 1.4 RESUMEN DEL DOCUMENTO

El presente Trabajo de Fin de Grado describe el desarrollo de **InmoTable**, una aplicación web innovadora para la gestión inmobiliaria. La memoria se organiza de la siguiente manera:

- **Capítulo 1. Introducción:** Presenta la motivación del proyecto, sus objetivos principales y la planificación general.
- **Capítulo 2. Análisis de viabilidad:** Evalúa la factibilidad técnica y económica de la solución, destacando su bajo coste y alta replicabilidad gracias al uso de Airtable.
- **Capítulo 3. Modelos de desarrollo y tecnologías:** Detalla la metodología Scrum empleada y justifica la elección de Angular, Laravel y Airtable frente a otras opciones.
- **Capítulo 4. InmoTable:** Describe en profundidad el diseño y la implementación del sistema, abarcando su arquitectura, el modelo de datos y las interfaces de usuario para clientes (web) y agentes (Airtable).



#### 1.4. RESUMEN DEL DOCUMENTO

---

- **Capítulo 5. Conclusiones y trabajos futuros:** Recoge los resultados clave del proyecto y propone futuras líneas de evolución.
- **Capítulo 6. Bibliografía:** Contiene todas las referencias documentales utilizadas.

---

## 2 ANÁLISIS DE VIABILIDAD

---

En este capítulo se evalúa la factibilidad técnica y económica de la solución, destacando su bajo coste y alta replicabilidad gracias al uso de Airtable.

### 2.1 SITUACIÓN ACTUAL

La concepción del proyecto **InmoTable** surge de un análisis del mercado actual de soluciones de gestión inmobiliaria, donde se ha identificado una oportunidad significativa para ofrecer una alternativa que combine la eficiencia del desarrollo web con la agilidad de las plataformas *low-code*. La motivación principal para realizar este proyecto se basa en los siguientes pilares de diferenciación:

- **Identificación de un Nicho de Mercado:** Se ha detectado una carencia de soluciones en el mercado que ofrezcan una aplicación web intuitiva y moderna para el cliente final, al mismo tiempo que proporcionan una plataforma de gestión interna (CRM/ERP) excepcionalmente flexible, de bajo coste y fácil de usar para los agentes inmobiliarios, sin la necesidad de un desarrollo a medida complejo para la administración del día a día. Las soluciones existentes suelen ser o bien CRMs robustos y costosos que requieren una alta inversión inicial y personalización, o bien herramientas muy simples que no escalan.
- **Eficiencia en el Desarrollo y Sencillez de Implementación:** El enfoque híbrido adoptado, que delega la mayor parte de la lógica de gestión interna a Airtable, convierte el proceso de desarrollo personalizado del backend y el frontend en algo notablemente más sencillo y rápido de programar. Al no tener que construir desde cero complejos paneles de administración, formularios dinámicos, módulos de análisis de datos o sistemas de automatización de flujos de trabajo (que ya están preconstruidos y son altamente configurables en Airtable), se logra una reducción drástica en el tiempo y el esfuerzo de ingeniería.
- **Minimización de Costes Operativos y de Licencia:** La elección de Airtable es fundamental desde el punto de vista económico. Al disponer de una opción básica gratuita y planes de pago muy asequibles y escalables, se reduce significativamente la barrera de entrada para las inmobiliarias, eliminando la necesidad de adquirir licencias costosas de software de gestión o de invertir en infraestructura de bases de datos tradicional. Esto no solo abarata el coste de adquisición, sino también el de operación y mantenimiento.

- **Alta Portabilidad y Replicabilidad del Modelo de Negocio:** Una de las ventajas más estratégicas de **InmoTable** es su inherente portabilidad. Una vez desarrollada y optimizada la base de Airtable para una inmobiliaria, el modelo puede ser replicado y adaptado a una nueva agencia de forma extremadamente rápida, simplemente clonando la base de Airtable y ajustando unas pocas variables de entorno (claves de API, nombres de base) en la aplicación web (Angular y Laravel). Esto hace que la solución sea altamente "franquiable" y atractiva para el mercado, permitiendo una expansión rápida y de bajo coste.

## 2.2 INGENIERÍA DE REQUISITOS

Durante esta fase se ha realizado un análisis detallado para identificar las necesidades que debe cubrir el sistema. Esta definición de requisitos constituye la base sobre la cual se diseñará y desarrollará la aplicación. Para organizar esta información, se han clasificado los requisitos en tres apartados principales:

- **Actores:** Los actores son todas aquellas entidades (personas o sistemas) que interactúan con el sistema.
- **Requisitos funcionales:** Son las funcionalidades que debe ofrecer el sistema para satisfacer las necesidades de los usuarios dentro del dominio inmobiliario.
- **Requisitos no funcionales:** Son las condiciones que debe cumplir el sistema respecto al entorno, calidad y restricciones de uso.

### 2.2.1 Actores

ACT-01	Sin Usuario (Web App)
Descripción	Usuario anónimo que navega por la aplicación web sin autenticarse, puede ver propiedades públicas y solicitar atención mediante contacto.
Comentarios	No dispondrá de acceso a funcionalidades que requieren autenticación.

Tabla 2.1: Actor ACT-01: Sin Usuario (Web App)

ACT-02	Cliente (Web App)
Descripción	Usuario registrado y autenticado en la aplicación web con acceso a funcionalidades personalizadas como gestión de intereses, solicitar y visualizar citar, y modificar perfil.
Comentarios	Requiere registro y autenticación para acceder a funcionalidades completas.

Tabla 2.2: Actor ACT-02: Cliente (Web App)

ACT-03	Agente (AirTable)
Descripción	Usuario con funciones operacionales, gestiona propiedades, clientes y citas desde AirTable con permisos de edición limitados.
Comentarios	Acceso exclusivo a través de AirTable, sin acceso directo a la aplicación web.

**Tabla 2.3:** Actor ACT-03: Agente (AirTable)

ACT-04	Administrador (AirTable)
Descripción	Usuario con funciones de administrador, control total sobre el sistema incluyendo gestión de usuarios, configuración de empresa y eliminación de datos.
Comentarios	Permisos completos en AirTable para gestión y configuración del sistema.

**Tabla 2.4:** Actor ACT-04: Administrador (AirTable)

## 2.2.2 Requisitos Funcionales

### 2.2.2.1 Requisitos Funcionales de la Aplicación Web

RF 01.1	Buscar Propiedades por Criterios
<b>Actores</b>	ACT-01, ACT-02
<b>Descripción</b>	Permite al usuario (visitante o cliente registrado) buscar propiedades utilizando filtros como precio, tipo de propiedad, número de habitaciones, etc., para refinar los resultados de su interés.
<b>Precondiciones</b>	Acceso a la aplicación web.

**Tabla 2.5:** RF 01.1: Buscar Propiedades por Criterios

RF 01.2	Visualizar Listado de Propiedades
<b>Actores</b>	ACT-01, ACT-02
<b>Descripción</b>	Muestra una lista paginada de propiedades disponibles, presentando información básica como la imagen principal, la referencia y el precio.
<b>Precondiciones</b>	Acceso a la aplicación web.

**Tabla 2.6:** RF 01.2: Visualizar Listado de Propiedades

RF 01.3	Visualizar Detalle de Propiedad
Actores	ACT-01, ACT-02
Descripción	Permite acceder a una página dedicada con toda la información detallada de una propiedad, incluyendo fotos adicionales, descripción completa y características específicas.
Precondiciones	Acceso a la aplicación web; Propiedad seleccionada del listado.

Tabla 2.7: RF 01.3: Visualizar Detalle de Propiedad

RF 01.4	Registrarse como Cliente
Actores	ACT-01, ACT-02
Descripción	Permite a nuevos usuarios crear una cuenta de cliente en la aplicación web para acceder a funcionalidades personalizadas y guardar preferencias.
Precondiciones	Acceso a la aplicación web; Usuario no autenticado.

Tabla 2.8: RF 01.4: Registrarse como Cliente

RF 01.5	Iniciar Sesión (Cliente)
Actores	ACT-01, ACT-02
Descripción	Permite a los clientes registrados autenticarse en el sistema utilizando sus credenciales (email y contraseña).
Precondiciones	Acceso a la aplicación web; Cuenta de cliente existente; Usuario no autenticado.

Tabla 2.9: RF 01.5: Iniciar Sesión (Cliente)

RF 01.6	Enviar Mensaje de Contacto
Actores	ACT-01, ACT-02
Descripción	Permite a cualquier usuario (visitante o cliente registrado) enviar una consulta general o un mensaje a la inmobiliaria a través de un formulario web.
Precondiciones	Acceso a la aplicación web.

Tabla 2.10: RF 01.6: Enviar Mensaje de Contacto

RF 01.7	Solicitar Cita para Propiedad
Actores	ACT-02
Descripción	Permite a un cliente registrado programar una visita para una propiedad específica, indicando la fecha y hora preferidas. La solicitud será enviada a Airtable para su gestión por el agente.
Precondiciones	Cliente autenticado (sesión iniciada); Propiedad existente y disponible para citas.

Tabla 2.11: RF 01.7: Solicitar Cita para Propiedad

### 2.2.2.2 Requisitos Funcionales del Sistema de Gestión Interna (Airtable)

RF 02.1	Crear Nueva Propiedad
Actores	ACT-03, ACT-04
Descripción	Permitir a los agentes y administradores añadir nuevos inmuebles al inventario utilizando los formularios personalizados de Airtable, incluyendo todos los detalles de la propiedad y la carga de fotos.
Precondiciones	Acceso a Airtable con rol ACT-03 o ACT-04.

Tabla 2.12: RF 02.1: Crear Nueva Propiedad

RF 02.2	Consultar Propiedades (en Airtable)
Actores	ACT-03, ACT-04
Descripción	Permitir a los agentes y administradores visualizar la información detallada de las propiedades en diversas vistas personalizadas de Airtable (grid, galería de fotos, vista de mapa, etc.).
Precondiciones	Acceso a Airtable con rol ACT-03 o ACT-04.

Tabla 2.13: RF 02.2: Consultar Propiedades (en Airtable)

RF 02.3	Actualizar Información de Propiedad
Actores	ACT-03, ACT-04
Descripción	Permitir a los agentes y administradores modificar los datos existentes de las propiedades en Airtable, incluyendo cambios de precio, estado, descripción o características.
Precondiciones	Acceso a Airtable con rol ACT-03 o ACT-04; Propiedad existente.

Tabla 2.14: RF 02.3: Actualizar Información de Propiedad

RF 02.4	Eliminar Propiedad
Actores	ACT-03, ACT-04
Descripción	Permitir a los agentes y administradores dar de baja o eliminar propiedades del inventario en Airtable.
Precondiciones	Acceso a Airtable con rol ACT-03 o ACT-04; Propiedad existente.

Tabla 2.15: RF 02.4: Eliminar Propiedad

RF 03.1	Programar y Modificar Citas
Actores	ACT-03, ACT-04
Descripción	Permitir a los agentes y administradores crear nuevas citas para los clientes o ajustar los detalles de citas ya programadas en Airtable, incluyendo fecha, hora, cliente y propiedad asociada.
Precondiciones	Acceso a Airtable con rol ACT-03 o ACT-04; Cliente y/o Propiedad existente.

Tabla 2.16: RF 03.1: Programar y Modificar Citas

RF 03.2	Visualizar Calendario de Citas
Actores	ACT-03, ACT-04
Descripción	Permitir a los agentes y administradores consultar la agenda de citas programadas en un formato de calendario interactivo dentro de Airtable, facilitando la planificación y seguimiento diario.
Precondiciones	Acceso a Airtable con rol ACT-03 o ACT-04.

Tabla 2.17: RF 03.2: Visualizar Calendario de Citas

RF 03.3	Actualizar Estado de Cita
Actores	ACT-03, ACT-04
Descripción	Permitir a los agentes y administradores cambiar el estado de una cita (ej., de "Pendiente" a "Confirmada", "Realizada" o "Cancelada") en Airtable, reflejando el progreso de la interacción con el cliente.
Precondiciones	Acceso a Airtable con rol ACT-03 o ACT-04; Cita existente.

Tabla 2.18: RF 03.3: Actualizar Estado de Cita

RF 04.1	Registrar Nuevo Cliente
Actores	ACT-03, ACT-04
Descripción	Permitir a los agentes y administradores registrar la información de nuevos clientes en la base de datos de Airtable a través de formularios, incluyendo datos de contacto y preferencias iniciales.
Precondiciones	Acceso a Airtable con rol ACT-03 o ACT-04.

Tabla 2.19: RF 04.1: Registrar Nuevo Cliente

RF 04.2	Consultar y Actualizar Cliente
Actores	ACT-03, ACT-04
Descripción	Permitir a los agentes y administradores acceder y modificar los detalles de los clientes existentes en Airtable, incluyendo sus preferencias de vivienda y el historial de interacciones.
Precondiciones	Acceso a Airtable con rol ACT-03 o ACT-04; Cliente existente.

Tabla 2.20: RF 04.2: Consultar y Actualizar Cliente

RF 05.1	Visualizar Análisis de Stock
Actores	ACT-03, ACT-04
Descripción	Proporcionar a los agentes y administradores acceso a dashboards visuales en Airtable con métricas generales del inventario de propiedades (total, disponibles, precio medio, superficie media, propiedades por tipo y estado) para una visión ejecutiva del negocio.
Precondiciones	Acceso a Airtable con rol ACT-03 o ACT-04.

Tabla 2.21: RF 05.1: Visualizar Análisis de Stock

RF 05.2	Visualizar Media de Precios por Tipo de Inmueble
Actores	ACT-03, ACT-04
Descripción	Permitir el acceso a gráficos y métricas que muestran el valor promedio de las propiedades desglosado por su tipo (ej. apartamentos, casas, oficinas), facilitando el análisis de tendencias de mercado.
Precondiciones	Acceso a Airtable con rol ACT-03 o ACT-04.

Tabla 2.22: RF 05.2: Visualizar Media de Precios por Tipo de Inmueble

RF 05.3	Visualizar Visitas por Propiedad
Actores	ACT-03, ACT-04
Descripción	Permitir a los usuarios monitorear el número de citas (visitas) asociadas a cada propiedad, incluyendo un detalle de las mismas, para evaluar el interés y la demanda de inmuebles.
Precondiciones	Acceso a Airtable con rol ACT-03 o ACT-04.

Tabla 2.23: RF 05.3: Visualizar Visitas por Propiedad

RF 05.4	Visualizar Tiempo Medio de Venta/Rotación
Actores	ACT-03, ACT-04
Descripción	Proporcionar acceso a datos y métricas que informan sobre el tiempo promedio que una propiedad permanece en el mercado hasta su venta o alquiler, desglosado por tipo de inmueble, para evaluar la eficiencia operativa.
Precondiciones	Acceso a Airtable con rol ACT-03 o ACT-04.

Tabla 2.24: RF 05.4: Visualizar Tiempo Medio de Venta/Rotación

RF 05.5	Visualizar Propiedades con Baja Rotación
Actores	ACT-03, ACT-04
Descripción	Identificar y listar las propiedades que han permanecido en el inventario por un período de tiempo superior a un umbral definido, mostrando sus detalles y días en el mercado para una gestión proactiva.
Precondiciones	Acceso a Airtable con rol ACT-03 o ACT-04.

Tabla 2.25: RF 05.5: Visualizar Propiedades con Baja Rotación

RF 05.6	Visualizar Distribución por Tipo de Propiedad
Actores	ACT-03, ACT-04
Descripción	Entender la composición del inventario de propiedades mediante gráficos que muestran la cantidad de inmuebles por cada tipo (ej. apartamentos, casas, chalets), ayudando a la estrategia de negocio.
Precondiciones	Acceso a Airtable con rol ACT-03 o ACT-04.

Tabla 2.26: RF 05.6: Visualizar Distribución por Tipo de Propiedad

RF 06.1	Administrar Usuarios Internos y Roles
Actores	ACT-04
Descripción	Permitir al administrador crear, modificar y eliminar registros de agentes o administradores directamente en Airtable, y asignar o ajustar sus roles de acceso para gestionar el personal interno.
Precondiciones	Acceso a Airtable con rol ACT-04.

Tabla 2.27: RF 06.1: Administrar Usuarios Internos y Roles

RF 06.2	Configurar y Supervisar Automatizaciones
Actores	ACT-03, ACT-04
Descripción	Permitir a los usuarios internos ajustar las reglas de automatización en Airtable (ej., envío de notificaciones por email al agente al recibir un contacto o al cliente tras confirmar una cita) y monitorear su ejecución para optimizar flujos de trabajo.
Precondiciones	Acceso a Airtable con rol ACT-03 o ACT-04.

Tabla 2.28: RF 06.2: Configurar y Supervisar Automatizaciones

RF 06.3	Personalizar Vistas y Formularios (Airtable)
Actores	ACT-04
Descripción	Permitir al administrador diseñar y ajustar las vistas de datos (grid, calendario, galería, mapa) y los formularios de entrada de datos en Airtable para optimizar la visualización y gestión operativa del equipo.
Precondiciones	Acceso a Airtable con rol ACT-04.

Tabla 2.29: RF 06.3: Personalizar Vistas y Formularios (Airtable)

### 2.2.3 Requisitos No Funcionales

RNF-01.1	Tiempo de Carga de Páginas
Tipo de requisito	No Funcional - Rendimiento
Descripción	Las páginas de la aplicación web deben cargar en menos de 3 segundos con conexión estándar de banda ancha.
Criterio de Aceptación	Tiempo de carga $\leq$ 3 segundos medido con herramientas como Lighthouse

Tabla 2.30: Requisito RNF-01.1

RNF-01.2	Sincronización con AirTable
Tipo de requisito	No Funcional - Rendimiento
Descripción	Los cambios realizados en AirTable deben reflejarse en la aplicación web en un tiempo máximo de 5 minutos.
Criterio de Aceptación	Latencia de sincronización $\leq$ 5 minutos para cualquier cambio en AirTable

Tabla 2.31: Requisito RNF-01.2

RNF-01.3	Optimización de Imágenes
Tipo de requisito	No Funcional - Rendimiento
Descripción	Las imágenes de propiedades deben optimizarse automáticamente con lazy loading y compresión para mejorar tiempos de carga.
Criterio de Aceptación	Imágenes con compresión automática y carga diferida implementada

Tabla 2.32: Requisito RNF-01.3

RNF-02.1	Diseño Responsive
Tipo de requisito	No Funcional - Usabilidad
Descripción	La aplicación debe ser completamente funcional y visualmente correcta en dispositivos móviles, tabletas y escritorio.
Criterio de Aceptación	Compatibilidad verificada en resoluciones desde 320px hasta 1920px

Tabla 2.33: Requisito RNF-02.1

RNF-02.2	Interfaz Intuitiva
Tipo de requisito	No Funcional - Usabilidad
Descripción	La navegación debe ser intuitiva permitiendo que un usuario nuevo pueda realizar tareas básicas sin entrenamiento previo.
Criterio de Aceptación	Máximo 3 clics para acceder a cualquier funcionalidad principal

Tabla 2.34: Requisito RNF-02.2

RNF-02.3	Accesibilidad Web
Tipo de requisito	No Funcional - Usabilidad
Descripción	La aplicación debe cumplir con estándares WCAG 2.1 nivel AA para garantizar accesibilidad a usuarios con discapacidades.
Criterio de Aceptación	Cumplimiento verificado con herramientas de auditoría de accesibilidad

Tabla 2.35: Requisito RNF-02.3

RNF-03.1	Compatibilidad de Navegadores
Tipo de requisito	No Funcional - Compatibilidad
Descripción	La aplicación debe funcionar correctamente en las últimas 2 versiones de Chrome, Firefox, Safari y Edge.
Criterio de Aceptación	Funcionalidad completa verificada en navegadores especificados

Tabla 2.36: Requisito RNF-03.1

RNF-03.2	Compatibilidad Móvil
Tipo de requisito	No Funcional - Compatibilidad
Descripción	La aplicación debe ser compatible con iOS 12+ y Android 8+ manteniendo funcionalidad completa.
Criterio de Aceptación	Pruebas exitosas en dispositivos con sistemas operativos especificados

Tabla 2.37: Requisito RNF-03.2

RNF-04.1	Autenticación Segura
Tipo de requisito	No Funcional - Seguridad
Descripción	Las contraseñas deben almacenarse hasheadas y la autenticación debe implementar tokens JWT con expiración automática.
Criterio de Aceptación	Implementación de bcrypt para passwords y JWT con expiración de 24h

Tabla 2.38: Requisito RNF-04.1

RNF-04.2	Protección contra Ataques
Tipo de requisito	No Funcional - Seguridad
Descripción	La aplicación debe implementar protección contra ataques XSS, CSRF y SQL injection.
Criterio de Aceptación	Sanitización de inputs y headers de seguridad implementados

Tabla 2.39: Requisito RNF-04.2

RNF-04.3	Conexión Segura
Tipo de requisito	No Funcional - Seguridad
Descripción	Toda la comunicación entre cliente y servidor debe realizarse a través de HTTPS con certificado SSL válido.
Criterio de Aceptación	Certificado SSL implementado y redirección automática HTTP a HTTPS

Tabla 2.40: Requisito RNF-04.3

RNF-05.1	Tiempo de Actividad
Tipo de requisito	No Funcional - Disponibilidad
Descripción	La aplicación debe mantener un tiempo de actividad del 99.5 % mensual excluyendo mantenimientos programados.
Criterio de Aceptación	Uptime $\geq$ 99.5 % medido mensualmente

Tabla 2.41: Requisito RNF-05.1

RNF-05.2	Recuperación ante Fallos
Tipo de requisito	No Funcional - Disponibilidad
Descripción	En caso de fallo del servicio, la aplicación debe recuperarse automáticamente en menos de 15 minutos.
Criterio de Aceptación	Tiempo de recuperación $\leq$ 15 minutos con monitoreo automático

Tabla 2.42: Requisito RNF-05.2

RNF-06.1	Capacidad de Usuarios Concurrentes
Tipo de requisito	No Funcional - Escalabilidad
Descripción	La aplicación debe soportar al menos 100 usuarios concurrentes sin degradación significativa del rendimiento.
Criterio de Aceptación	Pruebas de carga exitosas con 100+ usuarios simultáneos

Tabla 2.43: Requisito RNF-06.1

RNF-06.2	Crecimiento de Datos
Tipo de requisito	No Funcional - Escalabilidad
Descripción	El sistema debe manejar eficientemente el crecimiento hasta 10,000 propiedades y 50,000 usuarios registrados.
Criterio de Aceptación	Rendimiento mantenido con volúmenes de datos especificados

Tabla 2.44: Requisito RNF-06.2

RNF-07.1	Código Documentado
Tipo de requisito	No Funcional - Mantenibilidad
Descripción	El código debe estar documentado siguiendo estándares de la industria con comentarios claros y documentación técnica actualizada.
Criterio de Aceptación	Cobertura de documentación $\geq$ 80 % del código y README detallado

Tabla 2.45: Requisito RNF-07.1

RNF-07.2	Arquitectura Modular
Tipo de requisito	No Funcional - Mantenibilidad
Descripción	El código debe seguir principios de arquitectura limpia con separación clara de responsabilidades y componentes reutilizables.
Criterio de Aceptación	Implementación de patrones de diseño y separación por capas

Tabla 2.46: Requisito RNF-07.2

RNF-08.1	API de AirTable
Tipo de requisito	No Funcional - Integración
Descripción	La integración con AirTable debe manejar correctamente los límites de rate limit (5 requests por segundo) con reintentos automáticos.
Criterio de Aceptación	Implementación de rate limiting y cola de reintentos

Tabla 2.47: Requisito RNF-08.1

RNF-08.2	Manejo de Errores de API
Tipo de requisito	No Funcional - Integración
Descripción	La aplicación debe manejar graciosamente errores de conexión con AirTable mostrando mensajes apropiados al usuario.
Criterio de Aceptación	Fallbacks implementados y mensajes de error user-friendly

Tabla 2.48: Requisito RNF-08.2

RNF-09.1	Branding Dinámico
Tipo de requisito	No Funcional - Configurabilidad
Descripción	Los cambios de branding (colores, logo) desde AirTable deben aplicarse dinámicamente sin requerir redeploy de la aplicación.
Criterio de Aceptación	Cambios de branding visibles inmediatamente con caché invalidation

Tabla 2.49: Requisito RNF-09.1

RNF-09.2	Variables de Entorno
Tipo de requisito	No Funcional - Configurabilidad
Descripción	Todas las configuraciones críticas deben manejarse a través de variables de entorno para facilitar despliegues.
Criterio de Aceptación	Configuración externalizada sin hardcoding de valores sensibles

Tabla 2.50: Requisito RNF-09.2

RNF-10.1	Protección de Datos
Tipo de requisito	No Funcional - Legal
Descripción	La aplicación debe cumplir con RGPD para protección de datos personales incluyendo derecho al olvido y portabilidad.
Criterio de Aceptación	Implementación de políticas de privacidad y gestión de consentimientos

Tabla 2.51: Requisito RNF-10.1

### 2.3. CASOS DE USO

RNF-10.2	Cookies y Consentimiento
Tipo de requisito	No Funcional - Legal
Descripción	La aplicación debe implementar banner de cookies y gestión de consentimientos según normativa europea.
Criterio de Aceptación	Banner de cookies funcional con opciones de aceptación grupal

Tabla 2.52: Requisito RNF-10.2

## 2.3 CASOS DE USO

A partir de la exhaustiva definición de requisitos funcionales y no funcionales detallada previamente, se han diseñado los diagramas de casos de uso del sistema **InmoTable**. Estos diagrama, que podemos encontrar en las Figuras 2.1 y 2.2, es fundamental para esquematizar el desarrollo y comprender claramente las funcionalidades de la aplicación desde la perspectiva de sus usuarios.

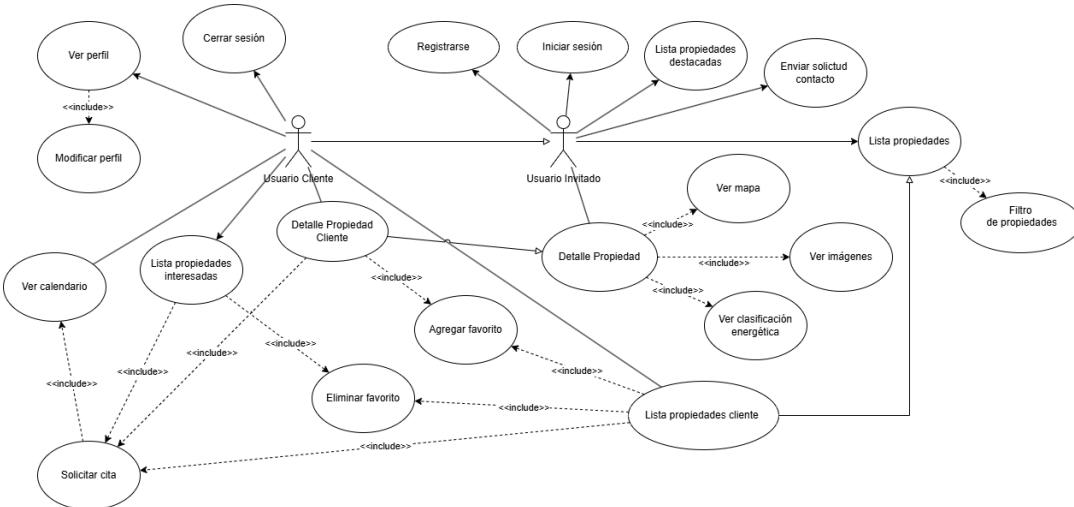
En estos diagramas se establecen las relaciones entre los distintos actores del sistema y las interacciones o funciones específicas que cada uno puede realizar.

### 2.3.1 Aplicación web

Para comprender claramente las funcionalidades de la aplicación web de **InmoTable** desde la perspectiva del usuario hay que entender que participan dos actores.

- **Sin Usuario (ACT-01):** Se trata de un usuario que no está registrado en la aplicación, y sus funciones son registrarse, iniciar sesión, ver listado de propiedades destacadas, enviar una solicitud de contacto, ver listado de propiedades, filtrar las propiedades, ver con más detalle las características de las propiedades.
- **Usuario Cliente (ACT-02):** Es el usuario que está registrado en la aplicación. Tiene la capacidad de realizar todas las funcionalidades disponibles para el usuario invitado. Además, puede ver y modificar su perfil, añadir propiedades a favoritos, eliminar favoritos, ver listado de propiedades interesadas, consultar el calendario y cerrar sesión.

### 2.3. CASOS DE USO



**Figura 2.1:** Casos de uso de la aplicación web

#### 2.3.2 Airtable

El diagrama representa las funcionalidades principales de la aplicación web de Airtable desde la perspectiva de dos actores principales:

- **Agente (ACT-03):** Es el usuario encargado de la gestión de la inmobiliaria desde la web de AirTable. Como iniciar sesión, gestionar propiedades, citas, contacto, clientes, empresas, agentes usuarios. Además de ver los listados de las interfaces que son las que se comunican con el frontend, y consultar el listado de formularios.
- **Administrador (ACT-04):** Es el usuario con el nivel más alto de permisos en la aplicación. Además de realizar todas las funciones del agente inmobiliario, puede gestionar usuarios de AirTable, configurar automatizaciones, gestionar la arquitectura del sistema.



**Figura 2.2:** Casos de uso de AirTable

## 2.4 ANÁLISIS DE COSTES

El análisis de costes es un componente fundamental de la viabilidad económica de **InmoTable**, destacando cómo la arquitectura híbrida adoptada permite una optimización significativa de los recursos financieros en comparación con el desarrollo y mantenimiento de una solución inmobiliaria tradicional. Los ahorros se manifiestan en varias fases del ciclo de vida del software y en el modelo de negocio a largo plazo.

### 1. Costes de Desarrollo (Optimización y Reducción)

La estrategia de desarrollo de **InmoTable** se ha centrado en minimizar la inversión en horas de programación personalizada para funcionalidades que Airtable puede proveer de forma nativa:

- **Reducción de Horas de Desarrollo de Gestión Interna:** El aspecto más significativo de ahorro radica en la delegación de la construcción de un panel de administración completo (CRM/ERP inmobiliario) a Airtable. Todas las funcionalidades de gestión de propiedades, clientes, citas, así como la generación de informes y dashboards analíticos, se configuran directamente en Airtable. Este enfoque ha permitido reducir considerablemente las horas de programación en el backend (Laravel) y frontend (Angular) para estas tareas, liberando tiempo para enfocarlo en la calidad de la experiencia del cliente y la robustez de la integración.
- **Eficiencia en el Desarrollo de Código Personalizado:** El tiempo de desarrollo de la aplicación web (Angular y Laravel) se ha concentrado en las funcionalidades orientadas al cliente y la capa de integración segura con la API de Airtable. Esto, junto con el uso de frameworks y librerías maduras (Angular Material, Bootstrap), ha permitido un desarrollo ágil y eficiente. La planificación temporal detallada en la sección correspondiente (Planificación Temporal) refleja cómo, a pesar de una mayor inversión real de horas (395 horas totales) para lograr un producto completo y diferenciado, el esfuerzo es significativamente menor que si se hubiera intentado construir todas las funcionalidades de gestión interna desde cero.

### 2. Costes de Licenciamiento y Plataforma

La elección de Airtable impacta directamente en los costes de software y plataforma, diferenciándose claramente de otras soluciones. La siguiente tabla resume la comparativa, enfocándose en las implicaciones monetarias y el esfuerzo de desarrollo asociado a la gestión interna para agentes:

## 2.4. ANÁLISIS DE COSTES

Servicio / Característica	Costo Inicial (Plan Gratuito)	Costo Mensual Típico (Para Equipo / Uso Medio)	Ahorro Directo en Desarrollo de UI/UX Admin	Coste de Replicabilidad / Expansión (por nueva inmobiliaria)
Airtable	Sí. 1.200 reg./base, 2GB adj. <b>Incluye Interfaces y Automatizaciones.</b>	Aprox. \$20-\$45 por usuario (planes Team/Business, escalable por funciones y volumen).	Máximo ahorro. Interfaces de gestión (dashboards, formularios, vistas) preconstruidas y configurables sin código.	Muy Bajo. Copiar Base de Airtable y mínimos cambios de variables de entorno en código.
Notion	Sí. 1.000 bloques, 5MB adj. DBs simples.	Aprox. \$8-\$15 por usuario (planes Plus/Business, por bloques/colaboradores).	Bajo. Bases de datos más simples, requiere desarrollo para dashboards/lógica analítica avanzada.	Bajo-Medio. Posible replicar espacios, pero puede implicar configuración manual y desarrollo de lógica.
Smartsheet	No (trial limitado). Versión pagada obligatoria para funcionalidad real.	Aprox. \$7-\$25 por usuario (planes Pro-Business).	Bajo. Más orientada a hojas de cálculo, requiere desarrollo de UI para gestión tipo CRM.	Medio. Replicación orientada a proyectos, no tan directa como clonar una base de negocio.
Firebase	Sí. Límites generosos de uso (reads/writes/almacenamiento).	Muy Variable (Pay-as-you-go). Puede ser bajo con poco uso o alto con mucho tráfico.	Nulo. Requiere desarrollo completo y costoso de todo el panel de administración/backend para agentes.	Medio. La clonación del backend es posible, pero el coste de desarrollar y replicar la UI de gestión persiste.
Supabase	Sí. Límites de DB/e-gress/funciones.	Aprox. \$25 (plan Pro), varía por recursos consumidos.	Nulo. Requiere desarrollo completo y costoso de todo el panel de administración/backend para agentes.	Medio. La clonación del backend es posible, pero el coste de desarrollar y replicar la UI de gestión persiste.

Tabla 2.53: Comparativa de Costes y Enfoque de Airtable con Competidores Clave

- **Conclusión de la Comparativa:** La tabla anterior ilustra que, si bien existen diversas plataformas con capacidades superpuestas, Airtable emerge como la opción más adecuada y costo-efectiva para el propósito específico de **InmoTable**. Su modelo de precios, la robustez de su API, y especialmente su suite de funcionalidades *low-code* para la gestión visual (formularios, vistas, interfaces analíticas) y automatización directamente accesibles a usuarios no técnicos y con un coste inicial nulo (plan gratuito para iniciar), representan un ahorro de desarrollo significativo y un valor añadido que alternativas como Firebase o Supabase no ofrecen en su capa de gestión visual. Este enfoque minimiza drásticamente la inversión inicial en desarrollo personalizado de paneles de administración y facilita la replicabilidad del modelo de negocio de **InmoTable** para futuras inmobiliarias con un coste marginal muy bajo.

### 3. Costes Operativos y de Mantenimiento

La arquitectura del proyecto también contribuye a una eficiencia a largo plazo en los costes de operación y mantenimiento:

- **Menor Carga de Mantenimiento de Código Personalizado:** Al delegar funcionalidades críticas de gestión a Airtable, la cantidad de código personalizado a mantener y actualizar se reduce. Esto implica menos bugs potenciales asociados al desarrollo de funciones de administración y una mayor agilidad en las actualizaciones del sistema, ya que las mejoras en Airtable se implementan y mantienen por el propio proveedor.
- **Facilidad de Uso y Menor Necesidad de Formación:** La interfaz intuitiva y amigable de Airtable reduce significativamente la curva de aprendizaje y la necesidad de formación intensiva para los agentes inmobiliarios, lo que se traduce en ahorros en costes de capacitación y una mayor productividad del personal desde el inicio.
- **Automatización para la Eficiencia Operativa:** Las automatizaciones incorporadas en Airtable reducen la dependencia de tareas manuales repetitivas (ej. envío de notificaciones), lo que optimiza el tiempo del personal y minimiza errores, contribuyendo a la eficiencia operativa y, por ende, a la reducción de costes a largo plazo.

### 4. Potencial de Monetización y Replicabilidad (Ventaja Económica Adicional)

Más allá de los ahorros para una única inmobiliaria, el modelo de costes de **InmoTable** es una base sólida para un modelo de negocio escalable:

- **Bajo Coste Marginal de Implementación:** La arquitectura modular y la centralización de la gestión en Airtable permiten que, una vez desarrollado el sistema, su adaptación a nuevas inmobiliarias sea extremadamente rápida y de bajo coste. Esto convierte al proyecto en una solución "plug-and-play" que puede generar ingresos adicionales con una inversión mínima por cada nueva instancia, maximizando el retorno de la inversión inicial del desarrollo.

En conclusión, el análisis de costes de **InmoTable** subraya que la estrategia de utilizar una arquitectura híbrida con Airtable no solo es una solución técnicamente viable y operativamente eficiente, sino que también representa una decisión económicamente inteligente que optimiza los recursos de desarrollo y operación, y abre un prometedor modelo de negocio replicable en el sector inmobiliario.

- **Optimización de Costes de Desarrollo Iniciales:** La estrategia de delegar las funcionalidades de gestión interna a Airtable reduce drásticamente las horas de desarrollo que se habrían invertido en construir un panel de administración personalizado para un CRM/ERP inmobiliario desde cero. Esto se traduce en un ahorro significativo en la inversión inicial en desarrollo de software, lo que hace el proyecto más accesible y atractivo.

## 2.4. ANÁLISIS DE COSTES

- **Modelo de Coste Escalable y Asequible de Airtable:** Airtable ofrece un plan básico gratuito, lo que permite a las inmobiliarias (especialmente las pequeñas y medianas) comenzar a utilizar el sistema sin una inversión inicial en licencias de software de base de datos. A medida que crecen, pueden escalar a planes de pago con costes asequibles, asegurando que la inversión se ajuste a sus necesidades y recursos.
- **Reducción de Costes de Mantenimiento y Operación:** Al aprovechar la plataforma Airtable, la inmobiliaria se beneficia de un mantenimiento y una infraestructura de base de datos gestionados por un tercero, lo que reduce los costes operativos internos asociados a la administración de servidores y bases de datos.
- **Alto Potencial de Replicabilidad (Modelo "Plug-and-Play"):** Como se mencionó en la propuesta de valor, la capacidad de la solución para ser "clonada" y adaptada a nuevas inmobiliarias con un esfuerzo mínimo de reconfiguración (cambiando solo las claves de API y variables de entorno en la aplicación web), la convierte en un producto de bajo coste marginal de implementación para nuevos clientes. Esto es una ventaja económica sustancial y abre la puerta a un modelo de negocio rentable y escalable.

# 3 MODELOS DE DESARROLLO Y TECNOLOGÍAS

En este capítulo se detalla la metodología Scrum empleada y justifica la elección de Angular, Laravel y Airtable frente a otras opciones.

## 3.1 METODOLOGÍAS DE DESARROLLO

La elección de una metodología de desarrollo de software es un pilar fundamental en la planificación y ejecución de cualquier proyecto, ya que define el marco de trabajo, la organización del equipo, la interacción con el cliente y la gestión de los cambios. Para el desarrollo de **InmoTable**, se consideraron diversas metodologías, tanto tradicionales como ágiles, para seleccionar la que mejor se adaptara a las características, objetivos y recursos del proyecto.

A continuación, se presenta un estudio y comparativa de las metodologías más relevantes, seguido de la justificación de la elección final.

### 3.1.1 Comparativa con Otras Metodologías Comunes

#### 3.1.1.1 Metodologías Tradicionales

##### 1. Metodología en Cascada (Waterfall Model):

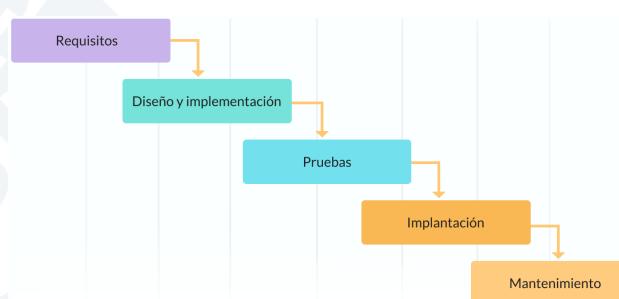


Figura 3.1: Diagrama de la Metodología en Cascada (Waterfall Model)

- **Descripción:** Es un enfoque lineal y secuencial, donde el desarrollo fluye de una fase a la siguiente de forma descendente: requisitos, diseño, implementación, pruebas, despliegue y mantenimiento. Cada fase tiene entregables específicos que deben ser aprobados antes de avanzar [3].
- **Ventajas:** Simplicidad, fácil gestión en proyectos pequeños y con requisitos muy estables, buena para documentación detallada.

- **Desventajas:** Poca flexibilidad ante cambios, los errores suelen detectarse tarde (en la fase de pruebas), la participación del cliente se limita a las fases iniciales. Resulta arriesgado para proyectos de mediana o alta complejidad con requisitos cambiantes.

## 2. Modelo en V (V-Model):

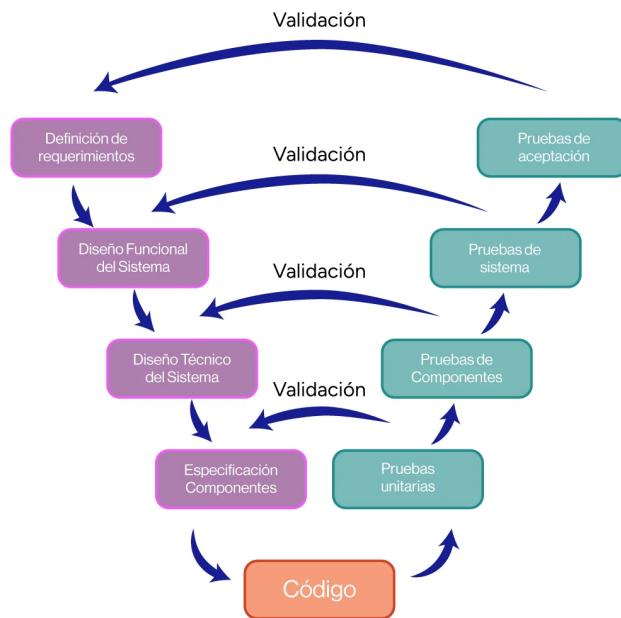


Figura 3.2: Diagrama del Modelo en V (V-Model)

- **Descripción:** Una extensión del modelo en cascada que enfatiza las actividades de verificación y validación. Cada fase de desarrollo tiene una fase de prueba correspondiente. Por ejemplo, el diseño de requisitos del sistema se valida con pruebas de aceptación del usuario [3].
- **Ventajas:** Mejora la calidad al integrar pruebas tempranas, adecuado para proyectos donde la fiabilidad es crítica.
- **Desventajas:** Similar a la cascada en su rigidez y falta de flexibilidad. Los cambios tardíos siguen siendo costosos.

### 3. Modelo en Espiral (Spiral Model):



Figura 3.3: Diagrama del Modelo en Espiral (Spiral Model)

- **Descripción:** Combina elementos de los modelos en cascada y prototipado, incorporando iteraciones con énfasis en la gestión de riesgos en cada vuelta de la espiral. Cada bucle de la espiral representa una fase de planificación, análisis de riesgos, ingeniería y evaluación [3].
- **Ventajas:** Fuerte gestión de riesgos, permite la construcción de prototipos, más flexible que cascada y V.
- **Desventajas:** Es más complejo de gestionar, costoso para proyectos pequeños y requiere experiencia en la evaluación de riesgos.

#### 3.1.1.2 Metodologías Ágiles

##### 1. Scrum:

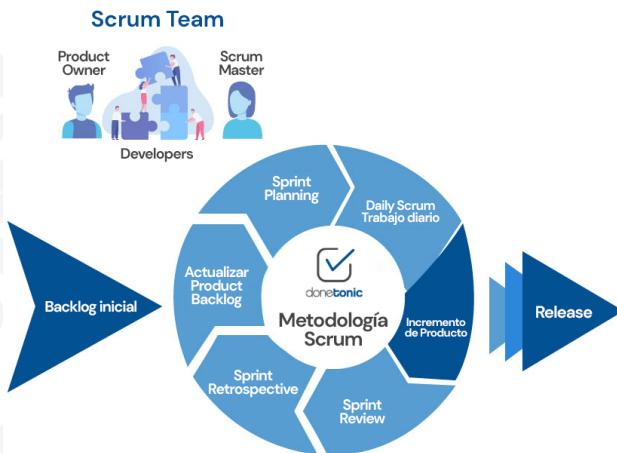


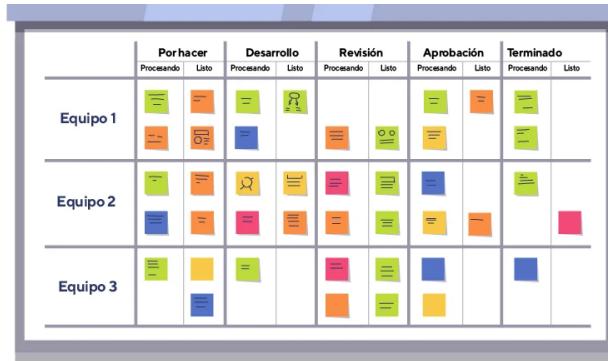
Figura 3.4: Diagrama de la Metodología Scrum

- **Descripción:** Es un marco de trabajo iterativo e incremental para gestionar proyectos complejos. Se organiza en ciclos cortos y fijos llamados sprints (generalmente de 1 a 4 semanas), al final de los cuales se entrega un incremento de producto funcional. Involucra roles definidos (Product Owner, Scrum Master, Equipo de Desarrollo) y ceremonias (Sprint Planning, Daily Scrum, Sprint Review, Sprint Retrospective) [4] [5].

### 3.1. METODOLOGÍAS DE DESARROLLO

- **Ventajas:** Alta adaptabilidad al cambio, entrega temprana y continua de valor al cliente, mejora la calidad a través de la retroalimentación constante, alta visibilidad del progreso.
- **Desventajas:** Puede ser difícil de implementar en organizaciones no preparadas, requiere compromiso del equipo y del Product Owner, el alcance del proyecto puede parecer menos predecible a largo plazo.

#### 2. Kanban:



**Figura 3.5:** Diagrama de la Metodología Kanban

- **Descripción:** Un método visual para gestionar el trabajo, centrado en la visualización del flujo de trabajo, la limitación del trabajo en progreso (WIP) y la maximización de la eficiencia del flujo. Se utiliza un tablero Kanban para representar las fases del proceso [4].
- **Ventajas:** Muy flexible, ideal para flujos de trabajo de mantenimiento o con cambios constantes, mejora la visibilidad de los cuellos de botella.
- **Desventajas:** Menos estructurado que Scrum, puede ser menos adecuado para proyectos con entregables de tamaño fijo o con fechas de lanzamiento específicas sin una gestión activa.

### 3. Extreme Programming (XP):

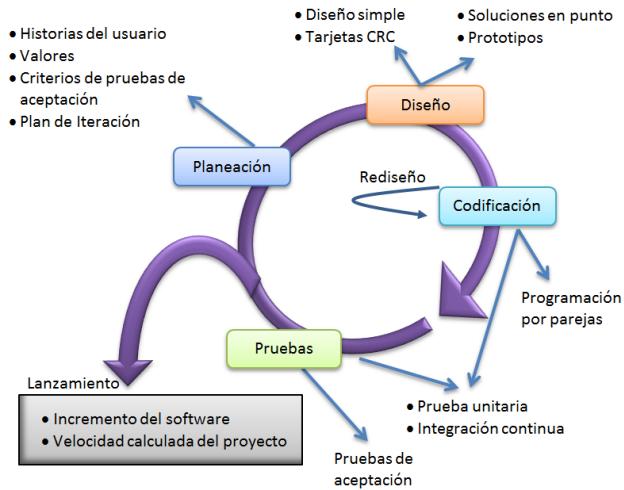


Figura 3.6: Diagrama de la Metodología Extreme Programming (XP)

- **Descripción:** Una metodología ágil que se enfoca en la ingeniería de software de alta calidad y la mejora continua. Destaca prácticas como el desarrollo impulsado por pruebas (TDD), programación en parejas, integración continua, refactorización y la simplicidad del diseño [4].
- **Ventajas:** Produce software de muy alta calidad, promueve buenas prácticas de ingeniería, retroalimentación constante.
- **Desventajas:** Requiere gran disciplina del equipo, puede ser difícil de implementar si no hay compromiso con sus prácticas rigurosas.

#### 3.1.1.3 DevOps

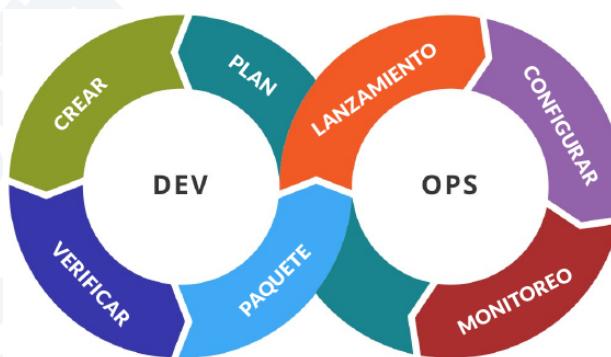


Figura 3.7: Diagrama del Esquema tareas DevOps

- **Descripción:** No es una metodología de desarrollo per se, sino un conjunto de prácticas culturales, herramientas y filosofías que buscan integrar los equipos de desarrollo (Dev) y operaciones (Ops) para automatizar y mejorar el proceso de entrega de software. Se centra en la entrega continua, la integración continua (CI/CD), la monitorización y la retroalimentación constante [6].

### 3.1. METODOLOGÍAS DE DESARROLLO

- **Rol en el Desarrollo:** DevOps complementa a las metodologías ágiles al permitir que los incrementos de software generados en cada iteración puedan ser probados, desplegados y monitoreados de manera rápida y fiable, cerrando el ciclo de retroalimentación.
- **Ventajas:** Acelera el tiempo de comercialización (time-to-market), mejora la fiabilidad del despliegue, reduce los errores y promueve la colaboración entre equipos.

#### 3.1.2 Justificación y Razonamiento de la Elección Final (Scrum)

La decisión de adoptar **Scrum** como metodología principal para el desarrollo de **InmoTable** se basa en una comparativa exhaustiva y en la alineación directa con los objetivos y características únicas del proyecto:

- **Flexibilidad frente a Requisitos Cambiantes:** Las metodologías tradicionales (Cascada, V-Model) fueron descartadas debido a su inherente rigidez y la dificultad para adaptarse a los cambios en los requisitos del cliente o las prioridades del mercado inmobiliario. Dado que **InmoTable** es un proyecto que busca innovar y adaptarse rápidamente a las necesidades del negocio (especialmente con la flexibilidad de Airtable), un enfoque predictivo habría introducido demasiados riesgos y retrasos. El Modelo en Espiral, aunque más flexible, añade una complejidad de gestión de riesgos que no era primordial para un TFG con un equipo reducido.
- **Entrega de Valor Continua y Retroalimentación Temprana:** Scrum, con sus sprints cortos y las ceremonias de Sprint Review, permite entregar incrementos de software funcionales de forma regular. Esto es crucial para **InmoTable** ya que:
  - Permite al cliente (o al director del TFG en el rol de Product Owner) validar el progreso y proporcionar retroalimentación constante sobre la aplicación para el cliente y las interfaces de Airtable, asegurando que el desarrollo se alinea con las expectativas.
  - Facilita la adaptación a las demandas del mercado o las necesidades específicas de los agentes inmobiliarios, permitiendo la iteración sobre funcionalidades como los dashboards de análisis de Airtable.
- **Gestión de un Proyecto Híbrido e Innovador:** La naturaleza híbrida de **InmoTable**, que combina desarrollo a medida (Angular y Laravel) con una plataforma *low-code* como Airtable para la gestión interna, se beneficia enormemente de la flexibilidad de Scrum. El enfoque iterativo permite descubrir y explotar las funcionalidades de Airtable (interfaces, formularios, automatizaciones) a medida que el proyecto avanza, adaptando el desarrollo de la API de Laravel (`AirtableService.php`) a estas integraciones.
- **Visibilidad y Transparencia del Progreso:** Las ceremonias de Scrum (especialmente el Daily Scrum y la Sprint Review) proporcionan una alta visibilidad del progreso del proyecto, de los desafíos y de las próximas tareas. Esto es vital en un TFG para mantener a los directores informados y para la propia autogestión y el ajuste del cronograma (como se reflejó en la justificación de las horas reales vs. estimadas).

- **Fomento de la Autonomía y Autoorganización:** Scrum promueve que el equipo de desarrollo se autoorganice para decidir la mejor manera de realizar el trabajo. Para un TFG, esto se traduce en una mayor responsabilidad y capacidad de decisión del estudiante, lo cual es un objetivo de aprendizaje clave.
- **Comparación con otras metodologías Ágiles:**
  - **Kanban:** Aunque Kanban es excelente para la gestión de flujo continuo y mantenimiento, Scrum se consideró más adecuado para el desarrollo inicial de un producto con objetivos de incremento definidos en cada sprint.
  - **Extreme Programming (XP):** Si bien XP enfatiza la calidad del código, sus prácticas más rigurosas (como la programación en parejas constante) podrían haber introducido una complejidad adicional no necesaria para un TFG individual, aunque los principios de TDD e integración continua son deseables.

En conclusión, Scrum proporcionó el marco ágil necesario para gestionar eficazmente un proyecto con un alcance innovador y una integración tecnológica compleja. Su enfoque iterativo, la entrega incremental y la retroalimentación constante fueron clave para superar los desafíos del desarrollo y asegurar que **InmoTable** se convirtiera en una solución funcional y valiosa, no solo como aplicación web, sino también como una potente herramienta de gestión empresarial para los agentes inmobiliarios.

### 3.2 ARQUITECTURAS DE SOFTWARE

El proyecto **InmoTable** adopta una **arquitectura de software de tres capas (o N-tier)** con un enfoque distribuido y una integración híbrida, donde cada capa posee responsabilidades bien definidas y la interacción entre ellas se realiza a través de interfaces claras. Esta elección arquitectónica no es arbitraria, sino que responde a una cuidadosa evaluación de las necesidades del negocio inmobiliario y a la búsqueda de eficiencia en el desarrollo, escalabilidad y facilidad de gestión.

Las tres capas principales son:

1. **Capa de Presentación (Frontend - Angular):** Responsable de la interacción directa con el usuario final (el cliente). Se encarga de la interfaz de usuario, la lógica de presentación y la comunicación con la capa de lógica de negocio.
2. **Capa de Lógica de Negocio/Aplicación (Backend - Laravel):** Es el intermediario entre el frontend y la capa de datos. Contiene la lógica de negocio central, maneja las peticiones del cliente, realiza validaciones, gestiona la seguridad (autenticación y autorización) y coordina las operaciones con la capa de datos.
3. **Capa de Datos (Airtable):** Aunque tradicionalmente solo se ocupa del almacenamiento, en este proyecto Airtable asume un rol ampliado. No solo es el repositorio principal de la información, sino que, gracias a sus funcionalidades *low-code/no-code*, también encapsula gran parte de la lógica de negocio y presentación orientada a la gestión interna de la empresa (para los agentes inmobiliarios).

La distribución del entorno a través de Docker refuerza la modularidad y facilita el despliegue y la consistencia del entorno de desarrollo y producción.

### 3.2.1 Justificación de la Arquitectura Seleccionada (Híbrida de Tres Capas)

La elección de esta **arquitectura híbrida de tres capas** se justifica por las siguientes ventajas clave, que abordan directamente las motivaciones y objetivos del proyecto:

- **Separación Clara de Responsabilidades:** Cada capa tiene un propósito específico, lo que mejora la modularidad, la mantenibilidad y facilita el desarrollo paralelo. El frontend se enfoca en la experiencia del usuario, el backend en la lógica del negocio y la seguridad, y Airtable en la gestión flexible de datos y la automatización para el equipo interno.
- **Escalabilidad Diferenciada:** Al separar las capas, es posible escalar cada una de forma independiente según la demanda. Por ejemplo, la aplicación Angular puede ser servida por un CDN para millones de usuarios, mientras que la lógica de Laravel y las operaciones de Airtable se escalan según el volumen de transacciones y datos internos.
- **Flexibilidad y Agilidad en la Gestión de Datos (Rol de Airtable):** Airtable no es una base de datos relacional tradicional, sino una herramienta tipo spreadsheet avanzada que permite a los agentes inmobiliarios diseñar y adaptar tablas y campos sobre la marcha sin requerir intervención de desarrollo. Esta flexibilidad es ideal para un sector como el inmobiliario, con necesidades de datos cambiantes.
- **Ahorro de Tiempo y Costes de Desarrollo para la Gestión Interna:** Al externalizar gran parte de la gestión de la empresa, las interfaces de análisis (ej. "Análisis de stock", "Visitas programadas"), los formularios de entrada de datos (ej. "Nueva Propiedad") y las automatizaciones (ej. envío de emails) a Airtable, se reduce significativamente el tiempo de programación en el backend y frontend para estas funcionalidades. Esto no solo acelera el desarrollo inicial, sino que también minimiza los costes de mantenimiento y futuras personalizaciones.
- **Fácil Replicabilidad y Modelo de Negocio "Franquiable":** La separación clara y el uso de Airtable permiten que la base de datos completa de una inmobiliaria se pueda "clonar" para otra nueva, simplemente replicando la base de Airtable y ajustando las variables de entorno (claves API, nombres) en los archivos environment del frontend y backend. Esto convierte el sistema en una solución altamente replicable para múltiples clientes inmobiliarios con un esfuerzo de adaptación mínimo.
- **Interfaz Intuitiva y Colaborativa para Agentes:** Airtable, por su naturaleza de hoja de cálculo potenciada, es inherentemente intuitiva y facilita la colaboración en tiempo real, permitiendo a múltiples usuarios trabajar simultáneamente. Los agentes no necesitan ser técnicos para gestionar y visualizar sus datos.

### 3.2.2 Comparativa con Otras Arquitecturas Comunes

Para comprender mejor la idoneidad de la arquitectura elegida, es útil compararla con dos de los modelos arquitectónicos más prevalentes en el desarrollo de software [7]:

#### 1. Arquitectura Monolítica:

- **Descripción:** En una arquitectura monolítica, todas las funcionalidades de la aplicación (interfaz de usuario, lógica de negocio y capa de datos) se construyen como una única unidad interdependiente.
- **Ventajas (para algunos casos):** Simplicidad inicial en el desarrollo y despliegue para proyectos pequeños, facilidad de depuración al estar todo en un mismo código base.
- **Desventajas para InmoTable:**
  - **Rigidez:** La interdependencia entre módulos dificultaría enormemente la adaptación rápida a cambios en las necesidades del mercado inmobiliario sin afectar a otras partes del sistema.
  - **Escalabilidad Limitada:** Para escalar una parte de la aplicación (ej. la base de datos), habría que escalar toda la aplicación, lo cual es ineficiente y costoso.
  - **Acoplamiento Fuerte:** Dificultaría la integración de herramientas de terceros (como Airtable) como una capa de gestión primaria sin reescribir una lógica considerable.
  - **Costo de Mantenimiento:** Los cambios y actualizaciones pueden volverse complejos y propensos a errores a medida que el sistema crece.

#### 2. Arquitectura de Microservicios:

- **Descripción:** Un enfoque en el que una aplicación se construye como una colección de servicios pequeños, independientes y acoplados de forma laxa, cada uno ejecutándose en su propio proceso y comunicándose a través de APIs.
- **Ventajas (generales):** Alta escalabilidad, resiliencia, desarrollo independiente de cada servicio, flexibilidad tecnológica.
  - **Desventajas para InmoTable (como opción principal para todo el sistema):**
    - **Complejidad Inicial:** La implementación de microservicios introduce una sobrecarga significativa en la configuración, despliegue, monitoreo y gestión (APIs Gateway, descubrimiento de servicios, comunicación inter-servicio).
    - **Curva de Aprendizaje:** Requiere un equipo con experiencia considerable en la gestión de sistemas distribuidos, lo que puede no ser necesario para un TFG o una startup con recursos limitados.
    - **Exceso para el Caso de Uso:** Para las funcionalidades actuales de **InmoTable**, especialmente con Airtable manejando gran parte de la "micro-lógica" del negocio para agentes, una arquitectura de microservicios completa para cada

### 3.3. TECNOLOGÍAS DE DESARROLLO

pequeña funcionalidad podría ser una exageración y añadir una complejidad innecesaria. Aunque el Laravel backend podría considerarse un "servicio" en sí mismo que expone una API para el frontend.

#### 3.2.3 Razonamiento de la Elección Final

La arquitectura de tres capas distribuida, con Angular para la experiencia de usuario del cliente, Laravel como una API segura y de lógica de negocio, y Airtable como la base de datos principal y plataforma de gestión interna para los agentes, se posiciona como la opción más idónea y estratégica para **InmoTable**.

Esta elección permite evitar los inconvenientes de la arquitectura monolítica al proporcionar una clara separación de preocupaciones y una escalabilidad más flexible. Al mismo tiempo, mitiga la complejidad inicial y el exceso de ingeniería de la arquitectura de microservicios al delegar las funcionalidades de gestión interna (tablas, formularios, vistas y automatizaciones) directamente a Airtable. Esta estrategia reduce significativamente el esfuerzo de desarrollo y los costes operativos, al tiempo que proporciona a los agentes una herramienta intuitiva y potente para su trabajo diario.

En definitiva, la arquitectura de **InmoTable** es una solución pragmática y optimizada que maximiza la eficiencia del desarrollo, aprovecha las fortalezas de plataformas *low-code* y establece un modelo fácilmente replicable para futuras expansiones de negocio en el sector inmobiliario.

### 3.3 TECNOLOGÍAS DE DESARROLLO

La selección de las tecnologías de desarrollo es un factor crítico que impacta directamente en la viabilidad, escalabilidad, mantenibilidad y la experiencia del usuario de un proyecto de software. Para **InmoTable**, la elección de un stack tecnológico robusto y adecuado a la arquitectura híbrida propuesta fue fundamental para cumplir con los objetivos establecidos, combinando la eficiencia de las plataformas *low-code* con la flexibilidad del desarrollo a medida.

A continuación, se presenta un análisis de las tecnologías consideradas para cada capa del sistema y la justificación de las elecciones finales.

#### 3.3.1 Tecnologías Frontend

El frontend es la capa con la que interactúa directamente el cliente final, por lo que la elección del framework es crucial para garantizar una experiencia de usuario moderna, reactiva y fluida. Las opciones más populares en el desarrollo de Single Page Applications (SPAs) son Angular, React y Vue.js.

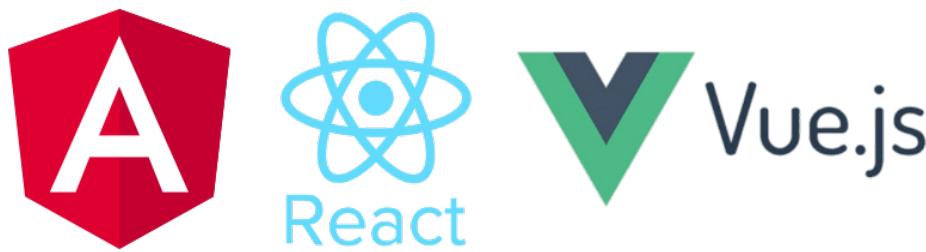


Figura 3.8: Logos de Angular, React y Vue.js

### 1. Angular:

- **Descripción:** Es un framework de desarrollo de aplicaciones web de código abierto, mantenido por Google, basado en TypeScript. Destaca por su enfoque "opinionado" que proporciona una estructura y patrones de diseño claros, lo que lo hace ideal para proyectos a gran escala y de larga duración. Ofrece un ecosistema completo con herramientas CLI, enrutamiento, gestión de estado y más. Angular se ejecuta en Node.js, un entorno de ejecución de JavaScript del lado del servidor, y utiliza npm como gestor de paquetes para instalar y gestionar sus dependencias. Para la construcción de la interfaz de usuario, se han utilizado librerías de componentes como Angular Material y el framework CSS Bootstrap para asegurar un diseño responsivo y elementos visuales consistentes y de alta calidad [8].
- **Ventajas:** Arquitectura estructurada y modular, rendimiento optimizado (Ivy), potente CLI, tipado estático con TypeScript que reduce errores, amplio soporte comunitario y de Google, ideal para aplicaciones empresariales y complejas.
- **Desventajas:** Curva de aprendizaje más pronunciada que otros frameworks, puede ser más pesado en el tamaño del bundle inicial si no se optimiza bien, mayor verbosidad en el código.

### 2. React:

- **Descripción:** Una biblioteca de JavaScript para construir interfaces de usuario, mantenida por Facebook (Meta). Se centra en la composición de componentes y en un flujo de datos unidireccional. Es más flexible que Angular, permitiendo mayor libertad en la elección de librerías para routing, gestión de estado, etc [8].
- **Ventajas:** Alta flexibilidad y menor curva de aprendizaje inicial, gran ecosistema de librerías, renderizado eficiente con Virtual DOM, muy popular en la industria.
- **Desventajas:** La libertad puede llevar a inconsistencias en proyectos grandes si no se establecen convenciones claras, requiere la integración de múltiples librerías de terceros para una aplicación completa.

### 3. Vue.js:

- **Descripción:** Un framework progresivo de JavaScript que busca ser más accesible que Angular y más estructurado que React. Es ligero, fácil de aprender y se puede adaptar desde pequeñas integraciones hasta SPAs complejas [8].

- **Ventajas:** Muy fácil de aprender y de integrar en proyectos existentes, alta reactividad, excelente rendimiento, tamaño de bundle reducido.
- **Desventajas:** Menos popular que React o Angular en grandes empresas, menor cantidad de recursos de aprendizaje avanzados en comparación.

#### Justificación de la Elección: Angular para InmoTable

La elección de **Angular** para el frontend de **InmoTable** se justifica por varios motivos alineados con los objetivos del proyecto:

- **Robustez y Escalabilidad:** Dada la intención de **InmoTable** de ser una solución de gestión inmobiliaria integral y potencialmente replicable, Angular proporciona una estructura sólida y escalable, con patrones claros que facilitan el crecimiento del proyecto y el mantenimiento a largo plazo.
- **Productividad con TypeScript:** El uso de TypeScript en Angular reduce significativamente los errores en tiempo de desarrollo gracias al tipado estático, lo que acelera la detección de fallos y mejora la calidad del código, un aspecto fundamental en un proyecto académico con plazos definidos.
- **Ecosistema Completo y Componentes UI:** Angular, al ejecutarse sobre Node.js, aprovecha su ecosistema de paquetes y herramientas. Ofrece soluciones integradas para enrutamiento (app.routes.ts, auth.routes.ts), gestión de formularios y servicios HTTP (api.service.ts). La capacidad de integrar fácilmente librerías de componentes como Angular Material y frameworks como Bootstrap agiliza la creación de interfaces de usuario visualmente atractivas y consistentes, centrándose en la experiencia del cliente. Las guardias de ruta (auth.guard.ts, role.guard.ts) demuestran su robustez en la seguridad de acceso.
- **Soporte a Largo Plazo:** El respaldo de Google y su ciclo de lanzamientos predecible garantizan la longevidad y el soporte continuo, aspectos importantes para un proyecto que podría evolucionar más allá del TFG.

#### 3.3.2 Tecnologías Backend

El backend es la capa fundamental que procesa la lógica de negocio, gestiona la persistencia de datos y expone las APIs para el frontend.



Figura 3.9: Logos de Laravel, .NET, Spring y Node.js

#### 1. Laravel (PHP):

- **Descripción:** Es un framework de PHP conocido por su sintaxis elegante y expresiva. Sigue el patrón de diseño Modelo-Vista-Controlador (MVC) y ofrece herramientas integradas para enrutamiento, ORM (Eloquent), autenticación (Passport), colas y más [9].
- **Ventajas:** Desarrollo rápido de aplicaciones web y APIs, vasta documentación y comunidad, robusto sistema de autenticación API (Passport), facilidad para integrar servicios externos (como Airtable a través de AirtableService.php), alto rendimiento para la mayoría de aplicaciones.
- **Desventajas:** Rendimiento potencialmente inferior a Node.js para aplicaciones altamente concurrentes con muchas operaciones de E/S no bloqueantes (aunque esto es marginal para el contexto del proyecto).

#### 2. .NET Core (C#):

- **Descripción:** Un framework de Microsoft de código abierto y multiplataforma para construir aplicaciones modernas, incluyendo APIs web. Utiliza el lenguaje C# y ofrece un rendimiento muy alto, especialmente en escenarios empresariales [9].
- **Ventajas:** Alto rendimiento, fuerte tipado, ecosistema robusto y maduro, excelente para aplicaciones empresariales y escalables, buen soporte de Microsoft.
- **Desventajas:** Puede tener una curva de aprendizaje más pronunciada para desarrolladores no familiarizados con el ecosistema .NET, más orientado a grandes empresas.

#### 3. Spring (Java):

- **Descripción:** Un framework de Java muy popular para construir aplicaciones empresariales de gran escala, especialmente con Spring Boot, que simplifica el desarrollo de microservicios y APIs REST [9].
- **Ventajas:** Muy robusto y maduro, vasto ecosistema de librerías, excelente rendimiento y escalabilidad, ideal para aplicaciones de nivel empresarial.
- **Desventajas:** Puede ser más complejo y verboso para proyectos más pequeños o con equipos reducidos, mayor consumo de recursos en comparación con algunos frameworks ligeros.

#### 4. Node.js (con Express o NestJS):

- **Descripción:** Un entorno de ejecución de JavaScript del lado del servidor. Permite construir APIs RESTful y aplicaciones en tiempo real utilizando JavaScript/TypeScript en todo el stack. Frameworks como Express (minimalista) o NestJS (arquitectura modular, basado en Angular) son comunes [9].
- **Ventajas:** Excelente para aplicaciones de E/S intensivas y en tiempo real, lenguaje unificado (JavaScript/TypeScript) en frontend y backend, gran comunidad.
- **Desventajas:** Puede no ser ideal para CPU-bound operations (tareas que requieren mucho procesamiento), la gestión de la complejidad puede ser un desafío sin un framework estructurado como NestJS.

### Justificación de la Elección: Laravel para InmoTable

La selección de **Laravel** como framework de backend para **InmoTable** se justifica por su equilibrio entre facilidad de desarrollo y potencia:

- **Desarrollo Ágil y Productividad:** Laravel es conocido por acelerar el desarrollo de APIs RESTful, lo que fue crucial para la fase de implementación del backend. Su sintaxis limpia y sus características incorporadas permiten codificar de manera eficiente las interacciones con la API de Airtable.
- **Robustez en Autenticación y Autorización:** La integración de Laravel Passport ofrece una solución completa y segura para la autenticación de APIs mediante OAuth2, esencial para gestionar los diferentes roles de usuario (clientes y agentes) y proteger los datos.
- **Facilidad de Integración:** Laravel simplifica la integración con APIs externas mediante su cliente HTTP robusto, lo cual es fundamental para la comunicación con Airtable. Los controladores (PropertyController.php, CitaController.php, etc.) actúan como el punto de control central para todas las operaciones de datos.
- **Coherencia con el Stack:** Al combinarlo con Angular, que también sigue un enfoque estructurado, se mantiene una coherencia en el paradigma de desarrollo, facilitando la curva de aprendizaje y la gestión del proyecto en su conjunto.

#### 3.3.3 Base de Datos y Gestión de Contenido

La capa de datos es donde reside la información crucial del negocio. La elección de la plataforma no solo se limita al almacenamiento, sino también a la provisión de herramientas de gestión interna para los agentes inmobiliarios, lo que convierte esta decisión en un pilar estratégico del proyecto.



Figura 3.10: Logo de AirTable

##### 1. Airtable:

- **Descripción:** Una plataforma *low-code* que combina las características de una base de datos relacional con la interfaz intuitiva de una hoja de cálculo. Ofrece potentes capacidades de colaboración, automatización, creación de formularios y desarrollo de interfaces (dashboards) personalizadas [10] [11] [12] [13].

■ **Ventajas Clave para InmoTable:**

- **Flexibilidad y Autonomía para Agentes:** Permite a los agentes inmobiliarios (usuarios no técnicos) diseñar, adaptar y gestionar la estructura de sus datos (tablas, campos) de forma intuitiva y en tiempo real, sin depender de un desarrollador para cada cambio. Esta adaptabilidad es clave para un sector dinámico.
- **Reducción Drástica del Desarrollo de Gestión Interna:** Las funcionalidades integradas de Airtable, como sus vistas personalizadas (calendarios, galerías, mapas), formularios de entrada de datos y automatizaciones de flujos de trabajo (envío de emails), eliminan la necesidad de programar estas complejas herramientas administrativas desde cero en el backend o frontend. Esto se traduce en un ahorro sustancial de tiempo y costes de desarrollo y mantenimiento.
- **Potentes Herramientas Analíticas para Agentes (Interfaces):** Las interfaces de Airtable diseñadas para el "Análisis de stock", "Media de precios por tipo de inmueble", "Tiempo medio de venta" y otras ofrecen a los agentes información de valor directamente, sin necesidad de un desarrollo adicional para estas visualizaciones.
- **Costo-Efectividad y Escalabilidad:** Ofrece un plan básico gratuito y es escalable a bajo coste, lo que lo hace muy atractivo para pequeñas y medianas inmobiliarias.
- **Costo-Efectividad y Escalabilidad:** La estructura de Airtable permite replicar fácilmente la base de datos completa para diferentes inmobiliarias, simplemente cambiando las claves de API y configuraciones en la aplicación web (environment.template.ts).

2. Alternativas (y por qué Airtable fue la opción preferida):

- **Bases de Datos Relacionales Tradicionales (ej. PostgreSQL, MySQL con phpMyAdmin o un CMS a medida):**



Figura 3.11: Logos de PostgreSQL, MySQL y phpMyAdmin

- **Descripción:** Sistemas de gestión de bases de datos maduros y robustos, ideales para almacenar datos estructurados y complejos. Ofrecen un control granular sobre el esquema y las relaciones. Típicamente, requieren un CMS (Content Management System) o un panel de administración personalizado para la interacción del usuario [14].

- **Razón de Descarte/Preferir Airtable:** Aunque son potentes para el almacenamiento, su principal desventaja para **InmoTable** es que requerirían un esfuerzo de desarrollo significativo para construir toda la interfaz de gestión interna para los agentes. Esto incluye el desarrollo de formularios de entrada de datos, vistas personalizadas (tablas filtradas, calendarios, galerías, mapas), cuadros de mando analíticos con gráficos y la programación de flujos de trabajo de automatización. Airtable provee todas estas funcionalidades out-of-the-box, reduciendo drásticamente la carga de programación y el tiempo de implementación para el lado de la gestión de la empresa.

- **Google Sheets + API personalizada:**



Google Sheets

**Figura 3.12:** Logo de Google Sheets

- **Descripción:** Utilizar hojas de cálculo de Google como la fuente de datos principal y construir una API (ej. con Google Apps Script o un backend como Laravel) para interactuar con ellas [15].
- **Razón de Descarte/Preferir Airtable:** Si bien Google Sheets es altamente flexible y colaborativo, carece de la robustez de una base de datos real. Sus capacidades de establecer relaciones entre hojas son limitadas y no posee funcionalidades nativas de automatización (más allá de scripts básicos) ni interfaces de visualización avanzadas como los dashboards de Airtable. Para replicar el nivel de gestión interna que Airtable ofrece, se necesitaría un desarrollo personalizado mucho mayor, anulando gran parte de la eficiencia buscada.

- **Plataformas No-Code/Low-Code (Otras):**



**Notion**



**smartsheet**



**Firebase**



**supabase**

**Figura 3.13:** Logos de Notion, Smartsheet, Firebase y Superbase

• **Notion:**

- **Descripción:** Una herramienta versátil que integra notas, bases de datos, wikis y gestión de proyectos. Es muy flexible para la organización de información de manera interconectada [10].
- **Razón de Descarte/Preferir Airtable:** Aunque Notion tiene "bases de datos", su fortaleza principal es la gestión de contenido y la creación de wikis, no el modelado de datos relacionales robustos y la exposición de una API tan orientada a datos como Airtable. Las interfaces de Airtable están más diseñadas para dashboards de negocio con métricas y gráficos, mientras que las de Notion son más generales y menos especializadas para análisis de datos estructurados complejos en un TFG.

• **Smartsheet:**

- **Descripción:** Una plataforma que combina las características de una hoja de cálculo con la gestión de proyectos, la colaboración y la automatización. Está fuertemente orientada a la planificación y el seguimiento de tareas [11].
- **Razón de Descarte/Preferir Airtable:** Smartsheet es excelente para la gestión de proyectos y flujos de trabajo basados en hojas de cálculo. Sin embargo, Airtable sobresale en ser una "base de datos con interfaz de hoja de cálculo", ofreciendo un modelo de datos más fundamentalmente relacional y una API más flexible para integraciones con aplicaciones externas como la tuya. Sus capacidades de "Interfaces" son más potentes y personalizables para el análisis de negocio puro que las de reporting de Smartsheet.

• **Firebase:**

- **Descripción:** Una plataforma Backend-as-a-Service (BaaS) de Google que ofrece servicios de autenticación, base de datos NoSQL (Firestore, Realtime Database), almacenamiento, hosting y funciones serverless [12].
- **Razón de Descarte/Preferir Airtable:** Firebase es ideal para desarrollar aplicaciones rápidamente al proporcionar un backend listo para usar. No obstante, su naturaleza NoSQL y la ausencia de una interfaz de administración built-in y amigable para usuarios no técnicos fue un factor decisivo. Para replicar la experiencia de gestión visual y las herramientas analíticas que Airtable ofrece a los agentes (formularios, dashboards, vistas interactivas), se necesitaría un esfuerzo de desarrollo de UI muy considerable en el frontend, lo que contradice el objetivo de ahorrar tiempo y costes en la gestión interna.

• **Supabase:**

- **Descripción:** Una alternativa open-source a Firebase, que ofrece una base de datos PostgreSQL, autenticación, funciones serverless y almacenamiento. Permite a los desarrolladores tener mayor control sobre su base de datos relacional.

- **Razón de Descarte/Preferir Airtable:** Aunque Supabase es una alternativa atractiva por su base de datos relacional (PostgreSQL) y su modelo open-source, comparte la misma limitación que Firebase en el contexto de **InmoTable**: no proporciona una interfaz de gestión visual *low-code* ni dashboards analíticos listos para usar para usuarios de negocio (agentes). El desarrollo de un panel de administración completo con las funcionalidades de Airtable (Análisis de stock, Media de precios, etc.) seguiría siendo una tarea de desarrollo a medida considerable, diluyendo uno de los principales beneficios de la arquitectura elegida para el TFG.

### 3.3.4 Conclusión sobre la Elección Tecnológica

La estrategia de tecnologías de **InmoTable** es una decisión deliberada para crear una solución eficiente en recursos y altamente funcional. Angular y Laravel proporcionan la potencia y seguridad necesarias para una aplicación web moderna orientada al cliente. La integración estratégica con Airtable es el pilar central que permite un modelo de negocio innovador: al delegar las complejidades de la gestión de datos, la visualización analítica y la automatización para el lado del agente a una plataforma *low-code* robusta y visual, se logra un ahorro significativo en tiempo de desarrollo y costes, al tiempo que se ofrece una herramienta excepcionalmente intuitiva y valiosa para los profesionales inmobiliarios. Este enfoque híbrido no solo cumple con los requisitos técnicos del TFG, sino que también establece un precedente de valor para la eficiencia en el desarrollo de soluciones empresariales replicables.

## 3.4 HERRAMIENTAS DE DESARROLLO

La selección de las herramientas adecuadas es tan crucial como la elección de la arquitectura o la metodología de un proyecto de software. Para el desarrollo del **Trabajo de Fin de Grado InmoTable**, se han empleado diversas herramientas que han facilitado cada fase del ciclo de vida del software, desde el análisis y diseño hasta el desarrollo, la gestión y la documentación. Estas herramientas han contribuido a la eficiencia, la colaboración y la calidad del producto final.

A continuación, se detalla el conjunto de herramientas utilizadas, agrupadas por su función principal:

### 1. Análisis y Planificación del Proyecto

Fundamental para la fase de diseño de la interfaz de usuario y la experiencia del usuario (UI/UX) de la aplicación cliente.



**draw.io**



**PlantUML**



**ONLINE GANTT**

**Figura 3.14:** Logos de Draw.io PlantUML y Online Gantt Chart

- **Draw.io:** Una herramienta versátil para la creación de diagramas de flujo, diagramas de entidad-relación (ERD) para conceptualizar la base de datos de Airtable, diagramas de arquitectura de software o mockups de alto nivel. Facilitó la visualización de la estructura del sistema y sus componentes.
- **PlantUML:** Utilizado para la generación de diagramas a partir de texto simple. Aunque no se han proporcionado archivos específicos de PlantUML, su mención sugiere su uso para la creación de diagramas de secuencia, diagramas de clases, o diagramas de componentes que visualizan la interacción entre Angular, Laravel y Airtable, así como los flujos de negocio. Esto es crucial para comprender la arquitectura y las interacciones del sistema.
- **Online Gantt Chart:** Empleada para la planificación temporal y el seguimiento del progreso del proyecto. Permitió visualizar las fases de desarrollo (investigación, backend, frontend, configuración de Airtable, pruebas y documentación) con sus respectivas horas estimadas y reales, siendo fundamental para la gestión del tiempo y la adaptación a las desviaciones observadas [cronograma detallado en la sección de "Planificación Temporal"].

## 2. Mockups y Diseño

Estas herramientas fueron esenciales para la fase inicial de conceptualización, modelado y gestión del plan de trabajo.



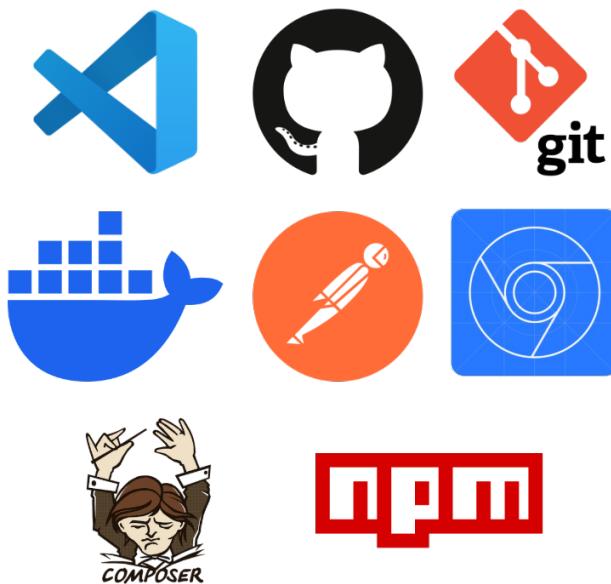
Figura 3.15: Logo de Figma

- **Figma:** Una potente herramienta de diseño de interfaces basada en la nube. Se utilizó para la creación de wireframes, mockups y prototipos de la interfaz de usuario del frontend de Angular. Permitió visualizar el diseño de las pantallas para el cliente (ej. listado de propiedades, detalle, formularios de citas y contacto), facilitando la iteración y la validación del diseño antes de la implementación [16].

## 3. Desarrollo del Software

Estas son las herramientas core para la escritura, depuración, control de versiones y despliegue del código.

### 3.4. HERRAMIENTAS DE DESARROLLO



**Figura 3.16:** Logos de Herramientas de Desarrollo del Software

- **Visual Studio Code (VS Code):** Un editor de código fuente ligero pero potente, compatible con múltiples lenguajes de programación (TypeScript, PHP, HTML, CSS/SCSS). Fue el entorno de desarrollo principal, ofreciendo características como autocompletado inteligente, depuración integrada, extensiones para Angular y Laravel, y soporte nativo para Git.
- **GitHub:** Una plataforma de alojamiento de repositorios de código basado en Git. Se utilizó para el control de versiones del código fuente del frontend (Angular) y backend (Laravel), permitiendo la gestión de cambios, la colaboración (si hubiera más de un desarrollador) y el respaldo del proyecto. Los archivos del proyecto se obtuvieron de un repositorio de GitHub [17].
- **Git (CLI):** La interfaz de línea de comandos de Git fue utilizada para todas las operaciones de control de versiones locales (ej. commit, push, pull, branch). Complementó a GitHub como la herramienta fundamental para el seguimiento y la gestión de la evolución del código fuente a lo largo del desarrollo.
- **Postman:** Una herramienta de colaboración para el desarrollo de APIs. Se utilizó para probar la API RESTful de Laravel de forma independiente, enviando solicitudes HTTP a los endpoints y verificando las respuestas, lo que facilitó la depuración de la comunicación entre el backend y Airtable, así como con el frontend.
- **Docker:** Una plataforma para el desarrollo, envío y ejecución de aplicaciones en contenedores. Se empleó para empaquetar la aplicación (frontend de Angular con Nginx y backend de Laravel con PHP-FPM) en entornos aislados y consistentes, facilitando la configuración del entorno de desarrollo y el futuro despliegue en producción.
- **Composer:** El gestor de dependencias para PHP, utilizado para instalar y gestionar todas las librerías y dependencias necesarias para el proyecto Laravel (composer.json) [18].

- **npm:** El gestor de paquetes de Node.js, indispensable para el proyecto Angular. Fue utilizado para instalar y gestionar todas las librerías, dependencias y scripts del frontend (definidos en package.json) [19].
- **Navegador Web y Herramientas de Desarrollo:** Cruciales para la depuración y optimización del frontend de Angular (Chrome DevTools). Permiten inspeccionar el DOM, CSS, JavaScript, peticiones de red y rendimiento.

#### 4. Documentación del Proyecto

Imprescindibles para la elaboración de la memoria y otros entregables del TFG.



Figura 3.17: Logos de LaTeX y Overleaf

- **LaTeX:** Un sistema de preparación de documentos de alta calidad, especialmente adecuado para documentos técnicos y científicos. Permite un control preciso sobre el formato y la estructura, lo que es ideal para la memoria de un TFG [20].
- **Overleaf:** Un editor LaTeX colaborativo en línea. Facilitó la escritura y compilación de la memoria del TFG en un entorno controlado, con la ventaja de la sincronización en la nube y la visualización en tiempo real, simplificando el proceso de redacción y revisión [21].

### 3.5 DESPLIEGUE

El despliegue de una aplicación es la fase crítica en la que el sistema de software se transfiere del entorno de desarrollo a un entorno de producción, haciéndolo accesible para los usuarios finales. Para InmoTable, la estrategia de despliegue se ha diseñado para complementar su arquitectura híbrida y modular, aprovechando plataformas de cloud hosting especializadas que garantizan eficiencia, escalabilidad y una gestión simplificada

#### 3.5.1 Estrategia de Despliegue Componentizada

Dada la clara separación de responsabilidades en la arquitectura de InmoTable (frontend en Angular, backend en Laravel, y Airtable como servicio externo), la estrategia de despliegue se ha basado en la descomposición del sistema en componentes desplegables de forma independiente. Esta aproximación permite seleccionar plataformas óptimas para cada capa, maximizando sus beneficios intrínsecos:

- **Despliegue del Frontend:** La aplicación web de cara al cliente (Angular) se beneficia de plataformas diseñadas para servir contenido estático y Single Page Applications (SPAs) de manera global y eficiente.

- **Despliegue del Backend:** La API RESTful (Laravel) requiere un entorno de servidor robusto capaz de procesar lógica de negocio, gestionar autenticación y orquestar comunicaciones con servicios externos.
- **Gestión de Datos (Airtable):** Airtable, al ser una solución Software-as-a-Service (SaaS) en la nube, gestiona su propio despliegue e infraestructura, lo que simplifica drásticamente la capa de persistencia para el desarrollador de InmoTable.

Las herramientas de despliegue elegidas se integran directamente con sistemas de control de versiones como Git, lo que facilita la implementación de prácticas de integración y despliegue continuo (CI/CD) para agilizar las actualizaciones.

### 3.5.2 Despliegue del Frontend (Aplicación Web Angular) en Vercel

El frontend de InmoTable, desarrollado con Angular, se ha configurado para su despliegue en **Vercel** [22], una plataforma de hosting serverless líder para aplicaciones frontend.

- **Justificación de la Elección:** Vercel fue seleccionado por su optimización específica para SPAs y frameworks de frontend como Angular. Ofrece una serie de ventajas clave para el despliegue de la capa de presentación:
  - **Despliegues Automatizados (CI/CD):** Se integra directamente con repositorios Git (ej. GitHub), automatizando el proceso de build y despliegue con cada git push a la rama de producción.
  - **Red de Entrega de Contenidos (CDN) Global:** Distribuye los activos estáticos de la aplicación a nivel mundial, garantizando tiempos de carga rápidos para los usuarios de InmoTable independientemente de su ubicación geográfica.
  - **Certificados SSL Automáticos:** Vercel configura y renueva automáticamente los certificados SSL, asegurando la comunicación segura (HTTPS) sin configuración manual.
  - **Escalabilidad Elástica:** La infraestructura sin servidor de Vercel permite que el frontend escale automáticamente para manejar picos de tráfico, manteniendo la disponibilidad y el rendimiento para los clientes.
  - **Configuración Simplificada:** Reduce la complejidad de la administración de servidores y la configuración de infraestructura para el frontend.
- **Proceso de Despliegue (Conceptual):** El proceso implica configurar el proyecto en Vercel para que identifique el directorio raíz del frontend (frontend/), y establecer el comando de construcción (ng build -configuration=production) y el directorio de salida (ej., dist/nombre-app-angular). Las variables de entorno, cruciales para que el frontend sepa dónde reside el backend (ej., la API \_ URL apuntando a Render/Railway), se configuran de forma segura directamente en el panel de Vercel.

### 3.5. DESPLIEGUE

---

#### 3.5.3 Despliegue del Backend (Laravel API) en Render

El backend de **InmoTable**, implementado como una API RESTful con Laravel, se ha diseñado para desplegarse como un servicio web gestionado. **Render** [23] ha sido seleccionado como la plataforma para este componente.

- **Justificación de la Elección:** Render es una plataforma versátil que soporta el despliegue de servicios web basados en PHP y Docker, lo cual se alinea perfectamente con la tecnología del backend de **InmoTable**. Sus principales beneficios incluyen:
  - **Soporte de Entornos de Ejecución:** Permite el despliegue de aplicaciones PHP utilizando un servidor de aplicaciones optimizado para producción, gestionando la infraestructura subyacente (ej., PHP-FPM y Nginx).
  - **Integración Continua:** Al igual que Vercel, Render se integra con repositorios Git, automatizando la reconstrucción y el despliegue del backend con cada push a la rama designada.
  - **Gestión Segura de Variables de Entorno:** Ofrece un mecanismo seguro para almacenar y acceder a variables sensibles como la clave de API de Airtable (AIRTABLE\_API\_KEY, AIRTABLE \_ BASE \_ ID), la APP \_ KEY de Laravel y los secretos de **Laravel Passport** necesarios para la autenticación API.
  - **Escalabilidad y Alta Disponibilidad:** Render proporciona capacidades para escalar el servicio web según la demanda y asegurar su alta disponibilidad en producción.
  - **Reducción de la Carga Operacional:** Simplifica la administración del servidor y la gestión de la infraestructura de backend, permitiendo a los desarrolladores centrarse en la lógica de negocio.
- **Proceso de Despliegue (Conceptual):** Implica conectar el repositorio Git que contiene el backend (backend/), configurar el directorio raíz, y especificar los comandos de construcción (ej., composer install -no-dev, php artisan config:cache, php artisan route:cache, php artisan view:cache para optimizar Laravel) y el comando de inicio del servidor web para PHP. Render asignará una URL pública al servicio, la cual será utilizada por el frontend.

#### 3.5.4 Interacción con Airtable en Producción

Es fundamental destacar que **Airtable** no forma parte de los despliegues en Vercel o Render. Al ser una plataforma SaaS, su gestión y disponibilidad son responsabilidad del propio proveedor. La comunicación entre el backend de Laravel desplegado y la base de datos de Airtable se realiza exclusivamente a través de la API de Airtable, utilizando las claves de API configuradas de forma segura como variables de entorno en Render. Esta delegación de la capa de datos es un componente clave de la eficiencia y el bajo coste operativo del proyecto.

En conclusión, la estrategia de despliegue multi-plataforma de **InmoTable** aprovecha las fortalezas de Vercel y Render para ofrecer una solución robusta y accesible. Esta aproximación no solo optimiza el rendimiento y la escalabilidad de cada componente, sino que también simplifica el ciclo de vida del desarrollo, facilitando el mantenimiento y las futuras actualizaciones del sistema.

### 3.6 CONCLUSIONES

El análisis y la selección de los modelos de desarrollo y las tecnologías constituyen un pilar fundamental sobre el cual se asienta la viabilidad y el éxito de InmoTable. Las decisiones tomadas en este capítulo han sido estratégicas para alinear la eficiencia del proceso de desarrollo con la propuesta de valor única del sistema.

En primer lugar, la adopción de la **metodología Scrum** ha demostrado ser la elección óptima. Su enfoque iterativo y flexible ha permitido una adaptación ágil a los requisitos cambiantes de un proyecto innovador como **InmoTable**, facilitando la integración continua de nuevas funcionalidades y la retroalimentación constante. Esto ha sido crucial para gestionar las fases del TFG de manera eficiente y controlada, asegurando la entrega de valor en cada ciclo.

En segundo lugar, la selección del **stack tecnológico híbrido** ha sido deliberada y fundamentada en la búsqueda de un equilibrio entre control, escalabilidad y eficiencia en costes:

- El **frontend desarrollado en Angular**, apoyado por **Angular Material y Bootstrap**, proporciona una aplicación web robusta, modular y visualmente atractiva para el cliente, garantizando una experiencia de usuario fluida y adaptable a cualquier dispositivo. Su ejecución sobre **Node.js** (junto con npm para la gestión de paquetes) completa un entorno de desarrollo frontend sólido.
- El **backend implementado con Laravel** (ejecutado con PHP-FPM y gestionado con Composer) actúa como una API RESTful segura y eficiente. Es la capa responsable de la lógica de negocio, la autenticación mediante Laravel Passport y la compleja orquestación de datos con Airtable a través de su `AirtableService`.
- La integración estratégica de **Airtable como base de datos central y plataforma de gestión low-code** es el elemento diferenciador que ha permitido un **ahorro drástico en los costes y tiempos de desarrollo** de interfaces de administración complejas. Al delegar la gestión de propiedades, clientes, citas, análisis de negocio y automatizaciones a las interfaces configurables de Airtable, se empodera a los agentes y administradores no técnicos, y se facilita enormemente la portabilidad y replicabilidad del modelo de negocio para futuras inmobiliarias.
- Finalmente, el uso de **Docker** para la orquestación de los entornos de desarrollo y despliegue garantiza la consistencia y simplifica la puesta en marcha de la aplicación.

En conjunto, las elecciones metodológicas y tecnológicas han permitido desarrollar un sistema **eficiente, escalable, costo-efectivo** y **altamente adaptable**, cumpliendo con los objetivos del TFG y sentando las bases para una solución de gestión inmobiliaria con una clara ventaja competitiva en el mercado.



**Figura 3.18:** Resumen de Modelos de Desarrollo y Tecnologías

En la Figura 3.18, se presenta un resumen visual de todas las tecnologías y herramientas clave seleccionadas para el desarrollo del proyecto **InmoTable**.

---

## 4 INMOTABLE

---

En este capítulo se describe en profundidad el diseño y la implementación del sistema, abarcando su arquitectura, el modelo de datos y las interfaces de usuario para clientes (web) y agentes (Airtable).

### 4.1 DESCRIPCIÓN DE LA APLICACIÓN

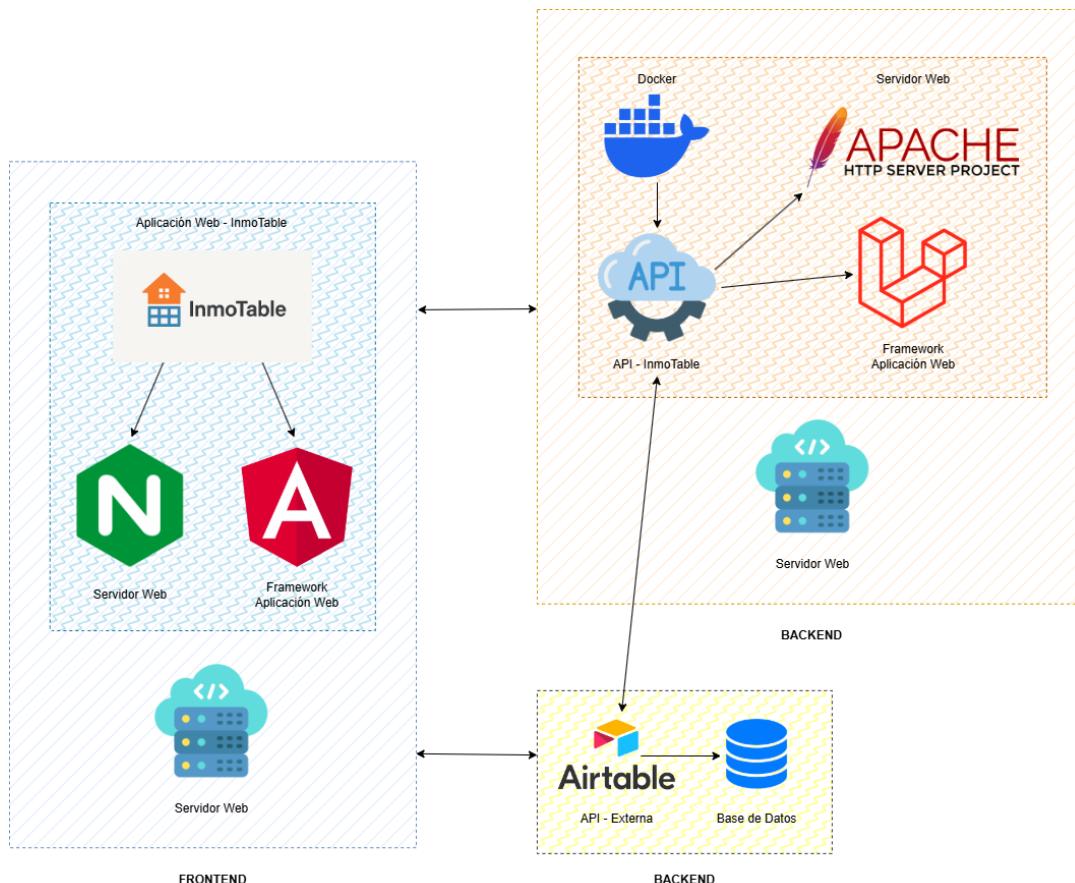
El proyecto **InmoTable** es una aplicación web de gestión inmobiliaria que implementa una arquitectura híbrida de tres capas para optimizar las operaciones del sector. Consiste en un frontend interactivo desarrollado con Angular para los clientes finales, que les permite buscar propiedades, visualizar detalles y solicitar citas. Este frontend se comunica con un backend robusto construido en Laravel, que actúa como una API segura y el intermediario principal para la gestión de la lógica de negocio y la persistencia de datos.

La singularidad de **InmoTable** radica en la utilización de Airtable como la base de datos central y la plataforma principal de gestión interna para los agentes inmobiliarios y administradores. Airtable no solo almacena los datos de propiedades, clientes y citas, sino que también provee interfaces visuales *low-code* (formularios, vistas personalizadas, dashboards analíticos) y automatizaciones que permiten a los agentes gestionar eficientemente todas sus operaciones diarias sin necesidad de desarrollo personalizado, ofreciendo una solución rentable y altamente replicable para múltiples inmobiliarias.

### 4.2 DISEÑO DE LA ARQUITECTURA

El diseño arquitectónico del sistema **InmoTable** es una pieza central que define la estructura y el comportamiento de sus componentes. Se ha optado por una arquitectura de tres capas (o N-tier) distribuidas y un enfoque híbrido, buscando un equilibrio óptimo entre la experiencia de usuario para el cliente final, la robustez de la lógica de negocio y la eficiencia en la gestión interna de la inmobiliaria. La Figura 4.1 ilustra esta estructura general.

## 4.2. DISEÑO DE LA ARQUITECTURA



**Figura 4.1:** Estructura General del Proyecto

Esta arquitectura se descompone en las siguientes capas y componentes clave:

### 1. Capa de Presentación (Frontend: Aplicación Web para Clientes)

Esta capa representa la interfaz directa con los clientes finales (Visitantes Web y Clientes Registrados). Su objetivo es proporcionar una experiencia de usuario moderna, intuitiva y fluida.

- **Tecnología Principal:** Angular (desarrollado con TypeScript y ejecutado sobre Node.js) [24].
- **Librerías de UI/UX:** Se utilizan Angular Material y Bootstrap para estandarizar y acelerar el desarrollo de componentes visuales, garantizando un diseño responsive y atractivo para diversos dispositivos [25] [26].
- **Funcionalidades:** Permite a los Visitantes Web buscar, filtrar y visualizar listados y detalles de propiedades, así como enviar mensajes de contacto. Los Clientes Registrados pueden además autenticarse y solicitar citas para propiedades.
- **Comunicación:** Interactúa con el backend a través de peticiones HTTP a una API RESTful.

## 2. Capa de Lógica de Negocio/Aplicación (Backend: API RESTful en Laravel)

Esta capa es el cerebro del sistema. Actúa como el intermediario seguro y la central de orquestación entre la aplicación web del cliente y la base de datos central de Airtable.

- **Tecnología Principal:** Laravel (PHP) [27].
- **Entorno de Ejecución:** Se apoya en **PHP-FPM** para el procesamiento de las solicitudes PHP, servido por **Nginx**.
- **Funcionalidades:**
  - **Gestión de Autenticación y Autorización:** Utiliza Laravel **Passport** para un sistema robusto de autenticación API basado en tokens (OAuth2), gestionando el acceso de clientes, agentes y administradores.
  - **Interacción con la Capa de Datos:** Contiene el `AirtableService.php`, que es el módulo encargado de traducir las solicitudes del frontend en llamadas a la API de Airtable (operaciones CRUD sobre propiedades, citas, clientes, etc.).
  - **Validación y Lógica de Negocio:** Aplica reglas de negocio y validaciones de datos antes de interactuar con la capa de persistencia o devolver respuestas al frontend.

## 3. Capa de Datos y Gestión Interna (Airtable: Plataforma para Agentes y Administradores)

Esta capa es el corazón de la gestión inmobiliaria para los usuarios internos, diferenciándose por su naturaleza *low-code* y su rica interfaz para la administración del negocio.

- **Tecnología Principal:** Airtable [28] [29] [30].
- **Funcionalidades Clave:**
  - **Almacenamiento de Datos:** Contiene todas las tablas estructuradas del negocio inmobiliario: Propiedades, Agentes, Clientes, Citas, Empresas, Usuarios, Contacto.
  - **Interfaces de Gestión:** Provee una interfaz visual completa para **Agentes Inmobiliarios y Administradores**, que incluye:
    - **Vistas Personalizadas:** Para visualizar datos en distintos formatos (calendarios, galerías, mapas).
    - **Formularios de Entrada de Datos:** Para el CRUD de entidades (ej. "Nueva Propiedad", "Nueva Cita").
    - **Dashboards Analíticos (Airtable Interfaces):** Herramientas de Business Intelligence integradas para el seguimiento del negocio (ej. "Análisis de Stock", "Media de Precios", "Tiempo Medio de Venta", etc.).
  - **Automatizaciones:** Gestión de flujos de trabajo automatizados para reducir tareas manuales (ej. envío de notificaciones por email).
  - **Comunicación:** Se comunica con el Backend (Laravel) a través de la API de Airtable.

#### 4. Orquestación del Entorno (Docker)

- **Tecnología: Docker y Docker Compose** [31].
- **Rol:** Encapsula las capas de Frontend (servida por Nginx en un contenedor) y Backend (Laravel y PHP-FPM en otro contenedor) en entornos aislados y portables. Esto asegura la consistencia entre el desarrollo y el despliegue, y facilita la replicabilidad de la aplicación.

#### Flujo General de Interacción del Sistema:

1. Los **Clients** interactúan directamente con la **Aplicación Web (Frontend Angular)**.
2. La **Aplicación Web (Frontend Angular)** realiza peticiones a la **API RESTful (Backend Laravel)** para obtener y enviar datos.
3. El **Backend (Laravel API)** procesa estas peticiones, aplica la lógica de negocio y se comunica bidireccionalmente con **Airtable (Capa de Datos y Gestión Interna)** a través de su API para leer y escribir información.
4. Los **Agentes Inmobiliarios y Administradores** interactúan directamente con **Airtable**, utilizando sus interfaces para gestionar propiedades, citas, clientes y acceder a los paneles de análisis y automatizaciones.

### 4.3 MODELO DE DATOS

El diseño del modelo de datos es un pilar fundamental en la arquitectura de cualquier sistema de software, ya que define cómo se organiza, se almacena y se relaciona la información crucial del negocio. Para **InmoTable**, el modelo de datos se ha concebido y materializado en Airtable, aprovechando sus capacidades de base de datos relacional con una interfaz amigable tipo hoja de cálculo. Este enfoque permite una gestión de datos eficiente y flexible por parte de los agentes inmobiliarios, al tiempo que soporta la lógica de la aplicación web.

La Figura 4.2 presenta el diagrama del modelo de datos de **InmoTable**, donde se visualizan las principales entidades del sistema, sus atributos y las relaciones que las conectan.

### 4.3. MODELO DE DATOS

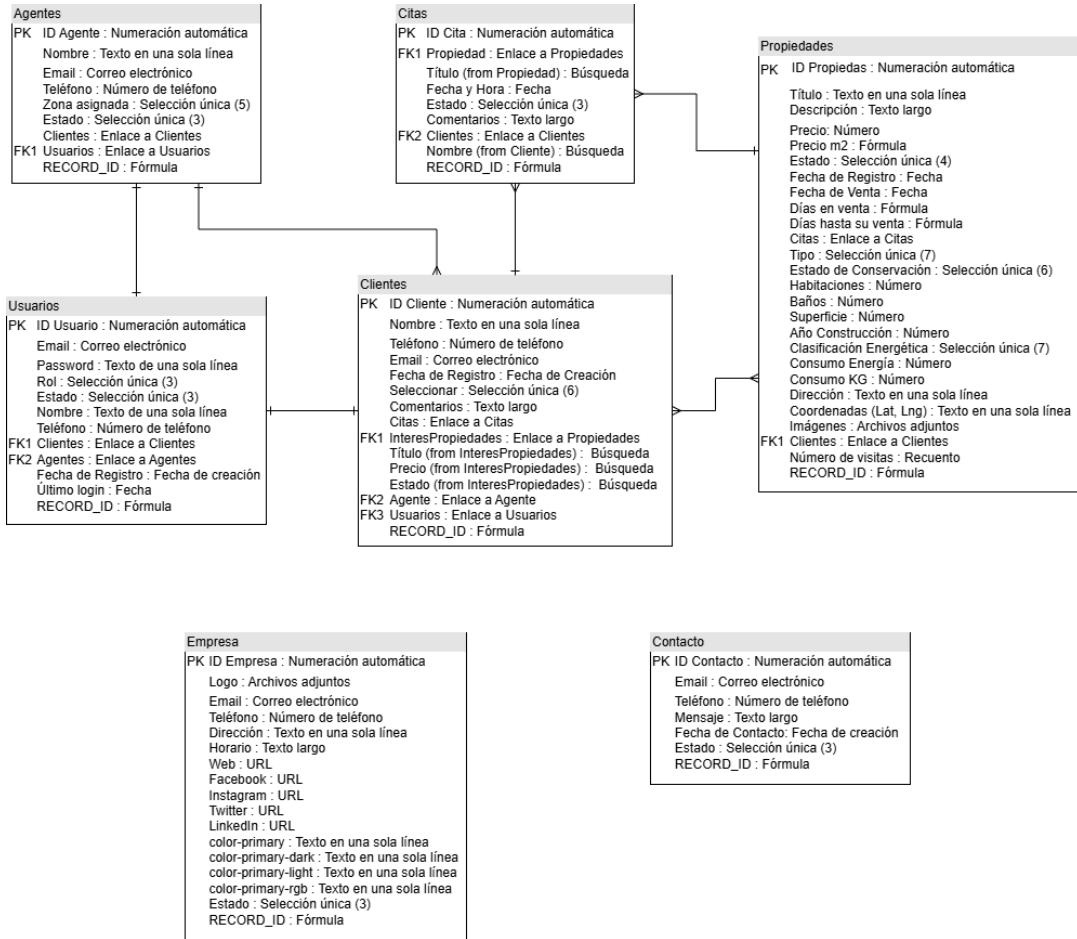


Figura 4.2: Diagrama del modelo de datos

El modelo de datos se compone de las siguientes entidades principales, cada una representando un componente clave de la gestión inmobiliaria:

#### 1. Entidades del Modelo de Datos

- **Agente**: Almacena la información de los agentes inmobiliarios.
- **Cita**: Representa las citas o reuniones programadas (ej., visitas a propiedades).
- **Cliente**: Contiene la información de los clientes interesados en los servicios de la inmobiliaria.
- **Empresa**: Almacena los datos de la inmobiliaria o empresa además de contenido dinámico para la aplicación web (ej., colores principales de la web).
- **Propiedad**: El catálogo central de inmuebles gestionados por la inmobiliaria.
- **Usuario**: Centraliza las credenciales y roles de acceso al sistema para autenticación.
- **Contacto**: Registra los mensajes y consultas recibidas a través del formulario de contacto de la aplicación web.

## 2. Consideraciones del Modelo de Datos en Airtable

La elección de Airtable como plataforma de datos introduce características específicas que refuerzan el modelo:

- **Interfaz Visual y Flexible:** La naturaleza de spreadsheet de Airtable permite a los agentes visualizar y manipular los datos de manera intuitiva, a la vez que se mantiene la robustez de las relaciones subyacentes. La gestión directa de tablas, formularios e interfaces por usuarios no técnicos es una de sus principales ventajas operacionales.
- **Campos Derivados y Analíticos:** Airtable facilita la creación de campos de fórmula y rollup (ej., Días en el Mercado, recuentos de Citas o Clientes Interesados), que no se almacenan directamente, sino que se calculan a partir de los datos existentes y las relaciones, proporcionando las métricas necesarias para los dashboards analíticos de los agentes.
- **Manejo de Archivos Multimedia:** El campo Fotos en la tabla Propiedad utiliza la capacidad de Airtable para almacenar adjuntos, simplificando la gestión de imágenes de propiedades.

## 4.4 DISEÑO E IMPLEMENTACIÓN

### 4.4.1 Prototipado de la interfaz

El prototipado de la interfaz de usuario (UI) es un paso esencial en el proceso de diseño centrado en el usuario, permitiendo visualizar y validar la experiencia del usuario (UX) antes de invertir recursos significativos en el desarrollo. Para **InmoTable**, esta fase se llevó a cabo utilizando Figma, una herramienta colaborativa de diseño que facilitó la creación de wireframes, mockups de alta fidelidad y prototipos interactivos. El objetivo principal fue obtener una representación visual clara de cómo interactuarían los clientes con la aplicación web, permitiendo recolectar retroalimentación temprana y realizar ajustes en el diseño.

A continuación, se presentan los mockups clave diseñados, seguidos de una justificación de cómo la implementación final pudo diferir.

#### 4.4. DISEÑO E IMPLEMENTACIÓN

##### 4.4.1.1 Mockup de la Página de Inicio (Landing Page)

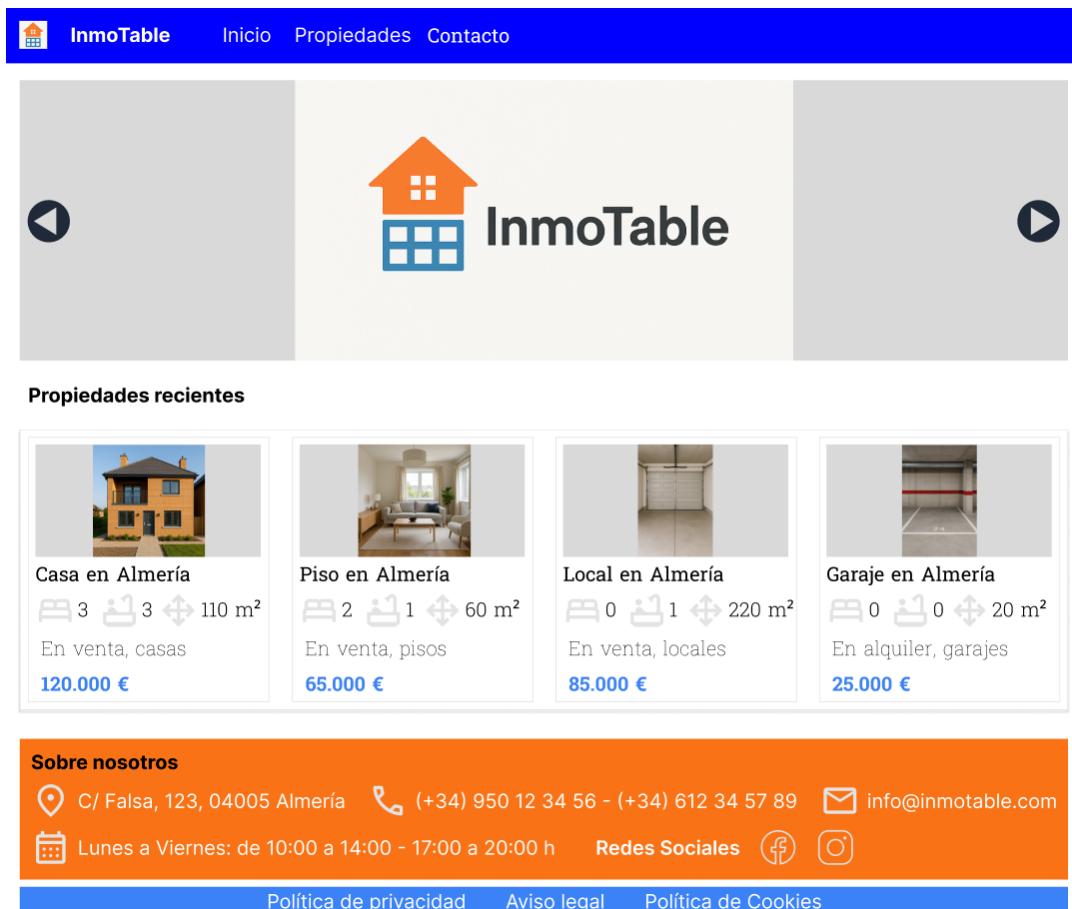
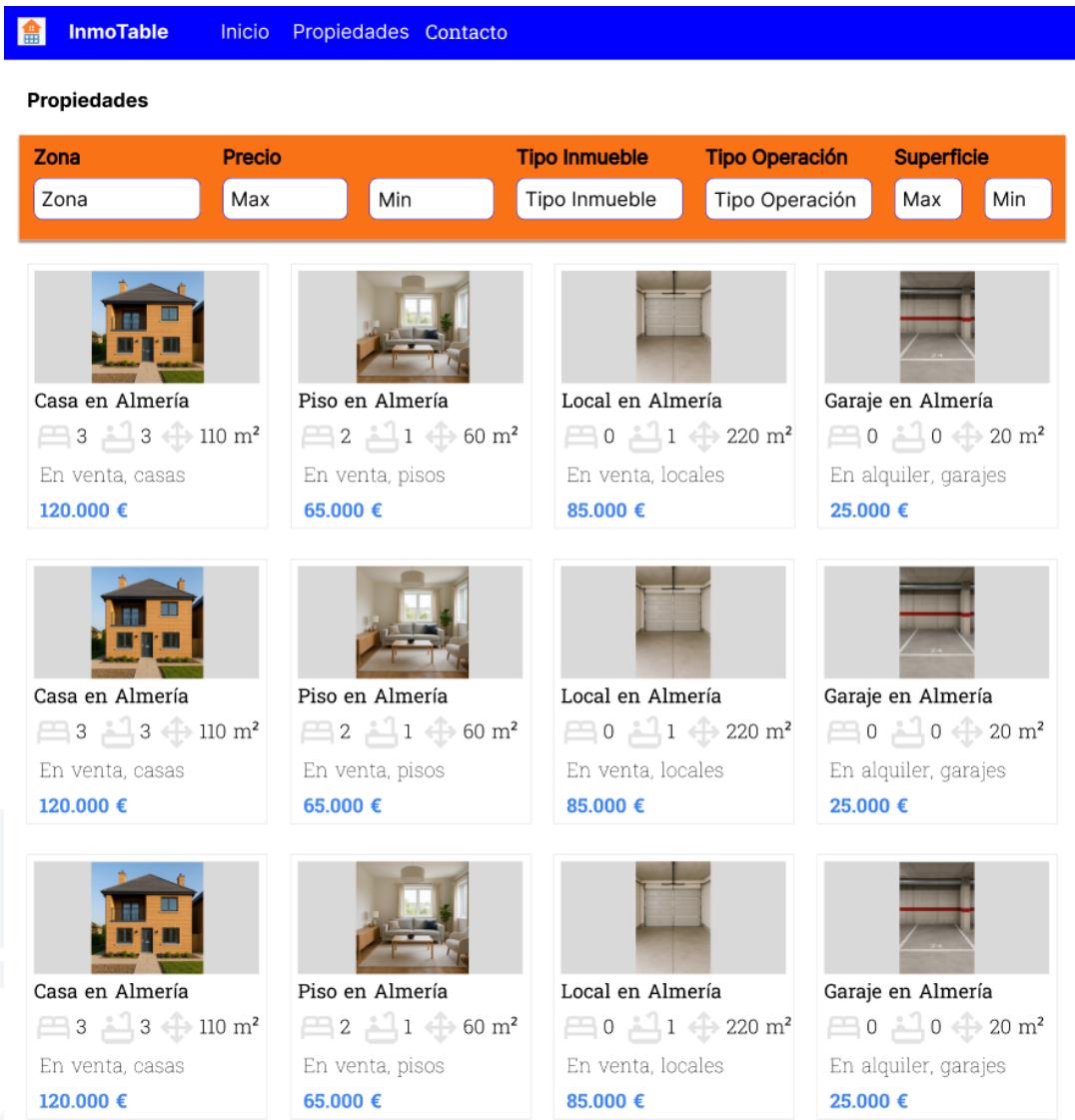


Figura 4.3: Mockup de la Página de Inicio (Landing Page)

La figura 4.3 representa la primera impresión del usuario al acceder a la aplicación. Típicamente incluiría una sección principal (hero section) con un mensaje de bienvenida o un carrusel de propiedades destacadas, una barra de navegación clara, y posiblemente secciones introductorias a las funcionalidades clave como la búsqueda de propiedades o el contacto.

#### 4.4.1.2 Mockup de la Página de Listado de Propiedades



The mockup displays a property listing page with the following structure:

- Header:** InmoTable, Inicio, Propiedades, Contacto.
- Section Title:** Propiedades.
- Filter Bar:** Zona, Precio, Tipo Inmueble, Tipo Operación, Superficie. Each category has dropdown menus for Max, Min values.
- Property Grid:** Arranged in a 3x4 grid. Each card includes an image, title, filters, and price.

Zona	Precio	Tipo Inmueble	Tipo Operación	Superficie
Zona	Max	Min	Tipo Inmueble	
			Tipo Operación	Max
				Min



Casa en Almería

3 habitaciones, 3 baños, 110 m<sup>2</sup>

En venta, casas

**120.000 €**



Piso en Almería

2 habitaciones, 1 baño, 60 m<sup>2</sup>

En venta, pisos

**65.000 €**



Local en Almería

0 habitaciones, 1 baño, 220 m<sup>2</sup>

En venta, locales

**85.000 €**



Garaje en Almería

0 habitaciones, 0 baños, 20 m<sup>2</sup>

En alquiler, garajes

**25.000 €**



Casa en Almería

3 habitaciones, 3 baños, 110 m<sup>2</sup>

En venta, casas

**120.000 €**



Piso en Almería

2 habitaciones, 1 baño, 60 m<sup>2</sup>

En venta, pisos

**65.000 €**



Local en Almería

0 habitaciones, 1 baño, 220 m<sup>2</sup>

En venta, locales

**85.000 €**



Garaje en Almería

0 habitaciones, 0 baños, 20 m<sup>2</sup>

En alquiler, garajes

**25.000 €**



Casa en Almería

3 habitaciones, 3 baños, 110 m<sup>2</sup>

En venta, casas

**120.000 €**



Piso en Almería

2 habitaciones, 1 baño, 60 m<sup>2</sup>

En venta, pisos

**65.000 €**



Local en Almería

0 habitaciones, 1 baño, 220 m<sup>2</sup>

En venta, locales

**85.000 €**



Garaje en Almería

0 habitaciones, 0 baños, 20 m<sup>2</sup>

En alquiler, garajes

**25.000 €**

**Figura 4.4:** Mockup de la Página de Listado de Propiedades

La figura 4.4 muestra cómo se visualizarían las propiedades disponibles en un formato de lista o cuadricula. Incluiría elementos de filtrado y búsqueda en la parte superior o lateral, y tarjetas de propiedad con información esencial como imagen, precio, ubicación y número de habitaciones. Es la interfaz principal para la exploración del inventario.

## 4.4. DISEÑO E IMPLEMENTACIÓN

### 4.4.1.3 Mockup de la Página de Detalle de Propiedad

InmoTable
Inicio
Propiedades
Contacto

**Propiedad de ejemplo**

65.000 € 2 1 60 m<sup>2</sup> Pedir cita

**Galería**



◀
▶

**Características**

Tipo:	Piso	Clasificación energética:																								
Estado:	Muy bien	<table border="1" style="border-collapse: collapse; width: 100px; height: 100px;"> <thead> <tr> <th style="width: 30px;"></th> <th style="width: 40px; text-align: center;">Consumo de Energía kWh/m<sup>2</sup> Año</th> <th style="width: 30px; text-align: center;">Consumo Kg CO<sub>2</sub>/m<sup>2</sup> Año</th> </tr> </thead> <tbody> <tr> <td style="color: green;">A</td> <td></td> <td></td> </tr> <tr> <td style="color: green;">B</td> <td></td> <td></td> </tr> <tr> <td style="color: green;">C</td> <td></td> <td></td> </tr> <tr> <td style="color: yellow;">D</td> <td></td> <td></td> </tr> <tr> <td style="color: orange;">E</td> <td></td> <td></td> </tr> <tr> <td style="color: red;">F</td> <td></td> <td></td> </tr> <tr> <td style="color: red;">G</td> <td></td> <td></td> </tr> </tbody> </table>		Consumo de Energía kWh/m <sup>2</sup> Año	Consumo Kg CO <sub>2</sub> /m <sup>2</sup> Año	A			B			C			D			E			F			G		
	Consumo de Energía kWh/m <sup>2</sup> Año		Consumo Kg CO <sub>2</sub> /m <sup>2</sup> Año																							
A																										
B																										
C																										
D																										
E																										
F																										
G																										
Habitaciones:	2																									
Baños:	1																									
Superficie:	60 m <sup>2</sup>																									
Año construcción:	2001																									
Precio m <sup>2</sup> :	1.083,33 €																									
Precio:	65.000 €																									

**Descripción**

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

**Visitas al inmueble**



Visitas a Propiedad - Marzo 2025 [Ejemplo Altable]

Fecha	Número de Visitas
2025-03-01	3
2025-03-02	1
2025-03-03	4
2025-03-04	1
2025-03-05	2
2025-03-06	3
2025-03-07	2
2025-03-08	1
2025-03-09	4
2025-03-10	1
2025-03-11	3
2025-03-12	2
2025-03-13	1
2025-03-14	2
2025-03-15	1
2025-03-16	2
2025-03-17	1
2025-03-18	2
2025-03-19	1
2025-03-20	5
2025-03-21	1
2025-03-22	4
2025-03-23	1
2025-03-24	3
2025-03-25	1
2025-03-26	2
2025-03-27	1
2025-03-28	2
2025-03-29	1
2025-03-30	2
2025-03-31	1

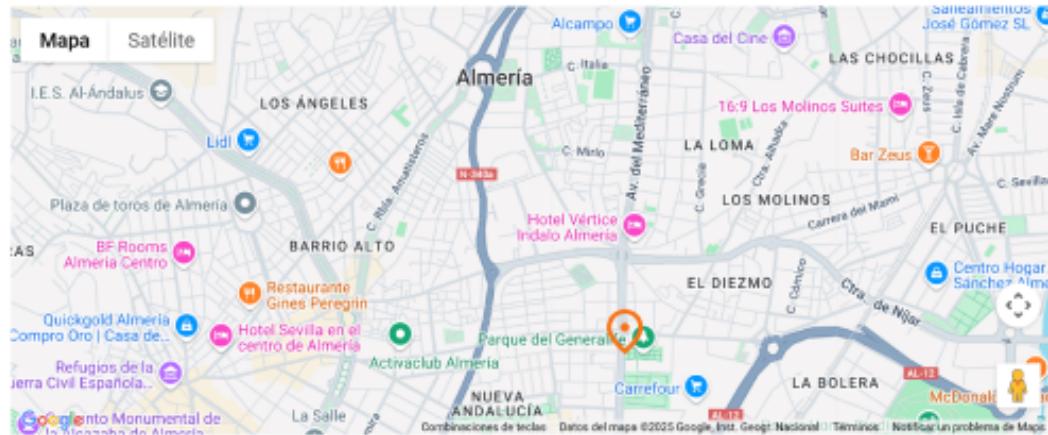
**Figura 4.5:** Mockup de la Página de Detalle de Propiedad

Grado en Informática  
2024/2025

62

#### 4.4. DISEÑO E IMPLEMENTACIÓN

Mapa



**Figura 4.6:** Mockup del Mapa de la Página de Detalle de Propiedad

Las figuras 4.5 y 4.6 detallan la vista individual de una propiedad. Presentaría una galería de imágenes o un visor principal, una descripción textual completa, características detalladas (superficie, baños, calificación energética), y secciones para solicitar más información o programar una visita.

##### 4.4.1.4 Mockup de Formulario de Solicitud de Cita

**Figura 4.7:** Mockup de Formulario de Solicitud de Cita

La figura 4.7 ilustra el formulario diseñado para que los clientes soliciten una cita, presumiblemente para visitar una propiedad. Típicamente incluiría campos para la información de contacto del cliente (nombre, email), la propiedad de interés y los detalles deseados de la cita (fecha, hora, tipo de cita). Es la interfaz principal para formalizar el interés del cliente en un inmueble y ser gestionado por los agentes.

#### 4.4.1.5 Mockup de Formulario de Contacto



The mockup shows a blue header bar with the InmoTable logo and navigation links: Inicio, Propiedades, and Contacto. Below the header, the word 'Contacto' is displayed. There are three input fields: 'Nombre' (Name), 'Teléfono' (Phone), and 'Email'. Below these is a large text area labeled 'Comentario' (Comment) for messages.

**Figura 4.8:** Mockup de Formulario de Contacto

La figura 4.8 representa un formulario general para que los usuarios (Visitantes Web o Clientes Registrados) puedan enviar mensajes, consultas o comentarios a la inmobiliaria. Contendría campos para el nombre, email y un área de texto para el mensaje.

#### 4.4.1.6 Justificación de las Desviaciones entre Prototipo e Implementación Real

Es una práctica común y esperable que existan diferencias entre los mockups iniciales y la interfaz final implementada en la aplicación web. Estas desviaciones no representan fallas en el diseño, sino una adaptación necesaria y optimización basada en factores que emergen durante el ciclo de desarrollo ágil y la codificación real. Las principales razones para estas modificaciones en **InmoTable** incluyen:

- **Restricciones Técnicas y Optimización de la Implementación:** Ciertos elementos de diseño, aunque atractivos en Figma, pueden ser complejos, demandar mucho tiempo o generar problemas de rendimiento al ser implementados en Angular. Las decisiones de implementación final priorizaron la eficiencia, la carga rápida y la estabilidad.
- **Aprovechamiento de Librerías de Componentes (Angular Material y Bootstrap):** Al utilizar frameworks de UI como Angular Material y Bootstrap, la implementación se adhiere a sus componentes prediseñados y estándares de estilo para mantener la consistencia y la velocidad de desarrollo. Esto puede resultar en variaciones visuales respecto a un mockup totalmente personalizado de Figma que no considera las limitaciones o convenciones de estas librerías.
- **Retroalimentación y Pruebas de Usabilidad:** Durante las pruebas internas o simulaciones de interacción con usuarios, se pudieron identificar puntos de fricción o mejoras en la usabilidad que llevaron a ajustes en el layout, la disposición de elementos o los flujos de navegación para hacer la aplicación más intuitiva y fácil de usar para el cliente final.

- **Evolución Natural del Proyecto:** El desarrollo ágil, bajo la metodología Scrum, permite que los requisitos y el diseño evolucionen a medida que se gana comprensión sobre el problema y la solución, llevando a optimizaciones que no eran evidentes en la fase inicial de prototipado.

En conclusión, los mockups sirvieron como una guía visual fundamental, pero la implementación real priorizó la viabilidad técnica, la eficiencia del desarrollo y una experiencia de usuario optimizada, adaptándose a las realidades de la codificación y la integración con las herramientas seleccionadas.

#### 4.4.2 Diseño de la interfaz

La fase de diseño de la interfaz ha sido fundamental para materializar la arquitectura y los requisitos de **InmoTable** en una experiencia de usuario coherente y efectiva. Dada la naturaleza híbrida del sistema, este diseño se ha bifurcado para satisfacer las necesidades específicas de dos perfiles de usuario principales: los clientes finales que interactúan con la aplicación web, y los agentes inmobiliarios y administradores que gestionan las operaciones internas a través de Airtable.

El objetivo ha sido crear una aplicación web intuitiva y visualmente atractiva para el cliente, utilizando Angular Material y Bootstrap para una interfaz moderna y responsive. Paralelamente, se han configurado potentes interfaces de gestión en Airtable que empoderan a los usuarios internos con herramientas visuales, formularios dinámicos y paneles de análisis sin necesidad de programación.

A continuación, se detalla el diseño de la interfaz de usuario, organizado por los módulos clave del sistema, mostrando cómo cada funcionalidad se aborda en ambas plataformas.

##### 4.4.2.1 Módulo: Componentes Globales de la Interfaz del Cliente

Este módulo describe los elementos de la interfaz de usuario que son omnipresentes o que establecen la primera interacción con la aplicación web para el cliente, proporcionando una base visual y funcional consistente en toda la experiencia de usuario.

###### 1. Página de Inicio (Home Page)

La página de inicio (`home.component`) es el punto de entrada principal para los **Visitantes Web y Clientes Registrados**, diseñada para causar una primera impresión impactante y dirigir al usuario hacia las funcionalidades clave del sistema. Esá inspirada en el mockup de la Figura 4.3) y corresponde al requisito funcional que busca ofrecer una experiencia de búsqueda atractiva y funcional.

#### 4.4. DISEÑO E IMPLEMENTACIÓN

# Encuentra tu hogar perfecto en InmoTable

La mayor selección de propiedades en Almería. Más de 200 propiedades disponibles.

  
**200+**  
 Propiedades

  
**150+**  
 Citas Realizadas

  
**98%**  
 Clientes Satisfechos

  
**15+**  
 Años de Experiencia

## Propiedades Destacadas

Descubre las mejores oportunidades inmobiliarias

**Garaje en Calle Murcia**

**242.043,8 €** 1323 €/m<sup>2</sup>

📍 Calle Murcia

	1 Habitaciones
	1 Baños
	183 m <sup>2</sup>

Construido en 2009 Visitas 1

**Ático en Calle Regocijos**

**139.248,4 €** 827 €/m<sup>2</sup>

📍 Calle Regocijos

	5 Habitaciones
	1 Baños
	168,4 m <sup>2</sup>

Construido en 2009 Visitas 1

**Terreno en Calle Granada**

**185.733,65 €** 1039 €/m<sup>2</sup>

📍 Calle Granada

	1 Habitaciones
	2 Baños
	178,7 m <sup>2</sup>

Construido en 2009 Visitas 2

**Piso en Calle Regocijos**

**327.856,76 €** 3380 €/m<sup>2</sup>

📍 Calle Regocijos

	1 Habitaciones
	2 Baños
	97 m <sup>2</sup>

Construido en 2009 Visitas 1

**Terreno en Calle Jovellanos**

**269.839,33 €** 1168 €/m<sup>2</sup>

📍 Calle Jovellanos

	1 Habitaciones
	1 Baños
	231,1 m <sup>2</sup>

Construido en 2009 Visitas 1

**Casa en Calle Jovellanos**

**499.989,48 €** 6173 €/m<sup>2</sup>

📍 Calle Jovellanos

	5 Habitaciones
	1 Baños
	81 m <sup>2</sup>

Construido en 2009 Visitas 1

[Ver Todas las Propiedades](#)

**Figura 4.9:** Página de Inicio

Grado en Informática  
2024/2025

66

#### 4.4. DISEÑO E IMPLEMENTACIÓN

---

Tal como se aprecia en la Figura 4.9 (Imagen inicio.jpg), la página de inicio se caracteriza por:

- **Sección principal (hero section):** Visualmente atractiva, esta sección contiene un mensaje principal, mencionando directamente el nombre de la inmobiliaria ("InmoTable" en este contexto) y un llamado a la acción claro.
- **Buscador de propiedades integrado:** Directamente en la hero section se ubica un buscador de propiedades con filtros predefinidos (ej., "Tipo de Propiedad", "Zona", "Precio"). Al utilizar este buscador, el usuario será redirigido dinámicamente al Listado de Propiedades, aplicando los filtros seleccionados, lo que agiliza el proceso de búsqueda.
- **Contenido Destacado:** Presenta un resumen de propiedades destacadas o categorías populares, invitando al usuario a explorar el inventario sin necesidad de navegar a una página de listado específica de inmediato.
- **Elementos de Navegación Integrados:** Incluye la barra de navegación en la parte superior y el pie de página en la inferior, manteniendo la coherencia de la interfaz.

Es importante destacar que la configuración general de colores de la aplicación son configurados dinámicamente. Sus valores se obtienen de campos específicos definidos en la tabla Empresa de Airtable, lo que facilita enormemente la personalización visual y la portabilidad de la aplicación para diferentes inmobiliarias con un mínimo esfuerzo de configuración en el código.

#### 4.4. DISEÑO E IMPLEMENTACIÓN

# Encuentra tu hogar perfecto en InmoJuandi

La mayor selección de propiedades en Almería. Más de 0 propiedades disponibles.

Tipo  
Cualquier
Zona  
Cualquier
Precio  
Cualquier precio



**201+**  
Propiedades



**150+**  
Citas Realizadas



**98%**  
Clientes Satisfechos



**15+**  
Años de Experiencia

## Propiedades Destacadas

Descubre las mejores oportunidades inmobiliarias



**Garaje en Calle Murcia**

**242.043,8 €** 1323 €/m<sup>2</sup>

Calle Murcia

1	Habitaciones
1	Baños
183	m <sup>2</sup>

Construido en 2009 | Visitas 1



**Ático en Calle Regocijos**

**139.248,4 €** 827 €/m<sup>2</sup>

Calle Regocijos

5	Habitaciones
1	Baños
168,4	m <sup>2</sup>

Construido en 2009 | Visitas 2



**Terreno en Calle Granada**

**185.733,65 €** 1039 €/m<sup>2</sup>

Calle Granada

1	Habitaciones
2	Baños
178,7	m <sup>2</sup>

Construido en 2009 | Visitas 2



**Piso en Calle Regocijos**

**327.856,76 €** 3380 €/m<sup>2</sup>

Calle Regocijos

1	Habitaciones
2	Baños
97	m <sup>2</sup>

Construido en 2009 | Visitas 1



**Terreno en Calle Jovellanos**

**269.839,33 €** 1168 €/m<sup>2</sup>

Calle Jovellanos

1	Habitaciones
1	Baños
231,1	m <sup>2</sup>

Construido en 2009 | Visitas 1



**Casa en Calle Jovellanos**

**499.989,48 €** 5173 €/m<sup>2</sup>

Calle Jovellanos

5	Habitaciones
1	Baños
81	m <sup>2</sup>

Construido en 2009 | Visitas 1

[Ver Todas las Propiedades](#)

**Figura 4.10:** Página de Inicio dinámica

Grado en Informática  
2024/2025

68

#### 4.4. DISEÑO E IMPLEMENTACIÓN

En la Figura 4.10 se visualiza página de inicio de la otra empresa configurada, donde se observa el cambio de título y de colores.

##### 2. Barra de Navegación (Nav / Header)

La barra de navegación (navbar.component) es un componente persistente que se mantiene en la parte superior de todas las páginas de la aplicación web, proporcionando acceso constante a las secciones principales del sistema. Visible en la Figura 4.11).



**Figura 4.11:** Barra de navegación

Incluye el logotipo de la inmobiliaria para branding, y un conjunto de enlaces claros que dirigen a los usuarios a las secciones fundamentales: "Propiedades", "Contacto", y opciones de autenticación como "Login" o "Registro". En dispositivos móviles, la navegación se adapta a un menú tipo "hamburguesa" para optimizar el espacio. Como se menciona anteriormente la configuración de colores globales de la aplicación en este componente además el logo y el nombre de la empresa son configurables dinámicamente desde la tabla Empresa de Airtable, lo que garantiza su fácil actualización y coherencia con la imagen de la inmobiliaria.

Al igual que la Barra de Navegación, la información de contacto y otros datos que se muestran en el pie de página son configurables dinámicamente desde la tabla Empresa de Airtable, lo que garantiza su fácil actualización y coherencia con la imagen de la inmobiliaria.

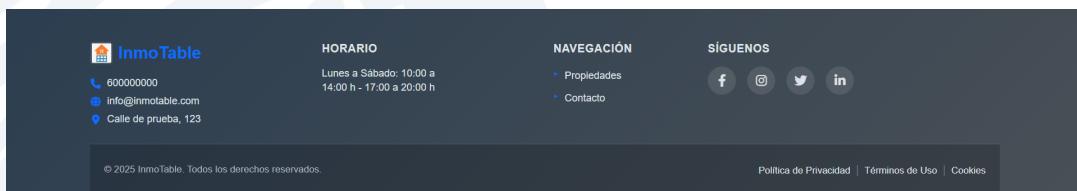


**Figura 4.12:** Barra de navegación dinámica

En la Figura 4.12 se visualiza barra de navegación de la otra empresa configurada, donde se observa el cambio de logo, título y de colores.

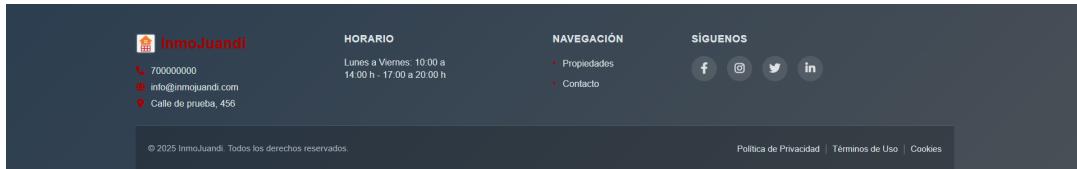
##### 3. Pie de Página (Footer)

El pie de página (footer.component) es otro componente global que se encuentra en la parte inferior de todas las páginas de la aplicación web. Visible en la Figura 4.13.



**Figura 4.13:** Pie de página

Contiene información esencial de la inmobiliaria (teléfono, dirección, email de contacto), enlaces a políticas legales (privacidad, términos y condiciones) y posibles iconos de redes sociales para la conexión externa. Al igual que la Barra de Navegación, la información de contacto y otros datos que se muestran en el pie de página son configurables dinámicamente desde la tabla Empresa de Airtable, lo que garantiza su fácil actualización y coherencia con la imagen de la inmobiliaria.



**Figura 4.14:** Pie de página dinámica

En la Figura 4.14 se visualiza pie de página de la otra empresa configurada, donde se observa el cambio de logo, título, datos de la empresa y de colores.

#### 4. Interfaz para el Agente/Administrador de Gestión de la Base de Datos de Empresa (Grid View) de AirTable

Este apartado se centra en la gestión de la información global de la inmobiliaria, permitiendo a los administradores centralizar y actualizar los datos corporativos que se utilizan para la configuración dinámica de la aplicación web. Corresponde al requisito funcional que busca facilitar la personalización y portabilidad del sistema.

**Figura 4.15:** Vista de Cuadrícula (Grid View) de Empresa en AirTable

**Figura 4.16:** Vista de Cuadrícula (Grid View) de Empresa en AirTable

Los administradores utilizan la Tabla Empresa en Airtable como su herramienta principal para este fin. La vista de cuadrícula (Grid View), mostrada en las Figuras 4.15 y 4.16, permite una manipulación directa de campos clave como:

- **Nombre de la Empresa:** Para el título de la aplicación y el branding general.
- **Logo:** Un campo de adjunto para subir la imagen del logotipo de la inmobiliaria, que se mostrará en la barra de navegación y otras secciones clave.
- **Colores:** Campos (ej., texto con códigos hexadecimales) para definir los colores globales principales de la aplicación (color primario, secundario, etc.), permitiendo personalizar la paleta de colores de la interfaz.
- **Dirección, Teléfono, Email:** Datos de contacto que se replican en el pie de página y otras secciones informativas de la aplicación web.
- Otros campos relevantes para la información corporativa o de branding.

Lo que es fundamental de esta gestión es que todos estos campos son consultados por el backend (Laravel) y luego por el frontend (Angular) para **configurar dinámicamente** la apariencia y la información de contacto de la aplicación web. Esto significa que cambios en el nombre de la empresa, el logo, los colores o los datos de contacto realizados en Airtable se reflejan automáticamente en la aplicación web **sin necesidad de modificar código**.

#### 4.4.2.2 Módulo: Gestión de Propiedades

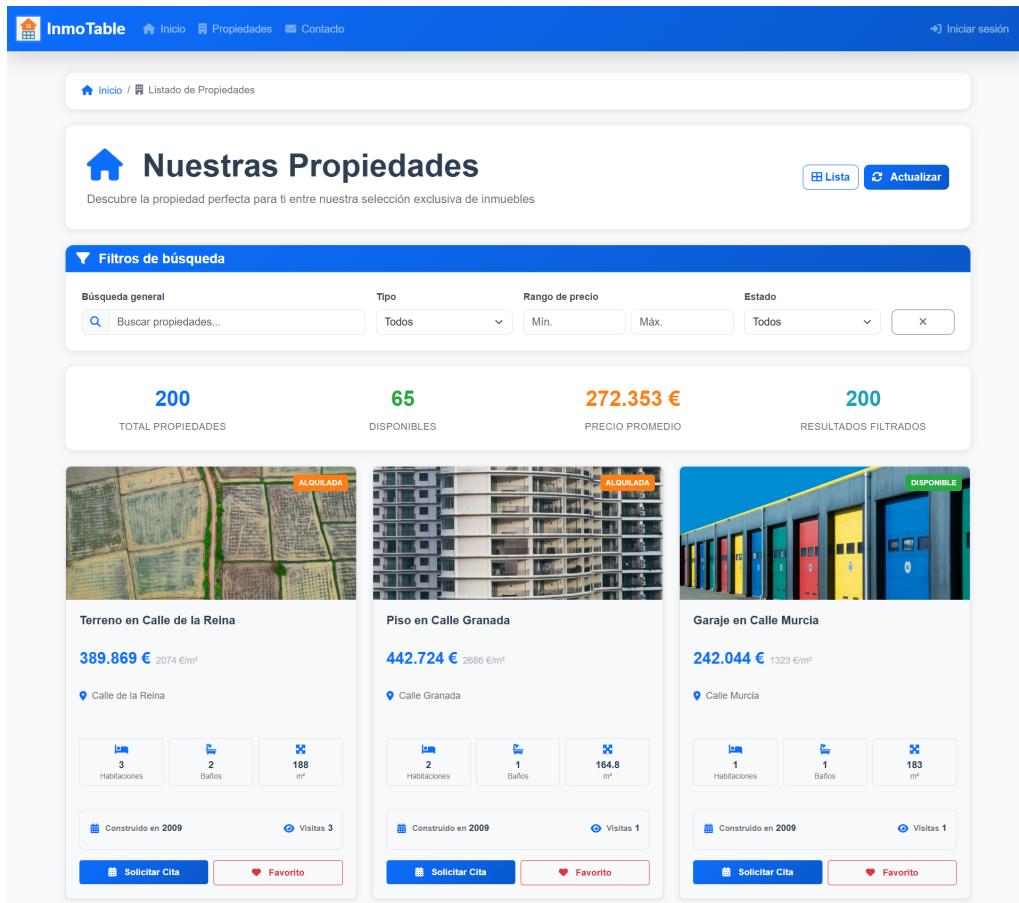
Este módulo es el corazón de la aplicación, permitiendo tanto la visualización de inmuebles por parte de los clientes como su gestión detallada por los agentes.

##### 1. Interfaz para el Cliente (Aplicación Web)

###### a) Listado de propiedades

La visualización del catálogo de propiedades se implementa a través del componente `property-list.component`. Inspirada en el mockup de la Figura 4.4), este listado se presenta en un formato de tarjetas (cards) claras y concisas, organizadas en una cuadrícula (grid) que se adapta dinámicamente al tamaño de la pantalla (diseño responsivo).

#### 4.4. DISEÑO E IMPLEMENTACIÓN



The screenshot shows the homepage of the InmoTable website. At the top, there's a blue header bar with the logo 'InmoTable' and navigation links for 'Inicio', 'Propiedades', and 'Contacto'. On the right side of the header is a 'Iniciar sesión' button. Below the header, the main content area has a title 'Nuestras Propiedades' with a house icon, followed by a subtitle 'Descubre la propiedad perfecta para ti entre nuestra selección exclusiva de inmuebles'. There are two buttons: 'Lista' and 'Actualizar'. A section titled 'Filtros de búsqueda' (Search filters) includes a search bar, dropdown menus for 'Tipo' (Todos), 'Rango de precio' (Min., Máx.), and 'Estado' (Todos), and a clear 'X' button. Below this, summary statistics are displayed: 'TOTAL PROPIEDADES' (200), 'DISPONIBLES' (65), 'PRECIO PROMEDIO' (272.353 €), and 'RESULTADOS FILTRADOS' (200). The main content area shows three property cards in a grid:

- Terreno en Calle de la Reina**: Alquilada, 389.869 €, 2074 €/m<sup>2</sup>. Address: Calle de la Reina. Features: 3 Habitaciones, 2 Baños, 188 m<sup>2</sup>. Built in 2009, 3 visits. Actions: 'Solicitar Cita' (blue button), 'Favorito' (red button).
- Piso en Calle Granada**: Alquilada, 442.724 €, 2686 €/m<sup>2</sup>. Address: Calle Granada. Features: 2 Habitaciones, 1 Baño, 164.8 m<sup>2</sup>. Built in 2009, 1 visit. Actions: 'Solicitar Cita' (blue button), 'Favorito' (red button).
- Garaje en Calle Murcia**: Disponible, 242.044 €, 1323 €/m<sup>2</sup>. Address: Calle Murcia. Features: 1 Habitaciones, 1 Baño, 183 m<sup>2</sup>. Built in 2009, 1 visit. Actions: 'Solicitar Cita' (blue button), 'Favorito' (red button).

**Figura 4.17:** Listado de propiedades en formato Grid (por defecto)

Cada tarjeta de propiedad (observada en las Figura 4.17) contiene la siguiente información esencial para una rápida identificación:

- **Imagen Principal:** Una fotografía destacada de la propiedad.
- **Título:** Nombre de la propiedad.
- **Precio:** El precio de venta o alquiler, de manera prominente.
- **Dirección:** Ubicación de la propiedad.
- **Características Clave:** Iconos o texto breve que indican atributos importantes como el número de habitaciones, número de baños y m<sup>2</sup>.
- **Tipo de Propiedad:** (ej. Piso, Dúplex).
- **Llamadas a la Acción en Tarjeta:** Cada tarjeta incluye acciones rápidas como "Solicitar Cita" (dirigiendo al appointment-form.component) y "Agregar a Favoritos". El detalle de estas funcionalidades se explicará más adelante en sus respectivas secciones.

Además de la vista en cuadrícula, la interfaz ofrece una opción para visualizar las propiedades en modo listado (véase Figura 4.18). Este formato proporciona una disposición más compacta y orientada a texto, ideal para una revisión rápida de múltiples propiedades, mostrando información clave en un estilo similar a una tabla.

#### 4.4. DISEÑO E IMPLEMENTACIÓN

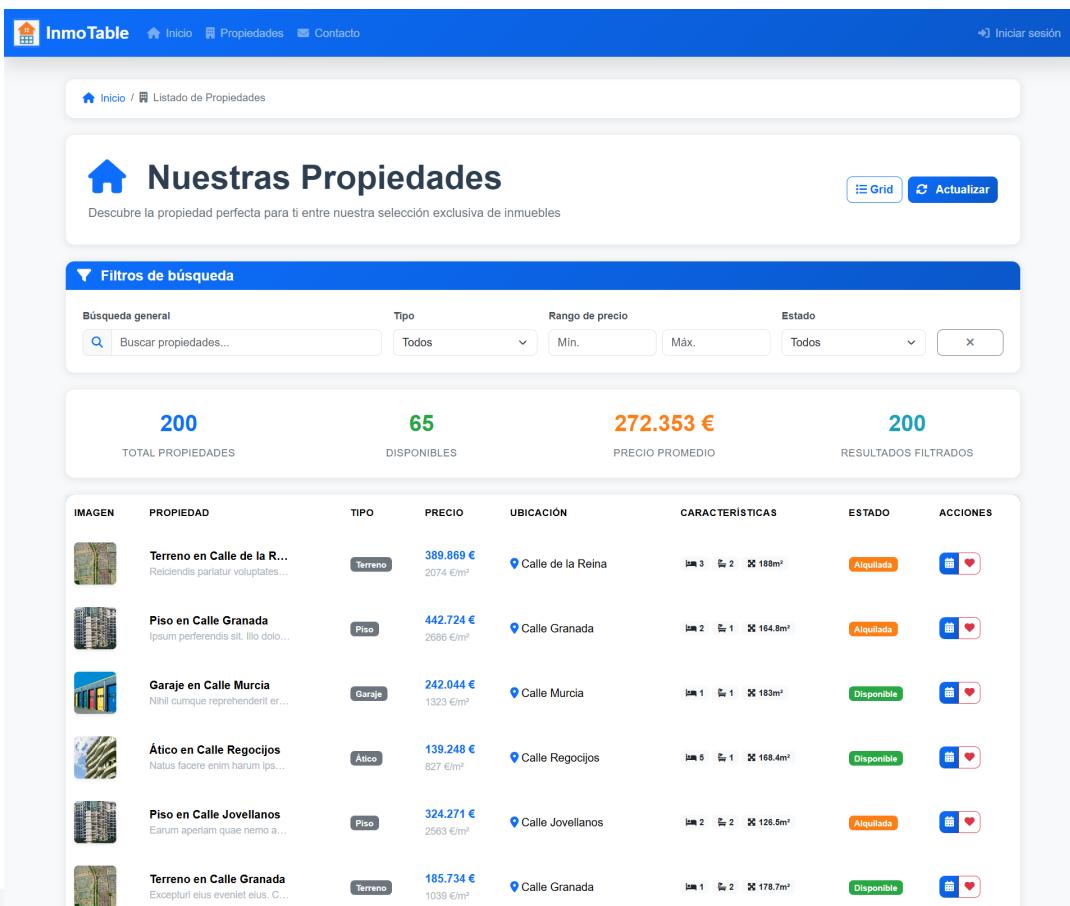
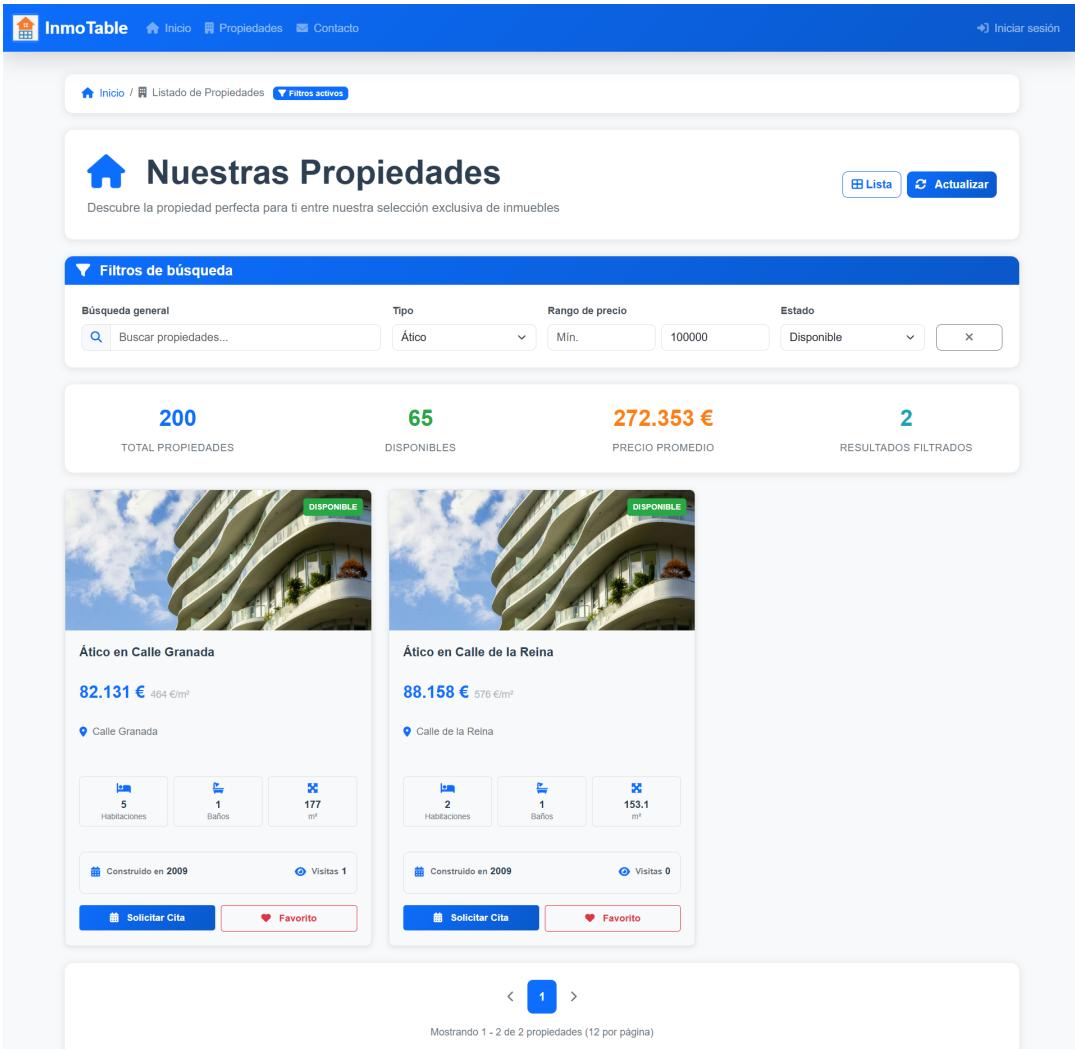


IMAGEN	PROPIEDAD	TIPO	PRECIO	UBICACIÓN	CARACTERÍSTICAS	ESTADO	ACCIONES
	Terreno en Calle de la Reina	Terreno	389.869 € 2074 €/m²	Calle de la Reina	3 habitaciones, 2 baños, 188m²	Alquilada	
	Piso en Calle Granada	Piso	442.724 € 2686 €/m²	Calle Granada	2 habitaciones, 1 baño, 164.8m²	Alquilada	
	Garaje en Calle Murcia	Garaje	242.044 € 1323 €/m²	Calle Murcia	1 habitación, 1 baño, 183m²	Disponible	
	Ático en Calle Regocijos	Ático	139.248 € 827 €/m²	Calle Regocijos	5 habitaciones, 1 baño, 168.4m²	Disponible	
	Piso en Calle Jovellanos	Piso	324.271 € 2563 €/m²	Calle Jovellanos	2 habitaciones, 2 baños, 126.5m²	Alquilada	
	Terreno en Calle Granada	Terreno	185.734 € 1039 €/m²	Calle Granada	1 habitación, 2 baños, 178.7m²	Disponible	

Figura 4.18: Listado de propiedades en formato List

Para facilitar la exploración del extenso inventario, la interfaz incorpora capacidades de filtrado y búsqueda avanzadas (visible en la Figura 4.19).

#### 4.4. DISEÑO E IMPLEMENTACIÓN



The screenshot shows the 'Nuestras Propiedades' (Our Properties) section of the InmoTable website. At the top, there is a search bar with placeholder text 'Buscar propiedades...', dropdown menus for 'Tipo' (Type: Ático), 'Rango de precio' (Price range: Min. 100000), and 'Estado' (Status: Disponible). Below the search bar, summary statistics are displayed: '200 TOTAL PROPIEDADES', '65 DISPONIBLES', '272.353 € PRECIO PROMEDIO', and '2 RESULTADOS FILTRADOS'. Two property cards are shown below:

- Ático en Calle Granada**: Precio: 82.131 € / 464 €/m<sup>2</sup>. Ubicación: Calle Granada. Detalles: 5 Habitaciones, 1 Baños, 177 m<sup>2</sup>. Construido en 2009. Visitas: 1. Botones: 'Solicitar Cita' (azul) y 'Favorito' (rojo).
- Ático en Calle de la Reina**: Precio: 88.158 € / 576 €/m<sup>2</sup>. Ubicación: Calle de la Reina. Detalles: 2 Habitaciones, 1 Baños, 153.1 m<sup>2</sup>. Construido en 2009. Visitas: 0. Botones: 'Solicitar Cita' (azul) y 'Favorito' (rojo).

At the bottom of the page, there is a navigation bar with arrows and the number '1' indicating the current page.

**Figura 4.19:** Filtros del listado de propiedades

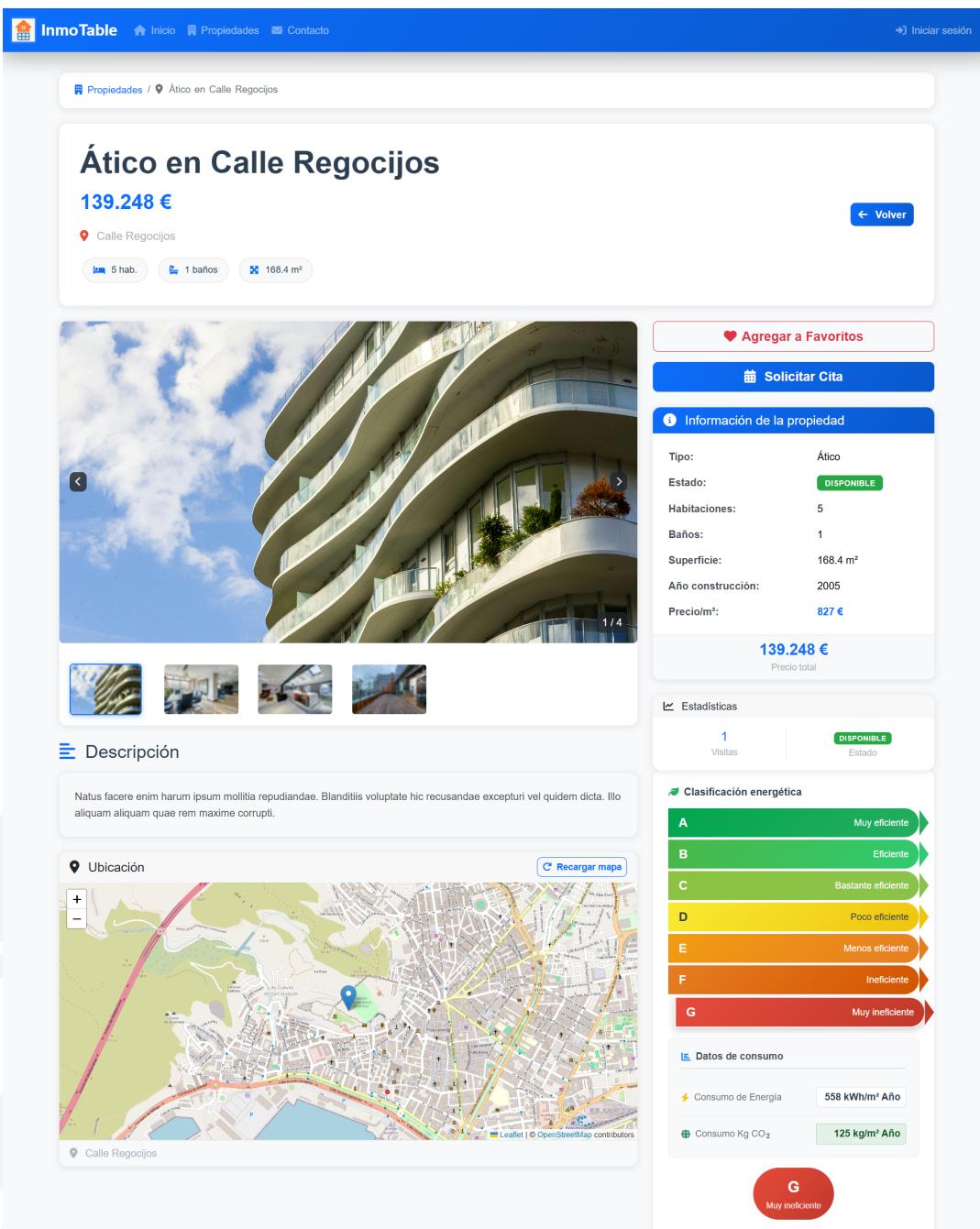
Los usuarios pueden aplicar filtros dinámicos basados en:

- **Búsqueda general:** Búsqueda por título de la propiedad y/o ubicación geográfica.
- **Tipo de Propiedad:** Filtrado por categorías (Piso, Dúplex, Ático, etc.).
- **Rango de Precios:** El precio de venta o alquiler, de manera prominente.
- **Rango de Precios:** Establecimiento de un presupuesto mínimo y máximo.
- **Estado:** Filtrado por estado de disponibilidad (Disponible, Alquilada, Vendida).

##### b) Detalle de Propiedad

La vista de detalle de propiedad, implementada en el `property-detail` component, ofrece una inmersión completa en la información de un inmueble seleccionado. Inspirada en el mockup de la Figura 4.5) y reflejada en la Figura 4.20).

#### 4.4. DISEÑO E IMPLEMENTACIÓN



The screenshot displays a property listing for an attic in Calle Regocijos. The main header shows the price **139.248 €**. Below it, the address **Calle Regocijos** is listed, along with room details: **5 hab.**, **1 baños**, and **168.4 m<sup>2</sup>**. To the right, there are buttons for **Agregar a Favoritos** (Add to Favorites) and **Solicitar Cita** (Request Appointment). A large image of the building's exterior is shown, with a thumbnail gallery below it. To the right, detailed property information is provided, including Type: **Ático**, State: **DISPONIBLE** (Available), Habitaciones: **5**, Baños: **1**, Superficie: **168.4 m<sup>2</sup>**, Año construcción: **2005**, and Precio/m<sup>2</sup>: **827 €**. The total price is also displayed as **139.248 €**. Below this, a section for **Estadísticas** (Statistics) shows **1 Visitas** (Visits) and **DISPONIBLE Estado** (Available Status). A **Clasificación energética** (Energy Efficiency Classification) chart is present, ranging from **A Muy eficiente** (Very efficient) at the top to **G Muy ineficiente** (Very inefficient) at the bottom. Consumption data is also listed: **Consumo de Energía 558 kWh/m<sup>2</sup> Año** (Energy consumption 558 kWh/m<sup>2</sup> Year) and **Consumo Kg CO<sub>2</sub> 125 kg/m<sup>2</sup> Año** (CO<sub>2</sub> consumption 125 kg/m<sup>2</sup> Year).

**Figura 4.20:** Filtros del listado de propiedades

Esta página presenta un diseño limpio y enfocado que prioriza la información visual y detallada:

- **Galería de Imágenes Principal:** Una sección prominente en la parte superior, que permite al usuario explorar múltiples fotografías de la propiedad en alta resolución, capturando diferentes ángulos y estancias.

- **Información Esencial y Precio:** El precio de la propiedad se muestra de manera destacada junto a la dirección y la referencia única.
- **Características Detalladas:** Se presenta un resumen visual y textual de las características clave, como el número de habitaciones y baños, superficie en metros cuadrados, y el tipo de propiedad.
- **Descripción Completa:** Un bloque de texto extenso proporciona una descripción detallada del inmueble, sus ventajas, distribución y entorno.
- **Calificación Energética:** Se incluye una sección dedicada a la calificación energética de la propiedad, con una representación visual del nivel de eficiencia (ej., A, B, C), añadiendo valor informativo al cliente. (Esto se refleja en `energy-rating.component`).
- **Llamadas a la Acción:** La página incorpora llamadas a la acción claras, como botones o enlaces, para que el cliente pueda "Solicitar Cita" para la propiedad (dirigiendo al `appointment-form.component`) o "Agregar a Favoritos". El detalle de estas funcionalidades se explicará más adelante en sus respectivas secciones.

La disposición de los elementos busca maximizar la legibilidad y la inmersión del usuario en la información de la propiedad, garantizando que el cliente encuentre todos los detalles que necesita antes de decidir un siguiente paso. El diseño responsive, logrado con **Angular Material** y **Bootstrap**, asegura que esta riqueza de información sea accesible y estéticamente agradable en cualquier dispositivo.

## 2. Interfaz para el Agente/Administrador (Airtable)

Este módulo en Airtable está diseñado para la gestión exhaustiva y colaborativa del inventario de propiedades por parte de los agentes inmobiliarios y administradores. Su objetivo es maximizar la eficiencia en la entrada, manipulación, visualización y análisis de los datos de propiedades sin necesidad de programación. Corresponde principalmente a los requisitos funcionales **RF 2.1, RF 2.2, RF 2.3, RF 2.4, RF 5.1, RF 5.2, RF 5.3, RF 5.4, RF 5.5, RF 5.6 y RF 6.3**.

### ■ Gestión de la Base de Datos de Propiedades (Grid View)

Los agentes utilizan la interfaz de tabla principal de Airtable como su centro de operaciones para ver, editar, añadir y eliminar registros de propiedades. Esta vista, similar a una hoja de cálculo, permite una manipulación directa y flexible de los datos. Muestra todas las columnas de la tabla Propiedades (referencia, dirección, precio, tipo, superficie, número de habitaciones/baños, descripción, fotos, agente asignado, estado, calificación energética, fecha de publicación) con sus tipos de campo configurados (texto, número, selección única, adjunto, etc.). La Figura 4.21 ilustra esta configuración, donde se aprecian los campos y la facilidad de edición directa.

#### 4.4. DISEÑO E IMPLEMENTACIÓN

Inmobiliaria											Datos	Automatizaciones	Interfaces	Formularios	Ayuda	Reaching plan limits	Compartir	Imprimir	
Agencias		Propiedades		Clientes		Citas		Usuarios		Empresa		Contacto		+		Extensiones		Herramientas	
Vistas		Grid view		Filtrar		Ocultar campos		Agrupar		Clasificar		Configurar color		Compartir y sincronizar					
Q Encontrar una vista		ID Propiedad	A Título	Descripción	Precio	Precio m²	Estado	Fecha de Registro	Fecha de Venta	Días en venta	Días hasta su venta								
Grid view		1	Ático en Calle Altamira	Doloribus consequuntur aut...	100.090,00	505	Disponible	24/3/2025 3:30pm			93								
Lista Resumida		2	Casa en Calle de la Reina	Repudiandis nulla similique...	277.409,00	5016	Vendida	8/2/2025 3:30pm	22/5/2025 12:00am			102							
Kanban Estado Disponibilidad		3	Ático en Calle del Barrio...	Illi architecto totam necesse...	240.176,50	2274	Alquilada	15/3/2025 3:30pm											
Kanban Tipo Propiedad		4	Ático en Calle Jovellanos	A quam quaerat voluptatis q...	256.741,20	3423	Disponible	22/5/2025 12:00am			35								
Galería tipo Cards		5	Casa en Calle Granada	Quidem porro quae id perfec...	397.880,00	1611	Disponible	22/5/2025 12:00am			35								
Crear		6	Ático en Calle Granada	Aperiam quae reciedisti aut...	82.131,10	464	Disponible	22/5/2025 12:00am			35								
+		7	Casa en Calle Real del Barri...	Elegiendi possimus quas qu...	419.811,20	1990	Alquilada	3/5/2025 3:30pm											
Cuadrícula	+	8	Local en Calle Altamira	Natus et ea pariatur et eaq...	114.292,20	1467	Vendida	22/5/2025 12:00am	22/5/2025 12:00am			0							
Calendario	+	9	Dúplex en Avenida Federico...	At incidunt eius quo odio et...	99.342,00	566	Vendida	3/5/2025 3:30pm	3/5/2025 3:30pm			9							
Galería	+	10	Ático en Calle Rambla Obis...	Parapatrum explicabo nam sol...	374.324,80	2009	Vendida	15/5/2025 3:30pm	22/5/2025 12:00am			6							
Kanban	+	11	Terreno en Calle de la Reina	Expedita magni perspectis...	392.883,00	2593	Vendida	5/4/2025 3:30pm	3/5/2025 12:00am			46							
Línea de tiempo	Team	12	Piso en Calle Granada	Architecto optio ex et sed...	319.523,50	2163	Alquilada	22/5/2025 12:00am											
+		13	Dúplex en Calle Rambla Obis...	Optio nulla at ventatis dolor...	81.145,60	402	Disponible	8/4/2025 3:30pm			78								
Crear		14	Terreno en Calle de la Reina	Reciedisti pariatur voluptat...	389.869,30	2074	Alquilada	22/5/2025 12:00am											
+		15	Garaje en Calle de la Reina	Iure aperiam rerum harum ut...	59.465,40	370	Disponible	22/5/2025 12:00am			35								
Cuadrícula	+	16	Terreno en Calle Regocijos	Iste excepturi sunt eius verit...	157.102,10	855	Vendida	22/3/2025 3:30pm	22/5/2025 12:00am										60
Calendario	+	17	Piso en Calle Altamira	Neque ut nenter tenuerit. Ve...	346.542,20	2866	Alquilada	3/2/2025 3:30pm											
Galería	+	18	Casa en Calle Jovellanos	Repudiandis nihil eum dolorem...	136.215,50	866	Alquilada	4/4/2025 3:30pm											
Kanban	+	19	Piso en Calle Granada	Esse neque labore minus in...	72.764,80	291	Disponible	18/4/2025 3:30pm				68							
Línea de tiempo	Team	20	Local en Calle Rambla Obis...	Elegiendi laboriosum exercitat...	50.969,90	398	Alquilada	6/5/2025 3:30pm											
+		21	Piso en Calle Regocijos	Veniam dignissimos distinctio...	310.631,10	1961	Disponible	3/1/2025 3:30pm			173								
Lista	+	22	Piso en Calle Murcia	Anime recusandae nisi repudi...	317.765,80	1718	Alquilada	19/3/2025 3:30pm											
Gantt	Team	23	Terreno en Paseo de Almería	Voluptatum ipsa repellendus...	304.352,10	2686	Disponible	22/5/2025 12:00am			35								
Nueva sección	Team	24	Dúplex en Calle Murcia	Commodi voluptate non dicta...	471.080,40	9077	Disponible	7/4/2025 3:30pm			79								
Formulario	+	201 propiedades	Suma 20305		54.470.539,99	Suma <math>\infty</math>										Suma 4197			Suma 1972

**Figura 4.21:** Vista de Cuadrícula (Grid View) de Propiedades en AirTable

- **Vista de Listado Compacto:** Una vista de listado optimizada (visible en la Figura 4.22) que muestra las propiedades en un formato tabular más compacto que la cuadricula principal, con campos resumidos y esenciales para una revisión rápida de grandes volúmenes de datos. Es ideal para una visualización rápida de la información clave sin necesidad de abrir cada registro.

**Figura 4.22:** Vista de Listado Compacto de Propiedades en AirTable

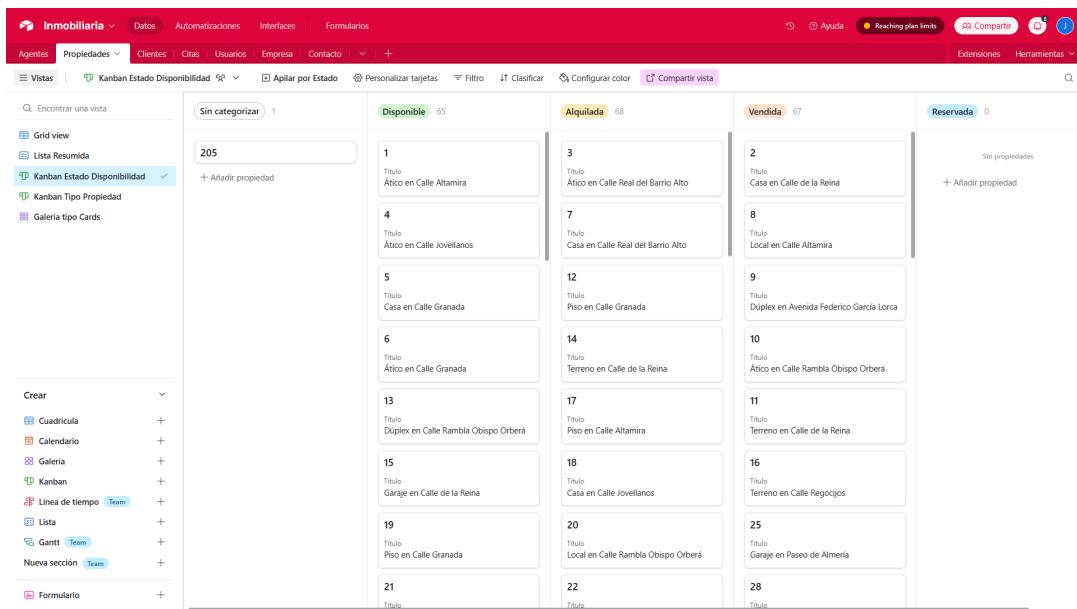
#### 4.4. DISEÑO E IMPLEMENTACIÓN

Además permite añadir nuevas propiedades directamente desde esta vista mediante un botón o una opción rápida, agilizando el flujo de trabajo de entrada de datos (visible en la Figura 4.23).



**Figura 4.23:** Añadir Propiedad en Vista de Listado Compacto de Propiedades en AirTable

- **Vista Kanban por Estado de Disponibilidad:** Una vista de tablero tipo Kanban (visible en la Figura 4.24) que organiza las propiedades en columnas según su Estado (ej., "Disponible", "Reservada", "Vendida", "Alquilada"). Esto permite a los agentes arrastrar y soltar propiedades entre estados, visualizando rápidamente el flujo de trabajo de ventas/alquileres y el progreso de cada inmueble.



**Figura 4.24:** Vista Kanban por Estado de Disponibilidad de Propiedades en AirTable

- **Vista Kanban por Tipo de Propiedad:** Otra vista Kanban (visible en la Figura 4.25) que clasifica las propiedades por su Tipo de Propiedad (ej., "Apartamentos", "Casas", "Locales"), ofreciendo una visión clara de la composición del inventario por categoría y facilitando la gestión por tipos de inmueble.



#### 4.4. DISEÑO E IMPLEMENTACIÓN

The screenshot shows a Kanban view in AirTable for property types. The columns are labeled 'Piso' (30), 'Casa' (31), 'Duplex' (28), and 'Ático' (27). Each card contains a large image, a title, and a price. A sidebar on the left shows navigation options like 'Grid view', 'Lista Resumida', and 'Galería tipo Cards'.

Tipo	Imagen	Título	Precio
Piso		Piso en Calle Granada	319.525,1
Casa		Casa en Calle de la Reina	277.409,9
Duplex		Duplex en Avenida Federico García Lorca	99.342,1
Ático		Ático en Calle Altamira	100.099,0
Piso		Piso en Calle Altamira	346.542,2
Casa		Casa en Calle Granada	397.880,8
Duplex		Duplex en Calle Rambla Obispo Orberá	81.145,6
Ático		Ático en Calle Real del Barrio Alto	240.176,5

Figura 4.25: Vista Kanban por Tipo de Propiedades en AirTable

- **Vista de Galería Visual:** Una vista de galería que presenta las propiedades en un formato tipo tarjeta (card) prominente, con una gran imagen y campos clave debajo (precio, dirección). Es ideal para la navegación visual y para presentaciones rápidas (visible en la Figura 4.26).

The screenshot shows a Gallery view in AirTable for properties. Each card displays a large image, a title, a brief description, and a price. A sidebar on the left shows navigation options like 'Grid view', 'Lista Resumida', and 'Galería tipo Cards'.

Imagen	Título	Descripción	Precio
	Ático en Calle Altamira	Doloribus consequuntur aut qui molestias. Adipisci atque doloremque nobis natus querat. In ipsum eius sit quis cum in.	100.099,0
	Casa en Calle de la Reina	Repudiandae nulla similique neque provident. Quo voluptates sequi sint fugit.	277.409,9
	Ático en Calle Real del Barrio Alto	Ilo architecto totam necessitatibus quaeerit distinctio. Suscipit odit corporis consequatur. Atque et ipsa ut labore et dolore incidunt nesciunt facere voluptatum consequuntur...	240.176,5
	Ático en Calle Jovellanos	A quam quaeat voluptas quis magni autem. Sapiente magni dicta fugit cum. Ut illam nobis laborum beatae facilis voluptates rem.	256.741,2
	Casa en Calle Granada	Quidem pono quae id perferenda fuga tempore. Dolorem veniam accusantium mollitia. Dolore architecto occassum dolorem temporibus similique architecto ...	397.880,8
	Ático en Calle Granada	Aperiam quas reciendis autem. Modi etem autem doloribus ratione ex ea.	
	Casa en Calle Real del Barrio Alto	Elegendi possimus quas quis optio laborum dolamit. Nam ex modicis et.	
	Local en Calle Altamira	Natus et ea paritur et eaque consecutetur officia. Ita sed ut non enim autem et.	
	Duplex en Avenida Federico García Lorca	At incidenti eius. Quo odio ea qui mollitia.	
	Ático en Calle Rambla Obispo Orberá	Paratur explicabo nam soluta voluptate. Cest undicatae narrat et.	

Figura 4.26: Vista de Galería Visual de Propiedades en AirTable

#### 4.4.2.3 Módulo: Gestión de Clientes y Preferencias

Este módulo abarca la información de los clientes y sus interacciones con la inmobiliaria, tanto desde la perspectiva del cliente como de la gestión del agente.

##### 1. Interfaz para el Cliente (Aplicación Web)

###### a) Autenticación (Inicio de Sesión)

Provee un sistema seguro y claro para que los usuarios puedan acceder a funcionalidades personalizadas en la aplicación web. Corresponde al requisito funcional **RF 1.5**.

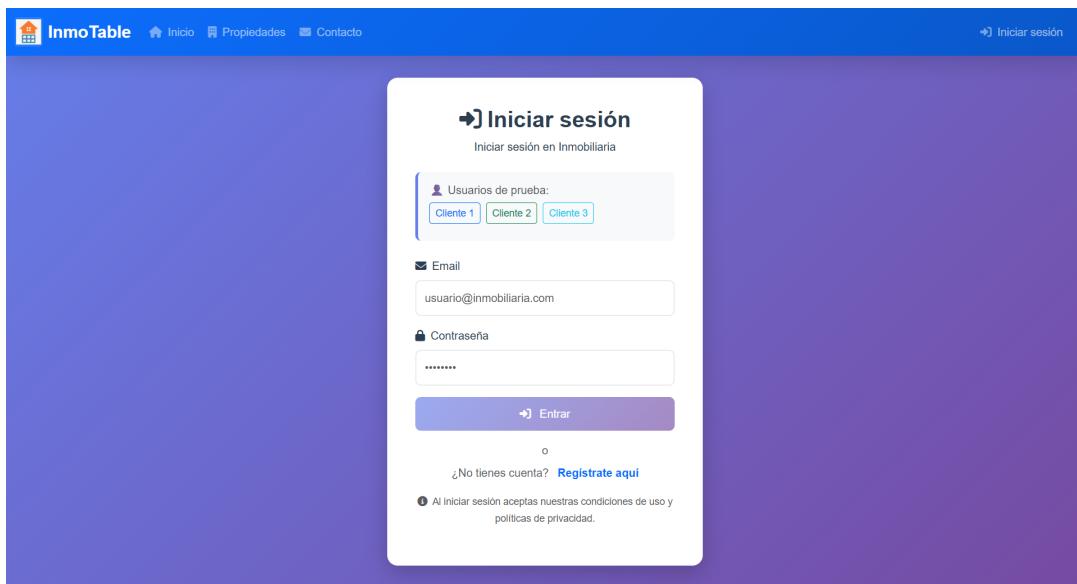


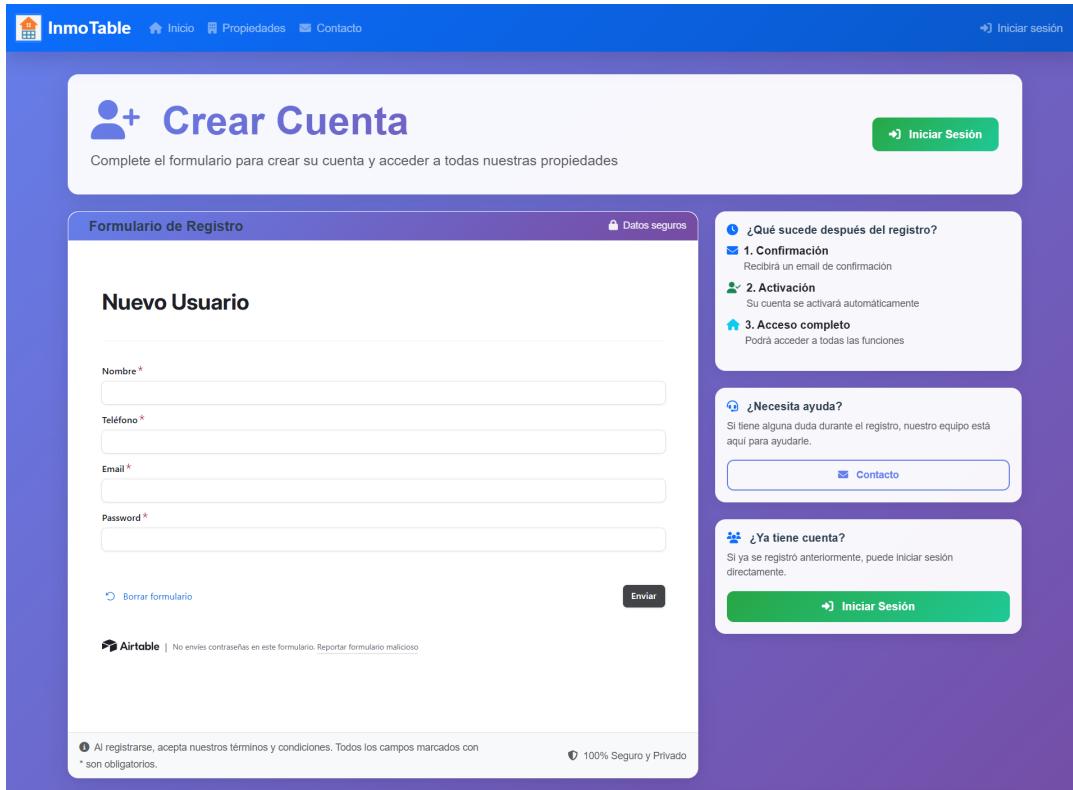
Figura 4.27: Inicio de sesión en Aplicación Web

Representado por la Figura 4.27, este formulario (`login.component.html`) está construido directamente en Angular. Requiere credenciales (email/nombre de usuario y contraseña) y, tras una validación exitosa, otorga acceso a las secciones restringidas.

###### b) Autenticación (Registro)

Provee un sistema seguro y claro para que los usuarios puedan crear una cuenta en la aplicación web. Corresponde al requisito funcional **RF 1.4**.

#### 4.4. DISEÑO E IMPLEMENTACIÓN



**Figura 4.28:** Registro en Aplicación Web

Visible en la Figura 4.28, este formulario, a diferencia del login, se ha implementado incrustando un formulario de Airtable. Permite a los nuevos usuarios crear una cuenta de cliente introduciendo datos como nombre, email, teléfono y contraseña. Una vez completado el registro, un registro de usuario se guarda en la tabla Usuarios de Airtable. Adicionalmente, se ha configurado una automatización en Airtable que, tras la creación de este usuario, crea automáticamente un registro de Cliente en la tabla Clientes, asociándolo a dicho usuario. Esto agiliza el flujo de gestión de clientes, delegando el proceso de alta inicial y vinculación al ecosistema de Airtable, cuyo detalle se ampliará en el Módulo: Automatizaciones (Visión General). Esta elección se justifica por la rapidez y sencillez de implementación que ofrece un formulario *low-code* de Airtable, demostrando una forma eficiente de recopilar datos de nuevos usuarios sin necesidad de un desarrollo a medida completo para esta fase inicial del ciclo de vida del cliente, contrastando con el esfuerzo de los formularios personalizados de Angular.

##### c) Menú Desplegable (Dropdown) de Usuario Logueado

Tras un inicio de sesión exitoso, la interfaz de usuario se actualiza dinámicamente. En la barra de navegación (Nav), se muestra un menú desplegable (visible en la Figura 4.29) que proporciona al usuario autenticado opciones adicionales, como "Ver mi perfil" o "Cerrar Sesión".

#### 4.4. DISEÑO E IMPLEMENTACIÓN



**Figura 4.29:** Menú Desplegable (Dropdown) de Usuario Logueado en Aplicación Web

##### d) Mi Perfil

Al hacer clic en "Ver mi perfil", el cliente accede a una vista (`user-form.component.html`) diseñada con componentes Angular nativos (visible en la Figura 4.30), donde pueden visualizar y editar sus datos personales y preferencias. La construcción de este formulario en Angular contrasta con la simplicidad del registro en Airtable, ilustrando la mayor dificultad y el esfuerzo programático necesario para desarrollar interfaces totalmente personalizadas en Angular, a cambio de un control granular sobre la lógica y la experiencia de usuario.

**Figura 4.30:** Mi Perfil en Aplicación Web

### e) Formulario de Contacto

Al hacer clic en "Ver mi perfil", el cliente accede a una vista (`user-form.component.html`) diseñada con componentes Angular nativos (basado en la Figura 4.7 del mockup y visible en la Figura 4.30), donde pueden visualizar y editar sus datos personales y preferencias. La construcción de este formulario en Angular contrasta con la simplicidad del registro en Airtable, ilustrando la mayor dificultad y el esfuerzo programático necesario para desarrollar interfaces totalmente personalizadas en Angular, a cambio de un control granular sobre la lógica y la experiencia de usuario.

Proporcionar un canal de comunicación general para cualquier usuario con la inmobiliaria. Corresponde al requisito funcional **RF 1.6**.

The screenshot shows a web browser window for 'InmoTable'. The header includes a logo, navigation links for 'Inicio', 'Propiedades', and 'Contacto', and a user profile 'Juan'. The main content area has a breadcrumb trail 'Propiedades / Contacto' and a title '✉️ Contacto'. A sub-header says '¿Tiene alguna pregunta? Nos encantaría ayudarle. Complete el formulario y nos pondremos en contacto con usted.' Below this is a blue header bar labeled 'Formulario de Contacto'. The main form section is titled 'Contacto' and contains four input fields: 'Nombre\*' (Juan Diego Ortega Aranda), 'Email\*' (joa102@inlumine.ual.es), 'Teléfono\*' (612345789), and a large 'Mensaje' text area. At the bottom left is a 'Borrar formulario' button, and at the bottom right is an 'Enviar' button. A small note from Airtable states 'No envíes contraseñas en este formulario. Reportar formulario malicioso'. The footer includes a note about required fields, a data protection statement, and a copyright notice.

**Figura 4.31:** Contacto en Aplicación Web

Este formulario (visible en la Figura 4.31, correspondiente al `contact.component`) se ha implementado incrustando un formulario de Airtable. Permite a cualquier usuario, sin necesidad de estar registrado, enviar mensajes o consultas. Contiene campos estándar como nombre, dirección de correo electrónico y un área de texto para el mensaje. Los datos enviados se guardan directamente en la tabla Contacto de Airtable, lo que simplifica la captura de leads y la gestión por parte del equipo interno.

f) **Gestión de Propiedades de Interés (Favoritos)**

Permite a los clientes registrados guardar y acceder fácilmente a las propiedades que les interesan, personalizando su experiencia de búsqueda y seguimiento. Corresponde al requisito funcional **RF 4.1** y **RF 4.2** (en cuanto al interés del cliente en propiedades).

- **Acción "Aregar a Favoritos":** Esta funcionalidad se ofrece directamente en las vistas de listado de propiedades (en las tarjetas) y en la página de detalle de propiedad, mediante un botón o ícono intuitivo.

La Figura 4.32 muestra el ícono para añadir una propiedad.

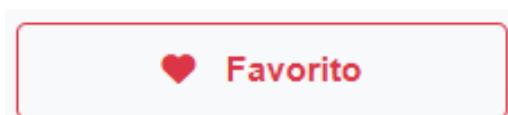


Figura 4.32: Botón Agregar Propiedad Interés (Favoritos) en Aplicación Web

Una vez agregada, el sistema proporciona una confirmación visual (ej., mensaje Figura 4.33).



Figura 4.33: Mensaje Agregar Propiedad Interés (Favoritos) en Aplicación Web

Y el ícono cambia para reflejar el estado (ej., como en la Figura 4.34, para "Quitar de Favoritos").

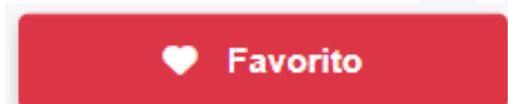


Figura 4.34: Botón Quitar Propiedad Interés (Favoritos) en Aplicación Web

Si se quita, también hay una confirmación visual (ej., mensaje Figura 4.35). Los clientes autenticados pueden marcar o desmarcar un inmueble como "favorito" con un solo clic.



Figura 4.35: Mensaje Quitar Propiedad Interés (Favoritos) en Aplicación Web

### g) Vista "Mis Propiedades de Interés"

Se implementa como una sección dedicada para el cliente (property - list - interested .component) a la que acceden los clientes registrados. Aquí solo se muestran las propiedades que el usuario ha marcado previamente como favoritas. Esta vista ofrece la opción de ver las propiedades en un formato de tarjetas (cards) (Figura 4.36).

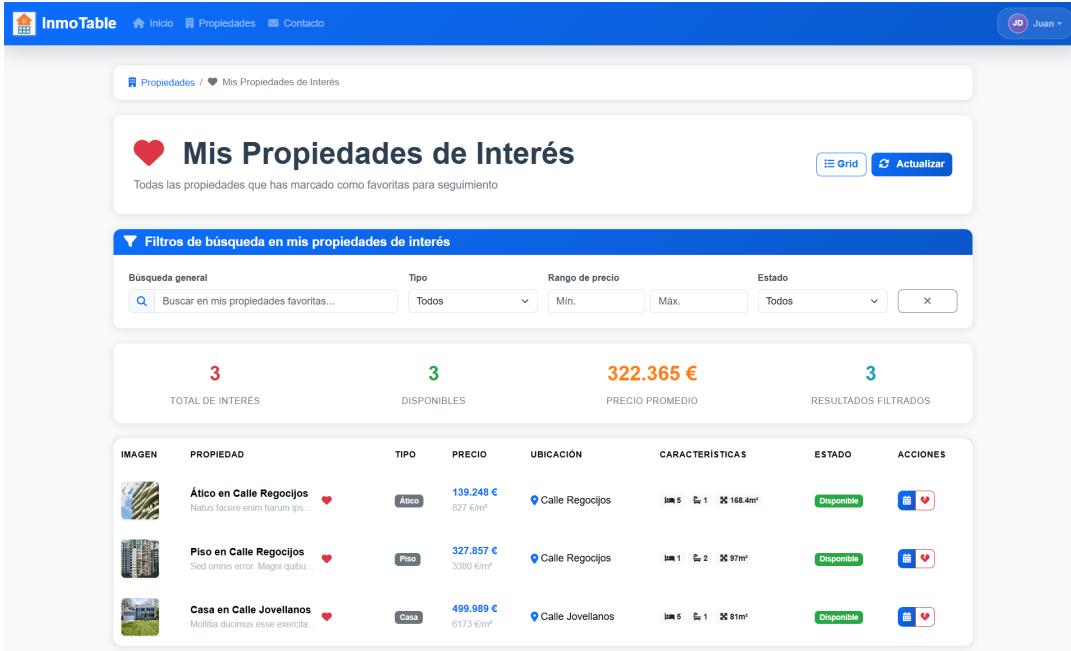
The screenshot shows the 'Mis Propiedades de Interés' (My Interested Properties) section of the InmoTable web application. At the top, there's a header with the logo, navigation links (Inicio, Propiedades, Contacto), and a user profile (Juan). Below the header, a breadcrumb navigation shows 'Propiedades / Mis Propiedades de Interés'. The main title 'Mis Propiedades de Interés' is displayed with a heart icon. A sub-header indicates 'Todas las propiedades que has marcado como favoritas para seguimiento'. There are buttons for 'Lista' (List) and 'Actualizar' (Update). A filter section titled 'Filtros de búsqueda en mis propiedades de interés' includes fields for general search, type, price range, and state. The results summary shows 'TOTAL DE INTERÉS: 3', 'DISPONIBLES: 3', 'PRECIO PROMEDIO: 322.365 €', and 'RESULTADOS FILTRADOS: 3'. Three property cards are listed:

- Ático en Calle Regocijos**: Precio: 139.248 €, m<sup>2</sup>: 827, Habitaciones: 5, Baños: 1, m<sup>2</sup>: 168,4. Construido en 2009. Includes 'Solicitar Cita' and 'Quitar' buttons.
- Piso en Calle Regocijos**: Precio: 327.857 €, m<sup>2</sup>: 3380, Habitaciones: 1, Baños: 2, m<sup>2</sup>: 97. Construido en 2009. Includes 'Solicitar Cita' and 'Quitar' buttons.
- Casa en Calle Jovellanos**: Precio: 499.989 €, m<sup>2</sup>: 3173, Habitaciones: 5, Baños: 1, m<sup>2</sup>: 81. Construido en 2009. Includes 'Solicitar Cita' and 'Quitar' buttons.

Figura 4.36: Mis Propiedad Interés (Favoritos) en formato Grid en Aplicación Web

O en modo listado (Figura 4.37), similar a las opciones del listado general de propiedades, pero enfocada en los elementos guardados por el usuario.

#### 4.4. DISEÑO E IMPLEMENTACIÓN



The screenshot shows a web application interface for managing favorite properties. At the top, there's a blue header bar with the logo 'InmoTable' and navigation links for 'Inicio', 'Propiedades', and 'Contacto'. A user profile 'Juan' is shown on the right. Below the header, a breadcrumb navigation 'Propiedades / Mis Propiedades de Interés' is displayed. The main title 'Mis Propiedades de Interés' is centered above a sub-header 'Todas las propiedades que has marcado como favoritas para seguimiento'. To the right of the title are 'Grid' and 'Actualizar' buttons. A section titled 'Filtros de búsqueda en mis propiedades de interés' (Search filters for my favorite properties) includes a search bar 'Buscar en mis propiedades favoritas...', dropdowns for 'Tipo' (Todos), 'Rango de precio' (Min., Máx.), and 'Estado' (Todos), and a clear button 'X'. Below this, four summary statistics are presented: 'TOTAL DE INTERÉS' (3), 'DISPONIBLES' (3), 'PRECIO PROMEDIO' (322.365 €), and 'RESULTADOS FILTRADOS' (3). The main content area displays a grid of three property items:

IMAGEN	PROPIEDAD	TIPO	PRECIO	UBICACIÓN	CARACTERÍSTICAS	ESTADO	ACCIONES
	Ático en Calle Regocijos	Ático	139.248 € 627 €/m <sup>2</sup>	📍 Calle Regocijos	habitaciones 6, baños 1, superficie 168.4m <sup>2</sup>	Disponible	
	Piso en Calle Regocijos	Piso	327.857 € 3380 €/m <sup>2</sup>	📍 Calle Regocijos	habitaciones 1, baños 2, superficie 97m <sup>2</sup>	Disponible	
	Casa en Calle Jovellanos	Casa	499.989 € 6173 €/m <sup>2</sup>	📍 Calle Jovellanos	habitaciones 5, baños 1, superficie 81m <sup>2</sup>	Disponible	

**Figura 4.37:** Mis Propiedad Interés (Favoritos) en formato List en Aplicación Web

La información de los intereses del cliente se refleja directamente en el campo Interesado en (Propiedades) de la tabla Clientes en Airtable, permitiendo a los agentes tener visibilidad de las preferencias de sus clientes.

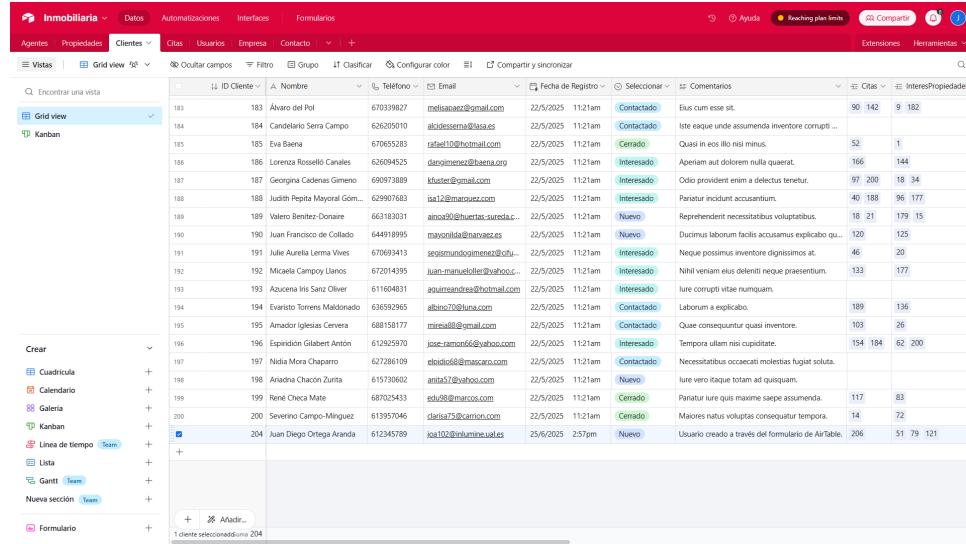
#### 2. Interfaz para el Agente/Administrador (Airtable)

Centraliza y gestiona la información de clientes, usuarios y agentes, permitiendo a los usuarios internos mantener una base de datos actualizada y segmentada para la operación del negocio. Corresponde a los requisitos funcionales **RF 4.1**, **RF 4.2**, **RF 6.1** y **RF 6.3** (en cuanto a la gestión de usuarios y agentes).

##### a) Gestión de la Base de Datos de Clientes (Grid View)

Los agentes y administradores utilizan la tabla Clientes en Airtable como su principal herramienta para consultar y editar los datos de los clientes.

#### 4.4. DISEÑO E IMPLEMENTACIÓN



The screenshot shows a grid view of client records. Each row contains the following information:

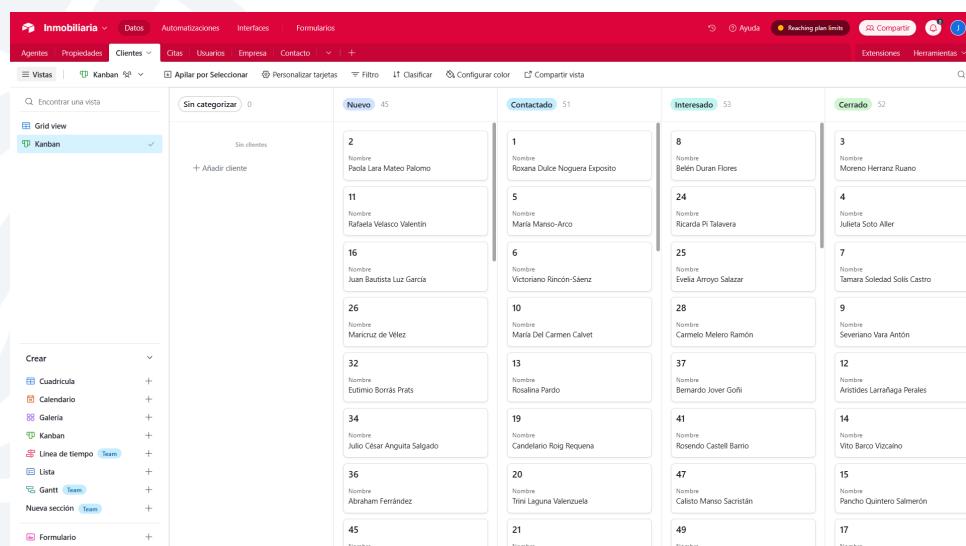
ID Cliente	A. Nombre	Teléfono	Email	Fecha de Registro	Estado	Comentarios	Citas	Interes/Propiedades
183	Álvaro del Pol	670339827	metisapaez@gmail.com	22/5/2025 11:21am	Contactado	Ius cum esse sit.	90	142
184	Candelario Serra Campo	626205010	akloeserna@raza.es	22/5/2025 11:21am	Contactado	Iste eaque unde assumenda inventore corrupti ...	52	1
185	Eva Soenia	670655283	rata10@hotmail.com	22/5/2025 11:21am	Cerrado	Quasi in eos illo nisi minima.	166	144
186	Lorenza Roselló Carulla	626949525	danielmejor@seuna.org	22/5/2025 11:21am	Interesado	Aperiam aut dolorem nulla querat.	97	200
187	Georgina Cadenas Gimeno	690973889	klueter@gmail.com	22/5/2025 11:21am	Interesado	Odio provident enim a detectus tenetur.	40	188
188	Judith Pepita Mayoral Gómez	629907683	iat12@marquez.com	22/5/2025 11:21am	Interesado	Paratur incident accusantium.	96	177
189	Valero Benítez-Domene	663183031	ana50@huertas-sureda.c...	22/5/2025 11:21am	Nuevo	Reprehenderit necessitatibus voluptatibus.	18	21
190	Juan Francisco de Collado	644918995	mayonilla@huarez.com	22/5/2025 11:21am	Nuevo	Ducimus laborum facilis accusamus explicabo qu...	120	125
191	Julie Aurelia Lemna Vives	670693413	segismundoyimez@ofu...	22/5/2025 11:21am	Interesado	Neque possumus inventore dignissimos at.	46	20
192	Micaela Campoy Llanos	672014955	juanmanueloller@yahoo.c...	22/5/2025 11:21am	Interesado	Nihil veniam eius deleniti neque praesentium.	133	177
193	Auzcena Iris Sanz Oliver	611604831	aureareneees@hotmail.com	22/5/2025 11:21am	Interesado	Iure corrupti vitae numquam.		
194	Everisto Torreño Maldonado	636592965	abito70@uina.com	22/5/2025 11:21am	Contactado	Parabon a explicabo.	189	136
195	Amador Iglesias Cervera	688158177	mizas80@gmail.com	22/5/2025 11:21am	Contactado	Quae consequntur quasi inventore.	103	26
196	Espíritu Gilberto Antón	612625970	jose.ramon05@yahoo.com	22/5/2025 11:21am	Interesado	Tempora ullam nisi cupiditate.	154	184
197	Nidia More Cháparro	627286109	elisidio@mascaro.com	22/5/2025 11:21am	Contactado	Necessitatibus occaecati molestias fugiat soluta.		
198	Ariadna Chacón Zurita	615730602	anita57@yahoo.com	22/5/2025 11:21am	Nuevo	Iure vero itaque totam ad quisquam.		
199	René Checa Mate	687025433	edub08@marcos.com	22/5/2025 11:21am	Cerrado	Paratur iure quis maxime saepe assumenda.	117	83
200	Severino Campo-Minguez	613937046	claris75@corazon.com	22/5/2025 11:21am	Cerrado	Maiores natus voluptas consequatur tempora.	14	72
204	Juan Diego Ortega Aranda	612345789	joa102@intumineales.es	25/6/2025 2:57pm	Nuevo	Usuario creado a través del formulario de AirTable.	206	51

Figura 4.38: Vista de Cuadrícula (Grid View) de Clientes en AirTable

La vista de cuadrícula (Grid View), mostrada en la Figura 4.38, permite una manipulación directa de campos como Nombre, Email, Teléfono y Preferencias de vivienda. Se utiliza el Formulario "Nuevo Cliente" para el registro estructurado de nuevos clientes.

#### b) Vista Kanban de Clientes

Una vista Kanban especializada (visible en la Figura 4.39) organiza los registros de clientes en columnas basadas en el campo Seleccionar (un campo de selección única que clasifica al cliente por su estado en el proceso de ventas: "Nuevo", "Contactado", "Interesado", "Cerrado" o "Perdido").



The screenshot shows a Kanban board with the following categories:

- Sin categorizar**: Contains 0 items.
- Nuevo**: Contains 45 items.
- Contactado**: Contains 51 items.
- Interesado**: Contains 53 items.
- Cerrado**: Contains 52 items.

Each category has a list of client names and their IDs.

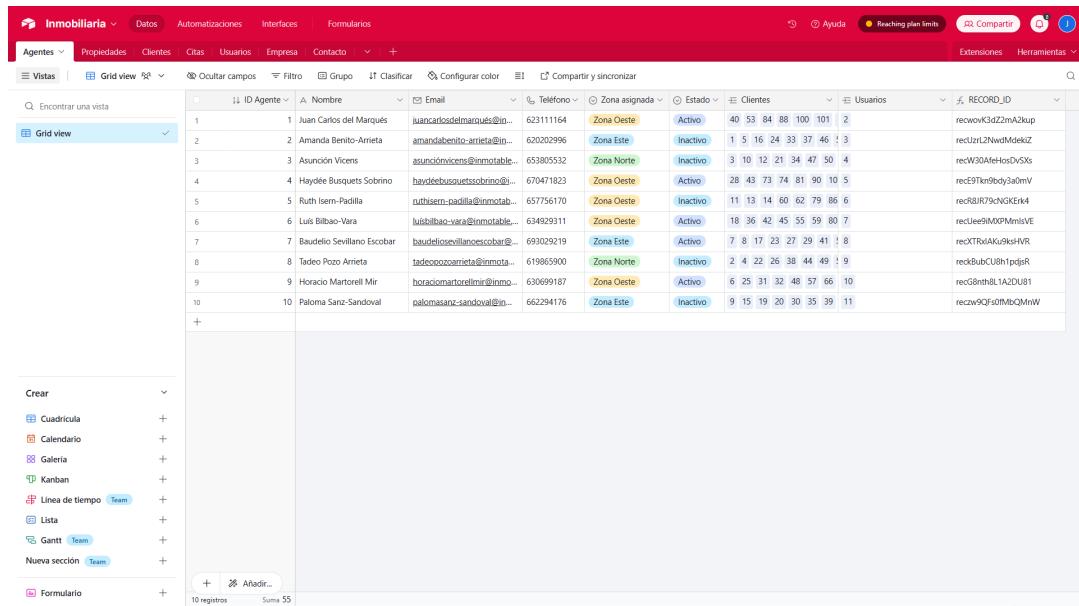
Figura 4.39: Vista Kanban de Clientes en AirTable

#### 4.4. DISEÑO E IMPLEMENTACIÓN

Esta vista permite a los agentes arrastrar y soltar clientes entre las diferentes etapas, ofreciendo una representación visual del embudo de ventas y el progreso de la relación con el cliente.

##### c) Gestión de la Base de Datos de Agentes (Grid View)

Centraliza la información detallada de los agentes inmobiliarios, permitiendo a los administradores gestionar al equipo y sus asignaciones.



ID	ID Agente	Nombre	Email	Teléfono	Zona asignada	Estado	Clients	Usuarios	Record_ID
1	1	Juan Carlos del Marqués	juancarlosdelmarques@... 62311164		Zona Oeste	Activo	40 53 84 88 100 101	2	recwovK3dZ2mA2kup
2	2	Amanda Benito-Arieta	amandabenito-arieta@... 620200996		Zona Este	Inactivo	1 5 16 24 33 37 46	1 3	recUzr2NwHddekZ
3	3	Asunción Vicenc	asuncionvicenc@immobiliar... 653805532		Zona Norte	Inactivo	3 10 12 21 34 47 50	4	recW30Af6osDvSx6
4	4	Haydeé Busquets Sobrino	haydeebusquessobrino@... 670471823		Zona Oeste	Activo	26 43 73 74 81 90	10 5	rec9TkrnbdyJa0mV
5	5	Ruth Isern-Padilla	ruthiser-n-padilla@immobiliar... 657756170		Zona Oeste	Inactivo	11 13 14 60 62 79	86 6	rec8IR79qNGErk4
6	6	Luis Bilbao-Vara	luisbilbao-vara@immobiliar... 634929311		Zona Oeste	Activo	18 36 42 45 55 59	80 7	recUee9IMQPMMisVE
7	7	Baudilio Sevillano Escobar	baudilosevillanosecobar@... 693029219		Zona Este	Activo	7 8 17 23 27 29	41	recCTR4AUk96shVR
8	8	Tadeo Pozo Arrieta	tadeopozozarrieta@immobiliar... 619865900		Zona Norte	Inactivo	2 4 22 26 38 44	49	rec8ubC08h1pdjsR
9	9	Horacio Martorell Mir	horaciomartorellmir@imm... 630699187		Zona Oeste	Activo	6 25 31 32 48	57 66	10
10	10	Paloma Sanz-Sandoval	palomasanz-sandoval@... 662294176		Zona Este	Inactivo	9 15 19 20 30 35	39	11

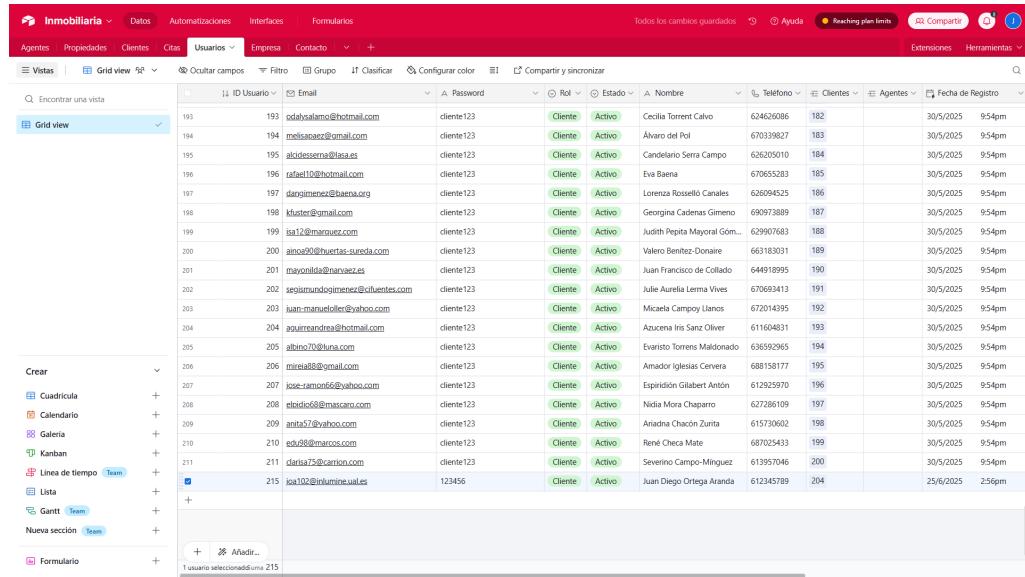
Figura 4.40: Vista de Cuadrícula (Grid View) de Agentes en AirTable

La tabla Agentes en Airtable (visible en la Figura 4.40) es la herramienta para la gestión de este personal. Muestra campos como Nombre del Agente, Email, Teléfono, DNI, y Empresa a la que pertenecen. También incluye campos vinculados a las Propiedades y Citas que tienen asignadas, ofreciendo una visión consolidada de su carga de trabajo. Se utiliza el Formulario "Nuevo Agente" para registrar nuevos agentes.

##### d) Gestión de la Base de Datos de Usuarios (Grid View)

Permite administrar las credenciales de acceso y los roles de todos los usuarios registrados en el sistema (clientes de la web, agentes, administradores), garantizando la seguridad y el control de acceso. Corresponde al requisito funcional **RF 6.1**.

#### 4.4. DISEÑO E IMPLEMENTACIÓN



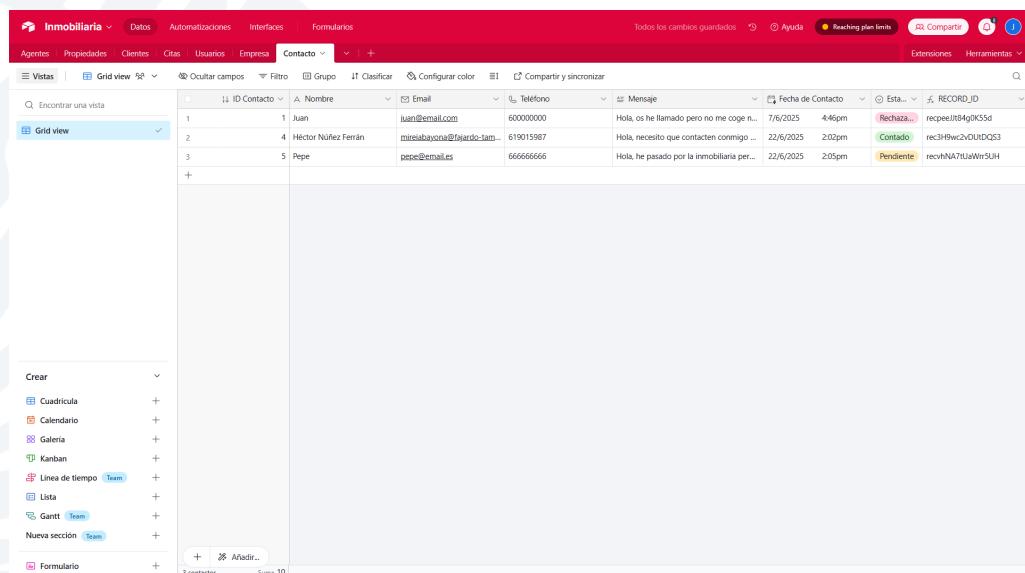
ID Usuario	Email	Password	Rol	Estado	Nombre	Teléfono	Clientes	Agentes	Fecha de Registro
193	cobalysalmo@hotmail.com	cliente123	Cliente	Activo	Cecilia Torrem Calvo	624603086	182		30/5/2025 9:54pm
194	melisnapez@gmail.com	cliente123	Cliente	Activo	Álvaro del Pol	670339827	183		30/5/2025 9:54pm
195	alidesserna@tasa.es	cliente123	Cliente	Activo	Candelario Sierra Campo	626205010	184		30/5/2025 9:54pm
196	rafael10@hotmail.com	cliente123	Cliente	Activo	Eva Baena	670655283	185		30/5/2025 9:54pm
197	dangimenez@seana.org	cliente123	Cliente	Activo	Lorena Roselló Canales	626094625	186		30/5/2025 9:54pm
198	fkuster@gmail.com	cliente123	Cliente	Activo	Georgina Cadena Gimeno	690973889	187		30/5/2025 9:54pm
199	sa12@marquez.com	cliente123	Cliente	Activo	Judith Pepita Mayoral Gómez	629907683	188		30/5/2025 9:54pm
200	ancho90@herreras-sureda.com	cliente123	Cliente	Activo	Valero Benítez-Donaire	661183031	189		30/5/2025 9:54pm
201	mayonista@narvaez.es	cliente123	Cliente	Activo	Juan Francisco de Collado	644919995	190		30/5/2025 9:54pm
202	seguimundojimenez@cifuentes.com	cliente123	Cliente	Activo	Julie Aurelia Lemar Vives	670693413	191		30/5/2025 9:54pm
203	juanmanueloller@yahoo.com	cliente123	Cliente	Activo	Micaela Campoy Iñáñez	672014395	192		30/5/2025 9:54pm
204	aguereandrea@hotmail.com	cliente123	Cliente	Activo	Azuquiza Iris Sanz Oliver	611604831	193		30/5/2025 9:54pm
205	alberto70@luana.com	cliente123	Cliente	Activo	Evaristo Torres Maldonado	636599265	194		30/5/2025 9:54pm
206	mireya88@gmail.com	cliente123	Cliente	Activo	Amador Iglesias Cervera	688158177	195		30/5/2025 9:54pm
207	kose.ramone6@yahoo.com	cliente123	Cliente	Activo	Espíritu Gilberto Antón	612925970	196		30/5/2025 9:54pm
208	eladiojim@mascaro.com	cliente123	Cliente	Activo	Nidia Mora Chaparro	627286109	197		30/5/2025 9:54pm
209	anita57@yahoo.com	cliente123	Cliente	Activo	Anaída Chacón Zurita	615730602	198		30/5/2025 9:54pm
210	edu@88marcos.com	cliente123	Cliente	Activo	René Checa Mate	687025433	199		30/5/2025 9:54pm
211	darits5@cerion.com	cliente123	Cliente	Activo	Severino Campo-Minguez	613957046	200		30/5/2025 9:54pm
215	iso102@inlumine.us.es	123456	Cliente	Activo	Juan Diego Ortega Aranda	612345789	204		25/6/2025 2:56pm

Figura 4.41: Vista de Cuadrícula (Grid View) de Usuarios en AirTable

La tabla Usuarios en Airtable (visible en la Figura 4.41) es la herramienta principal para que los administradores controlen estos registros. Muestra campos como Nombre de Usuario, Email, Contraseña (hash), y Rol. Incluye campos vinculados a Agentes y Clientes para asociar las cuentas de acceso con los perfiles de negocio correspondientes.

#### e) Gestión de la Base de Datos de Contacto (Grid View)

Los agentes y administradores utilizan la tabla Citas en Airtable como su principal herramienta para visualizar las consultas de los usuarios anónimos y/o de los clientes registrados.



ID Contacto	Nombre	Email	Teléfono	Mensaje	Fecha de Contacto	Estado	RECORD_ID
1	Juan	juan@email.com	600000000	Hola, os he llamado pero no me coge n... 7/6/2025 4:46pm		Rechazada	rech1J84qK55d
2	Héctor Núñez Ferrán	mirabolavaya@fajardo.com	619015987	Hola, necesito que contacten conmigo ... 22/6/2025 2:02pm		Contado	rech3H9wczv0UDQG3
3	Pepe	pere@email.es	666666666	Hola, he pasado por la inmobiliaria per... 22/6/2025 2:05pm		Pendiente	rechNA7tUWr5UH

Figura 4.42: Vista de Cuadrícula (Grid View) de Clientes en AirTable

## 4.4. DISEÑO E IMPLEMENTACIÓN

La vista de cuadrícula (Grid View), mostrada en la Figura 4.42, permite una manipulación directa de campos como Nombre, Email, Teléfono, Mensaje y Estado (Oendiente por defecto). Una vez el Agente haya contactado con el usuario, pondrá el Estado en Contactado o Rechazado.

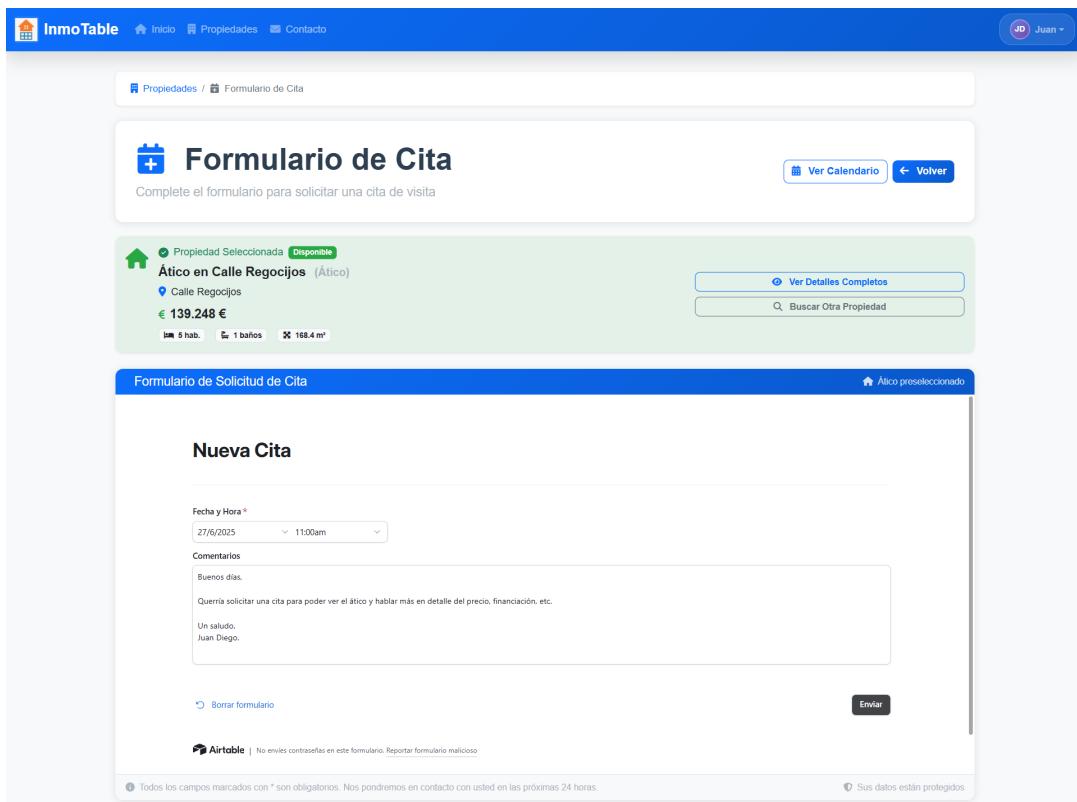
### 4.4.2.4 Módulo: Gestión de Citas

Este módulo abarca la solicitud de citas por parte de los clientes y la gestión completa de la agenda por los agentes.

#### 1. Interfaz para el Cliente (Aplicación Web)

##### a) Formulario de Solicitud de Cita

Permite a los clientes (principalmente clientes registrados) solicitar una cita para visitar una propiedad o para otras consultas. Corresponde al requisito funcional **RF 1.7.**



**Figura 4.43:** Solicitud de Cita en Aplicación Web

Este formulario (basado en la Figura 4.7 del mockup y visible en Figura 4.43) se ha implementado incrustando un formulario de Airtable. Se configura para que cargue por defecto la propiedad de interés (pasada por parámetro: `prefill _ Propiedad = Record _ ID` en la URL del formulario de Airtable) y el cliente logueado (pasado por parámetro: `prefill _ Cliente = Record _ ID` en la URL), ocultando estos campos para el usuario final. Esto asegura que la cita esté directamente vinculada a la propiedad y al cliente correcto en la tabla Citas de Airtable, con campos visibles para la fecha, hora, y tipo de cita.

The screenshot shows a web-based form titled "Nueva Cita". At the top, there are buttons for "Añadir" (Add), "Editar" (Edit), "Ver datos" (View data), and "Compartir" (Share). The form fields include:

- Fecha y Hora \***: A date and time input field.
- Estado**: A dropdown menu showing "Pendiente" (Pending).
- Comentarios**: A text area for comments.
- Propiedad \***: A dropdown menu showing "51 Taulo" with the note "Atico en Calle Regocijos". There is also a "+ Añadir propiedad" (Add property) button.
- Cliente \***: A dropdown menu showing "204 Joan Diego Ortega Ar..." with the note "Teléfono: 612345789". There is also a "+ Añadir cliente" (Add client) button.
- Borrar formulario**: A link to clear the form.
- Enviar**: A dark blue "Send" button.

At the bottom left, there is a small "Airtable" logo and a note: "No envíes contraseñas en este formulario. Reportar formulario malicioso".

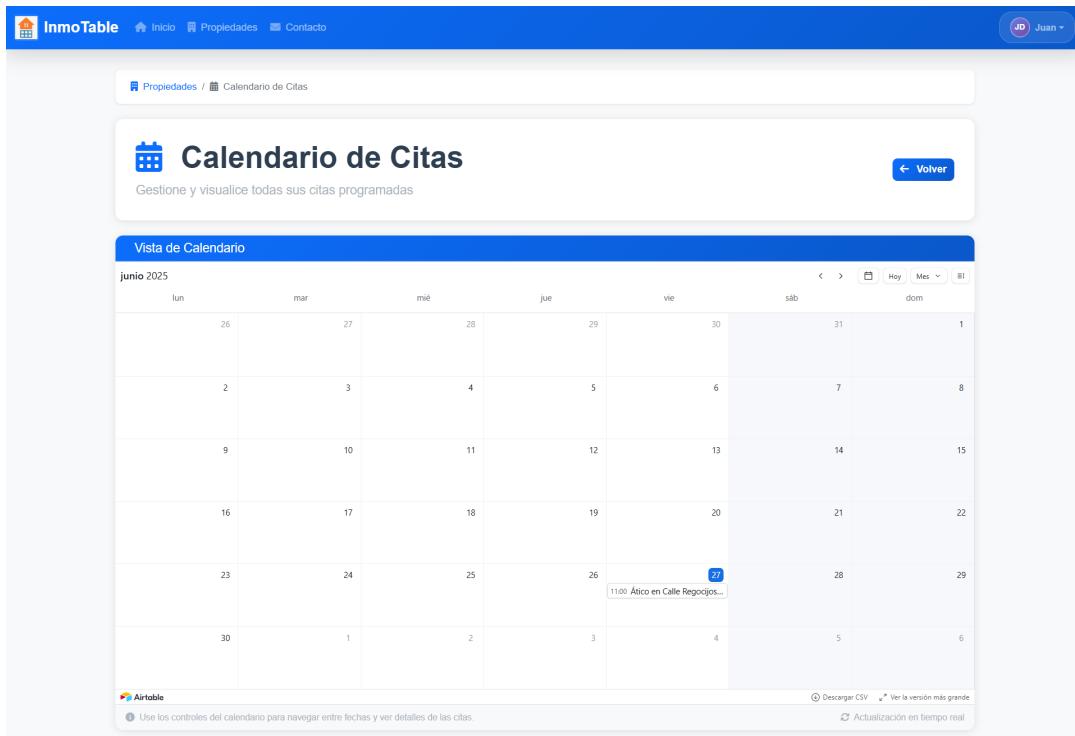
**Figura 4.44:** Vista de Cuadrícula (Grid View) de Citas en AirTable

La Figura 4.44 ilustra la configuración de este formulario en Airtable.

**b) Visualización del Calendario de Citas (Cliente)**

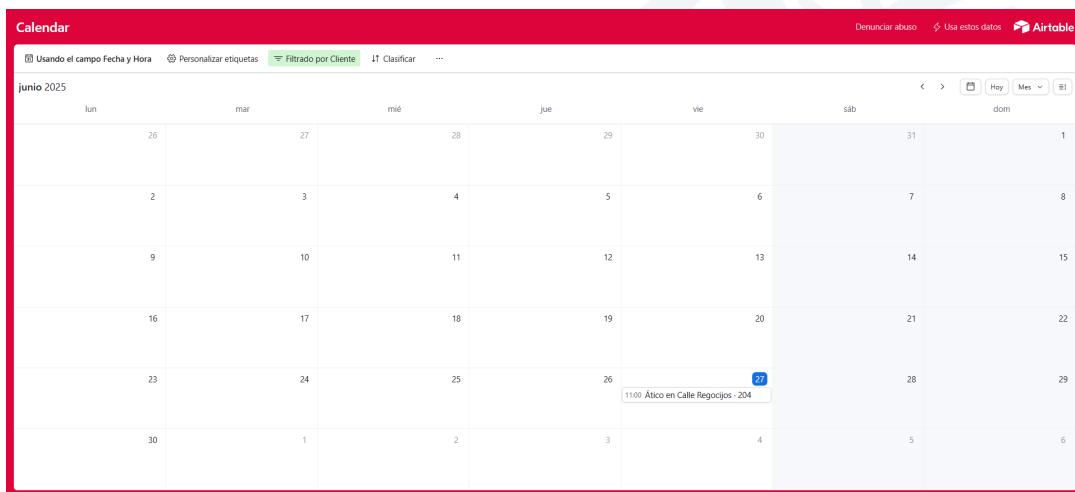
Ofrece a los clientes registrados una vista personalizada de sus citas programadas, permitiéndoles tener un seguimiento visual de sus interacciones con la inmobiliaria.

#### 4.4. DISEÑO E IMPLEMENTACIÓN



**Figura 4.45:** Calendario de Citas en AirTable

Esta funcionalidad se implementa incrustando una vista de calendario de Airtable (appointment-calendar.component), la cual es filtrada dinámicamente mediante un parámetro en la URL de Airtable (filter \_ Cliente = ID \_ Cliente). Esto asegura que cada cliente vea únicamente sus propias citas. La Figura 4.45 muestra la interfaz del calendario en la aplicación web.



**Figura 4.46:** Vista de Calendario de Citas en AirTable

Y la Figura 4.46 ilustra la configuración de este componente en Airtable.

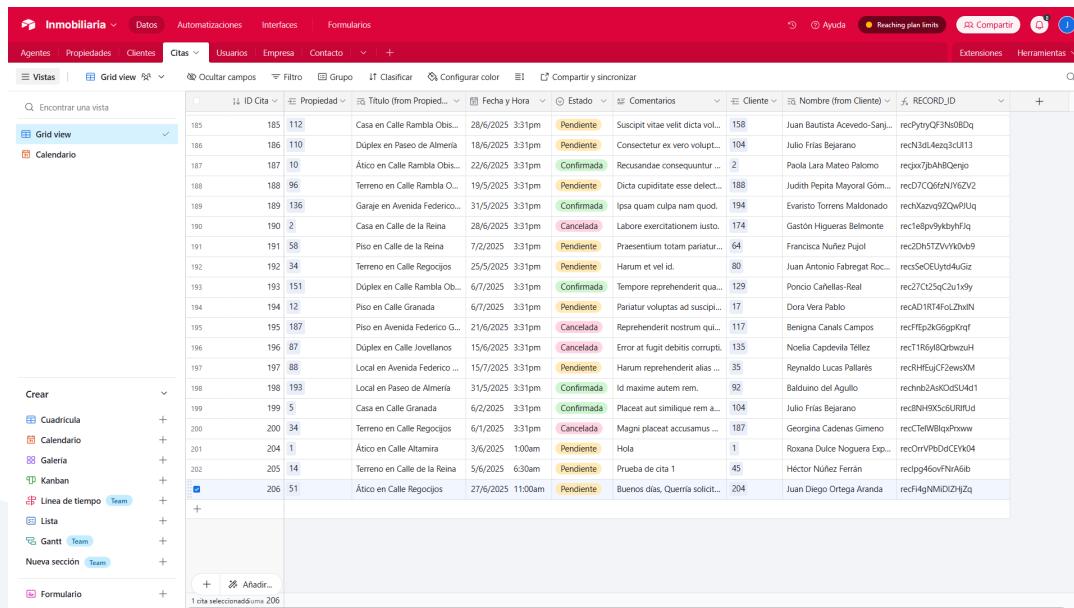
## 4.4. DISEÑO E IMPLEMENTACIÓN

### 2. Interfaz para el Agente/Administrador (Airtable)

Permitir a los agentes y administradores programar, modificar, dar seguimiento y visualizar todas las citas (de todos los clientes) de manera eficiente, gestionando su agenda y el flujo de interacciones. Corresponde a los requisitos funcionales **RF 3.1**, **RF 3.2** y **RF 3.3**.

#### a) Gestión de la Base de Datos de Citas (Grid View)

Los agentes y administradores utilizan esta vista de la tabla **Citas** en Airtable como la base para una gestión detallada.



ID Cita	Propiedad	Título (from Propiedad)	Fecha y Hora	Estado	Comentarios	Cliente	Nombre (from Cliente)	RECORD_ID
185	185	112	Casa en Calle Rambla Obis...	28/6/2025 3:31pm	Pendiente	Suscipit vitae velit dicta vol...	158	Juan Bautista Acevedo-San...
186	186	110	Dúplex en Paseo de Almeria...	18/6/2025 3:31pm	Pendiente	Consecetur ex vero voluptu...	104	Julio Fries Bejarano
187	187	10	Ático en Calle Rambla Obis...	22/6/2025 3:31pm	Confirmada	Recusandae consequuntur ...	2	Paola Lara Mateo Palomo
188	188	96	Terreno en Calle Rambla O...	19/5/2025 3:31pm	Pendiente	Dicta cupiditate esse detect...	188	Judith Pepey Mayoral Gom...
189	189	136	Garaje en Avenida Federico...	31/5/2025 3:31pm	Confirmada	Ipsa quam culpa nam quod...	194	Evaristo Torrens Maldonado
190	190	2	Casa en Calle de la Reina	28/6/2025 3:31pm	Cancelada	Labore exercitationem iusto...	174	Gastón Higueras Belmonte
191	191	58	Piso en Calle de la Reina	7/2/2025 3:31pm	Pendiente	Præsentium totam paratur...	64	Francisco Núñez Pujol
192	192	34	Terreno en Calle Regocjos	25/5/2025 3:31pm	Pendiente	Harm et vel id.	80	Juan Antonio Fabregat Roc...
193	193	151	Dúplex en Calle Rambla Ob...	6/7/2025 3:31pm	Confirmada	Tempore reprehenderit qua...	129	Poncio Cañellas-Real
194	194	12	Piso en Calle Granada	6/7/2025 3:31pm	Pendiente	Paratur voluptatis ad suscipi...	17	Dora Vera Pablo
195	195	187	Piso en Avenida Federico G...	21/6/2025 3:31pm	Cancelada	Reprehenderit nostrum qua...	117	Benigna Canals Campos
196	196	87	Dúplex en Calle Jovellanos	15/6/2025 3:31pm	Cancelada	Error at fugit debitis corrupt...	135	Noelia Capdevila Téllez
197	197	88	Local en Avenida Federico ...	15/7/2025 3:31pm	Pendiente	Harum reprehenderit alias ...	35	Reinaldo Lucas Pallares
198	198	193	Local en Paseo de Almería	31/5/2025 3:31pm	Confirmada	Id maxime autem rem...	92	Baldomero del Agullo
199	199	5	Casa en Calle Granada	6/7/2025 3:31pm	Confirmada	Placeat aut similique rem a...	104	Julio Fries Bejarano
200	200	34	Terreno en Calle Regocjos	6/1/2025 3:31pm	Cancelada	Magni placeat accusamus ...	187	Georgina Cadenas Gimeno
201	201	1	Ático en Calle Altamira	3/6/2025 1:00am	Pendiente	Hola	1	Roxana Dulce Noguera Exp...
202	202	14	Terreno en Calle de la Reina	5/6/2025 6:30am	Pendiente	Prueba de cita 1	45	Héctor Núñez Ferrán
203	203	51	Ático en Calle Regocjos	27/6/2025 11:00am	Pendiente	Buenos días, quería solicitar...	204	Juan Diego Ortega Aranda

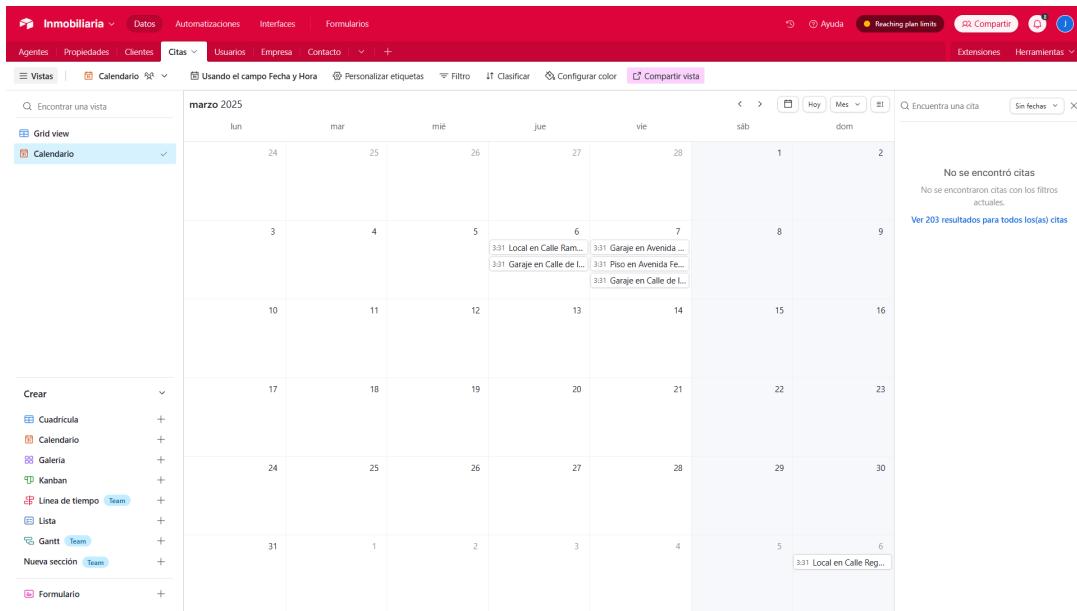
Figura 4.47: Vista de Cuadrícula (Grid View) de Citas en AirTable

La vista de cuadrícula (Grid View), mostrada en la Figura 4.47, permite una visualización tabular de todas las citas con campos como "Fecha de la Cita", "Cliente", "Agente", "Propiedad", "Tipo de Cita" y "Estado". Desde aquí, pueden editar directamente cualquier detalle de la cita o acceder al formulario de detalles.

#### b) Vista de Calendario (para Agentes/Administradores)

Los agentes y administradores también utilizan la Vista de Calendario de Airtable (visible en las Figura 4.48) para una visión global de todas las citas de la inmobiliaria.

#### 4.4. DISEÑO E IMPLEMENTACIÓN



**Figura 4.48:** Vista de Calendario de Citas en AirTable

Esta vista les permite ver la agenda completa, filtrar por agente o por propiedad, y arrastrar/soltar citas para reprogramarlas, optimizando la gestión de tiempo de todo el equipo.

##### 4.4.2.5 Módulo: Análisis de Negocio

Este módulo es fundamentalmente para agentes y administradores, y se gestiona íntegramente a través de las potentes interfaces de Airtable, proporcionando vistas clave del negocio.

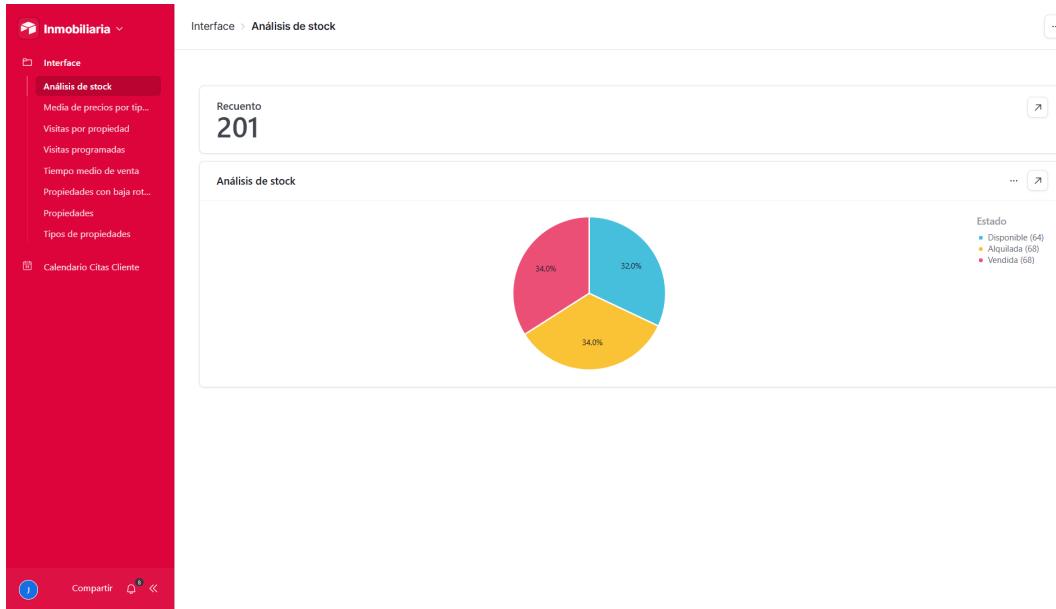
El propósito es transformar los datos brutos en información accionable para la toma de decisiones estratégicas, permitiendo a los usuarios internos monitorear el rendimiento, identificar tendencias y áreas de mejora en el negocio inmobiliario. Corresponde a los requisitos funcionales **RF 5.1, RF 5.2, RF 5.3, RF 5.4, RF 5.5 y RF 5.6**.

A continuación se detallan todas las interfaces creadas en Airtable.

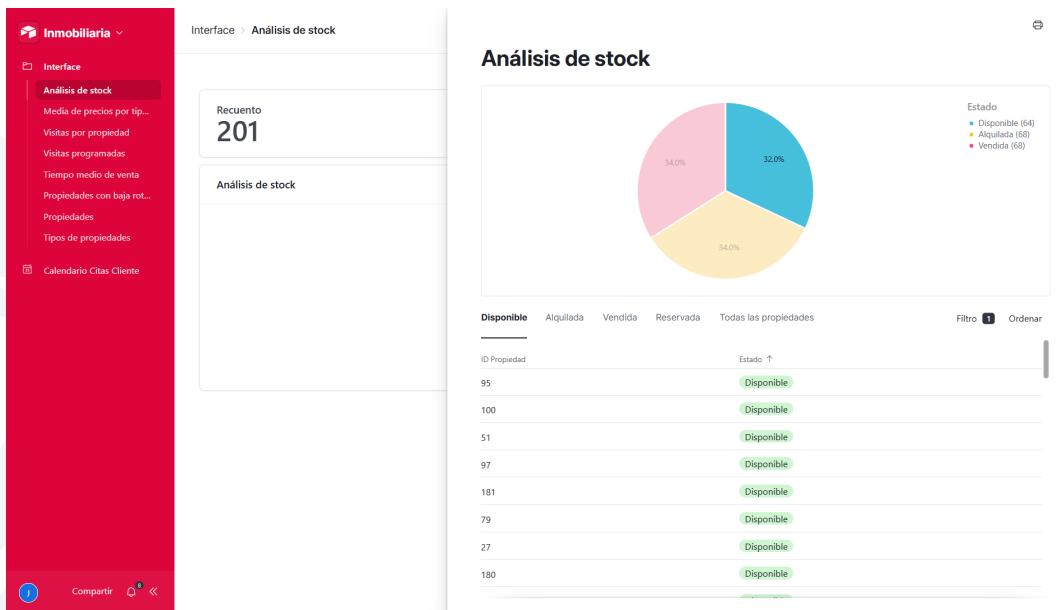
##### 1. Análisis de Stock

En las Figuras 4.49 y 4.50 muestra cuadros de mando con métricas agregadas (total de propiedades, disponibles, precio medio, superficie media) y gráficos de distribución de propiedades por tipo y estado, ofreciendo una visión ejecutiva del inventario.

#### 4.4. DISEÑO E IMPLEMENTACIÓN



**Figura 4.49:** Interfaz de Análisis de Stock en AirTable

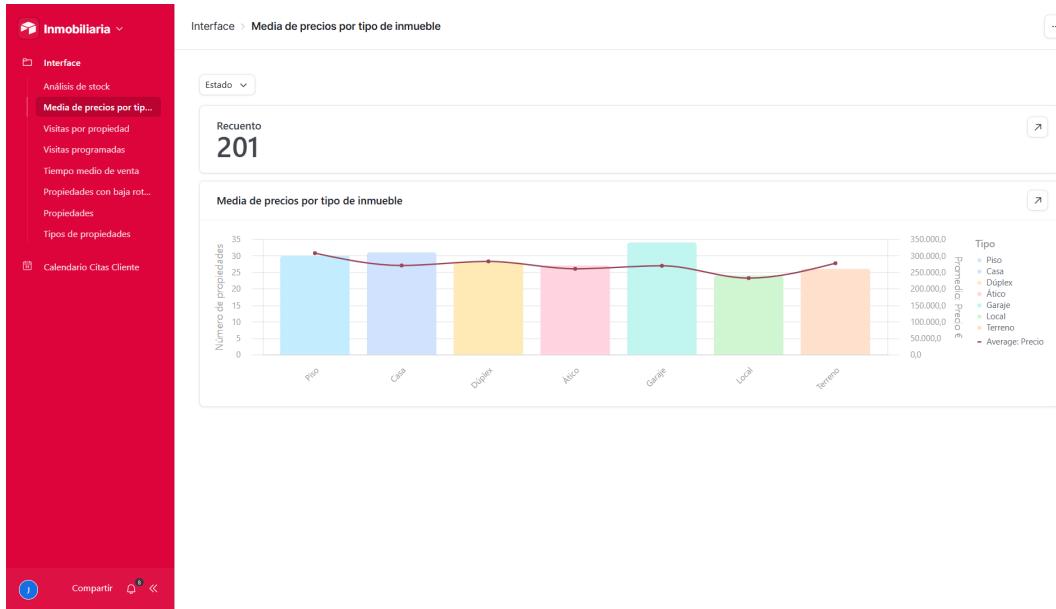


**Figura 4.50:** Detalle de Interfaz de Análisis de Stock en AirTable

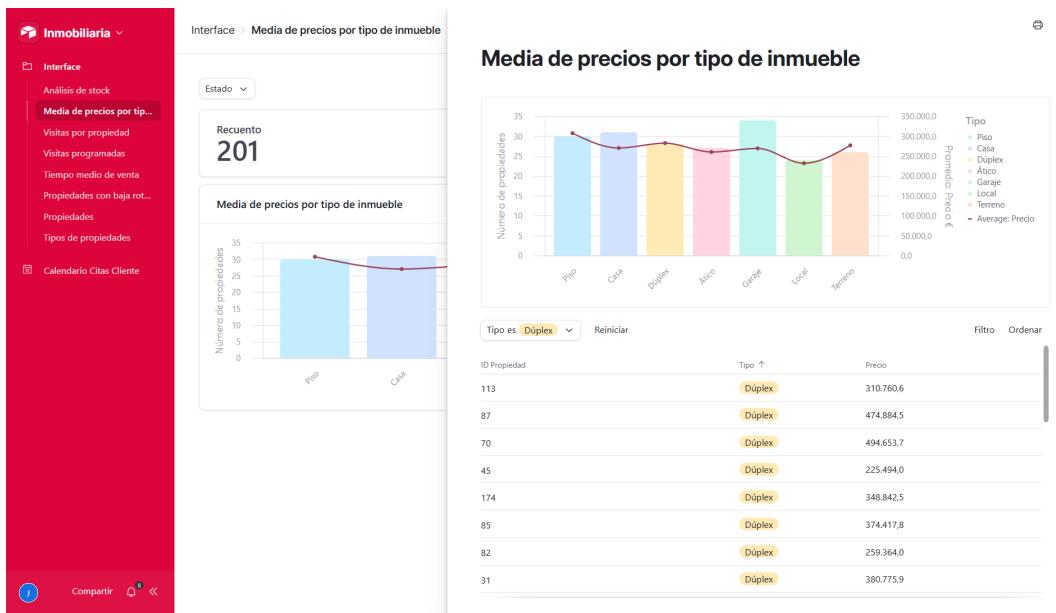
## 2. Media de Precios por Tipo de Inmueble

En las Figuras 4.51, 4.52 y 4.53 muestra gráficos de barras que comparan los precios promedio entre diferentes tipos de propiedades (ej. apartamentos vs. casas), facilitando el análisis de tendencias de mercado y la valoración.

#### 4.4. DISEÑO E IMPLEMENTACIÓN



**Figura 4.51:** Interfaz de Media de Precios por Tipo de Inmueble en AirTable



**Figura 4.52:** Detalle de Interfaz de Media de Precios por Tipo de Inmueble en AirTable

#### 4.4. DISEÑO E IMPLEMENTACIÓN

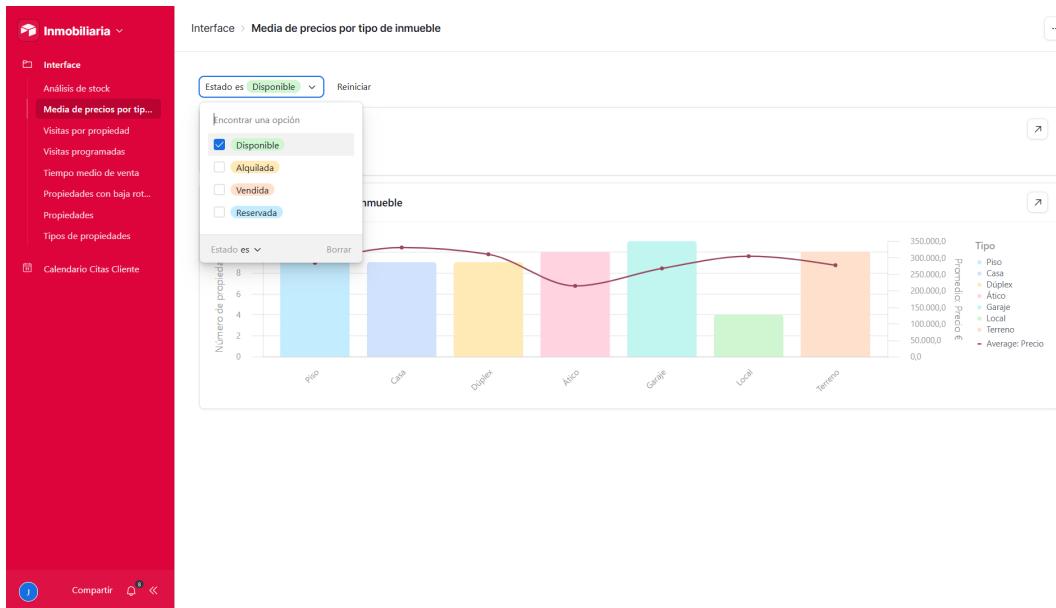


Figura 4.53: Filtro de Interfaz de Media de Precios por Tipo de Inmueble en AirTable

### 3. Visitas por Propiedad

En las Figuras 4.54 y 4.55 muestra listados y contadores que muestran el interés generado por cada inmueble (número de citas asociadas), permitiendo identificar las propiedades más demandadas.

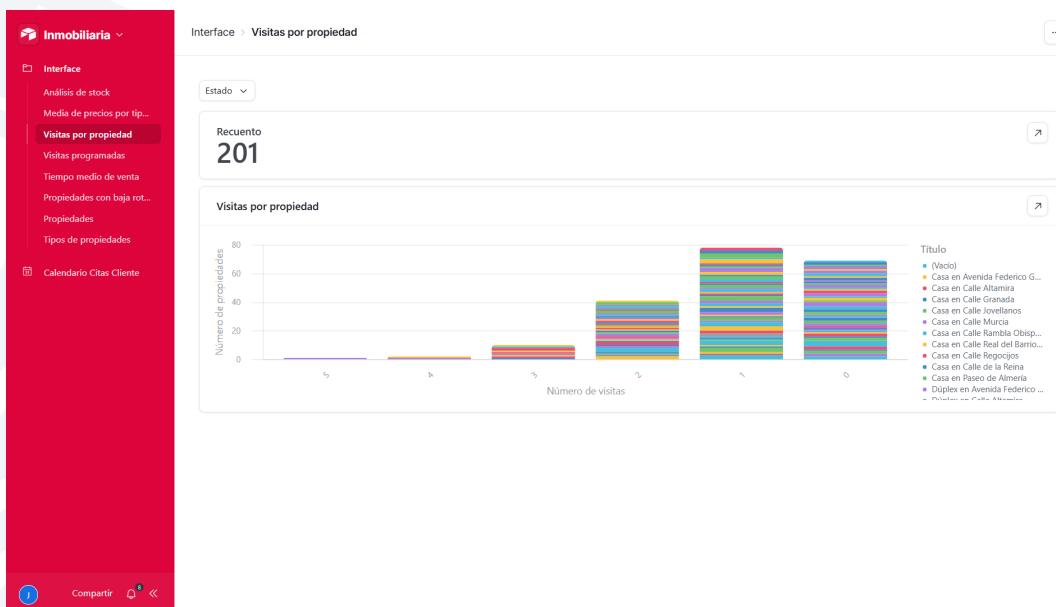


Figura 4.54: Interfaz de Visitas por Propiedad en AirTable

#### 4.4. DISEÑO E IMPLEMENTACIÓN



Figura 4.55: Detalle de Interfaz de Visitas por Propiedad en AirTable

#### 4. Visitas Programadas

En las Figuras 4.56, 4.57 y 4.58 muestra resúmenes y gráficos de citas por estado y día, complementando la vista de calendario para una comprensión rápida de la carga de trabajo y el progreso de las citas.

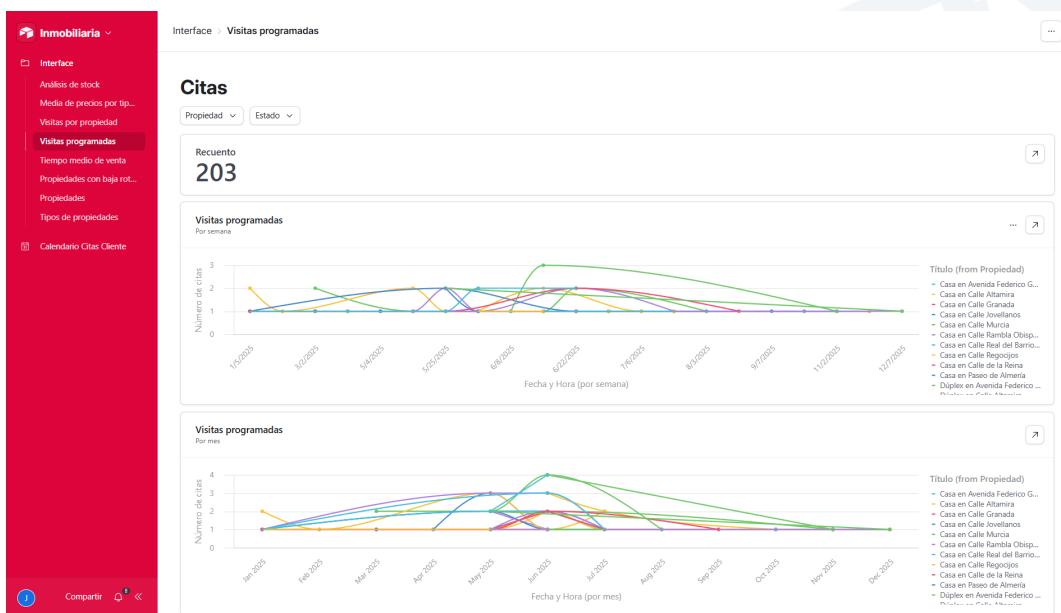


Figura 4.56: Interfaz de Visitas Programadas en AirTable

#### 4.4. DISEÑO E IMPLEMENTACIÓN

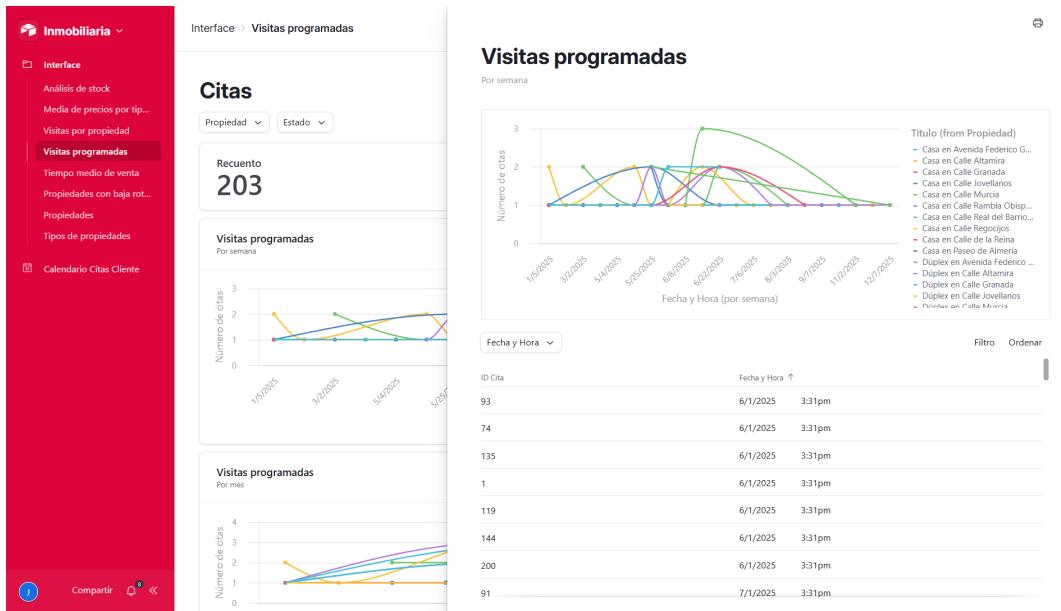


Figura 4.57: Detalle por semana de Interfaz de Visitas Programadas en AirTable

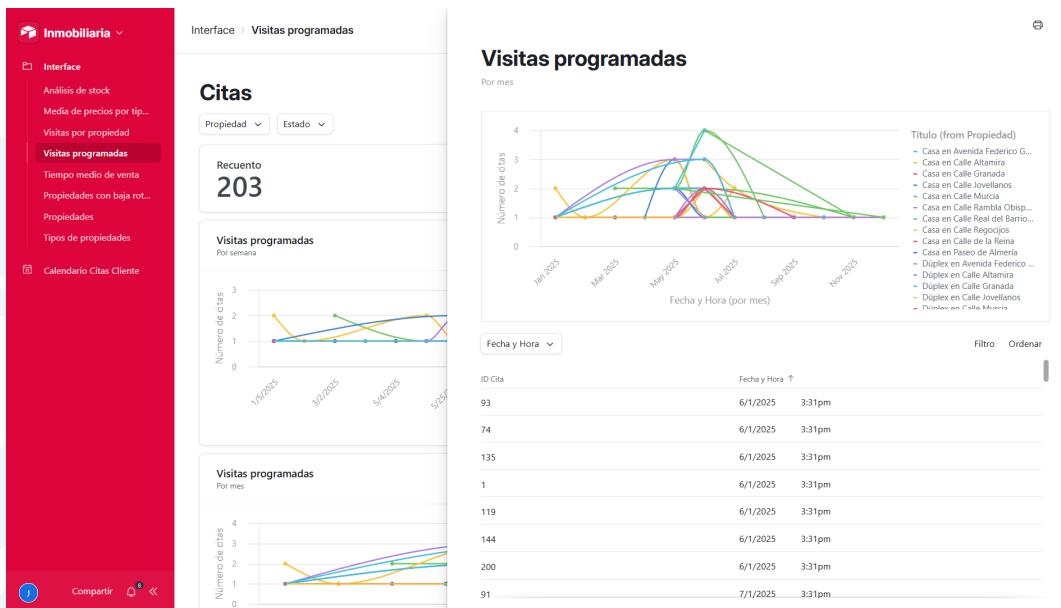


Figura 4.58: Detalle por mes de Interfaz de Visitas Programadas en AirTable

#### 5. Tiempo Medio de Venta

En las Figuras 4.59, 4.60 y 4.61 muestra métricas y gráficos sobre la eficiencia en la venta o alquiler de propiedades (días en el mercado por tipo de propiedad), lo que ayuda a identificar cuellos de botella.

#### 4.4. DISEÑO E IMPLEMENTACIÓN

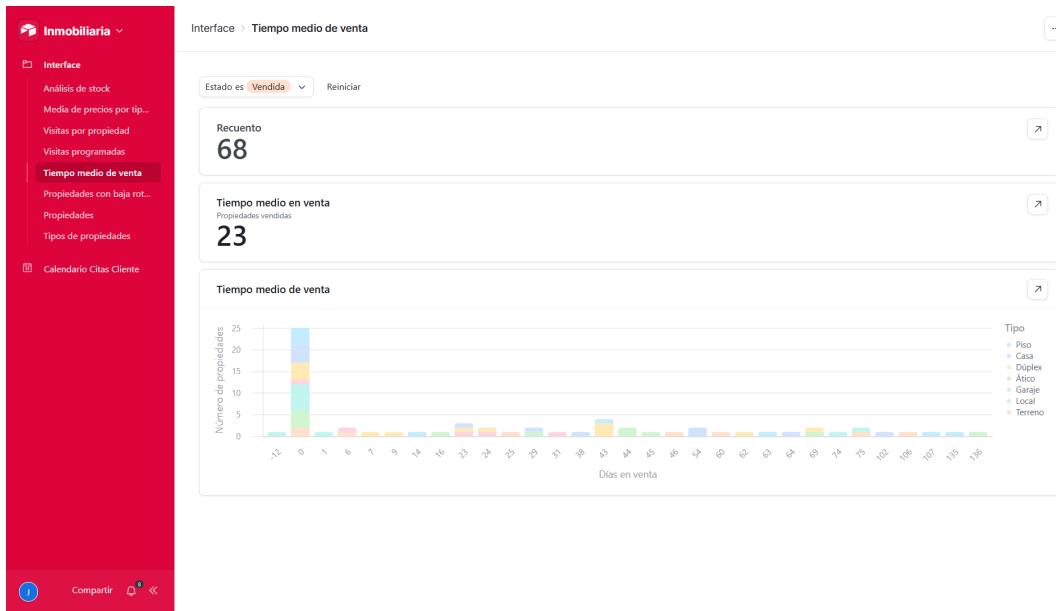


Figura 4.59: Interfaz de Tiempo Medio de Venta en AirTable

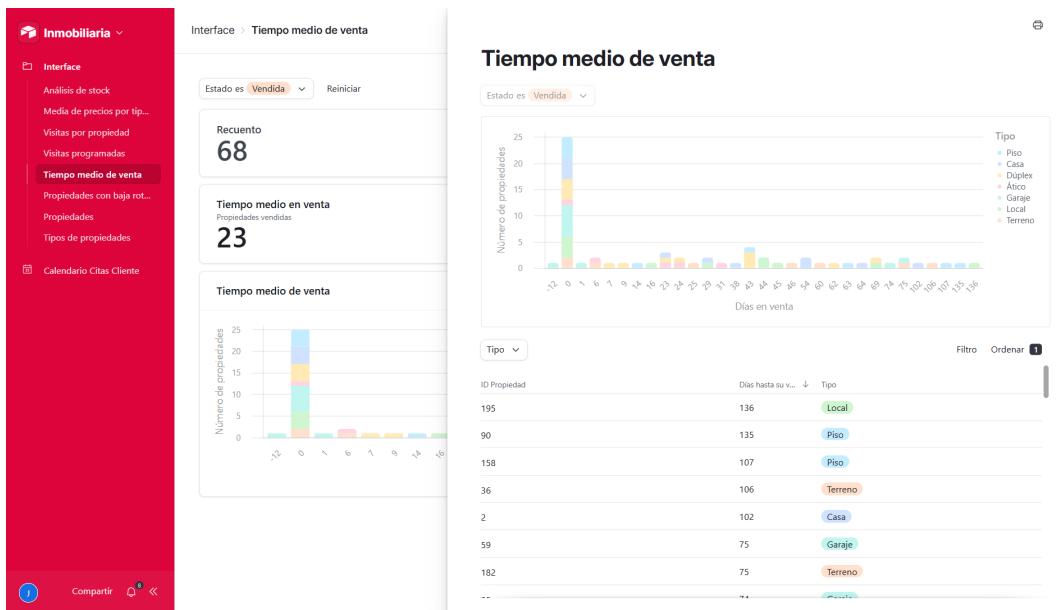


Figura 4.60: Detalle de Interfaz de Tiempo Medio de Venta en AirTable

#### 4.4. DISEÑO E IMPLEMENTACIÓN

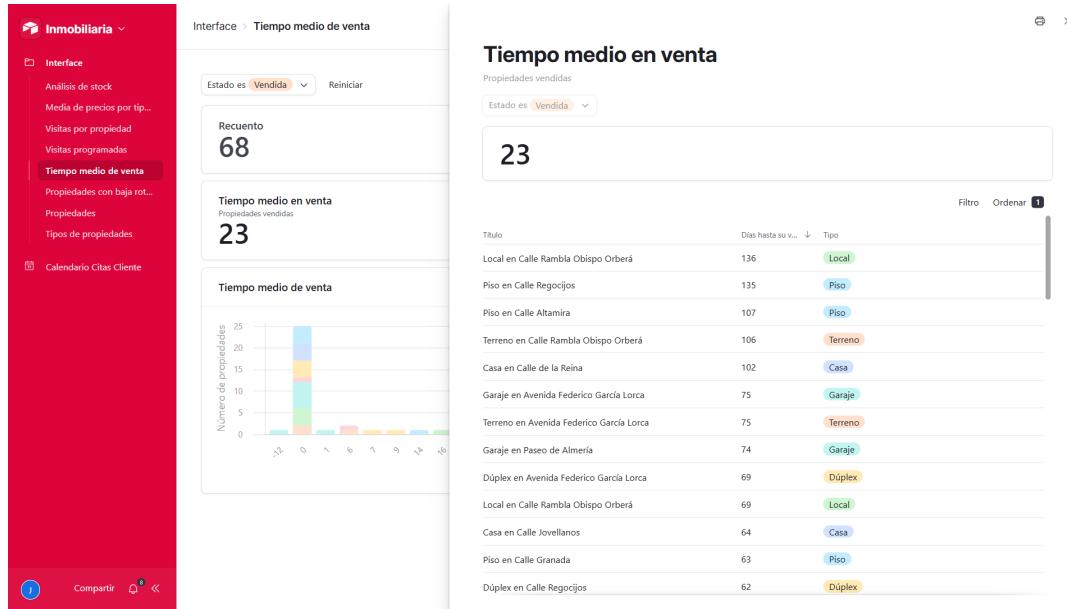


Figura 4.61: Detalle de Interfaz de Tiempo Medio de Venta en AirTable

#### 6. Propiedades con Baja Rotación

En las Figuras 4.62 y 4.63 muestra un listado específico de inmuebles que requieren atención especial por su prolongado tiempo en el mercado, con métricas clave para la gestión proactiva.

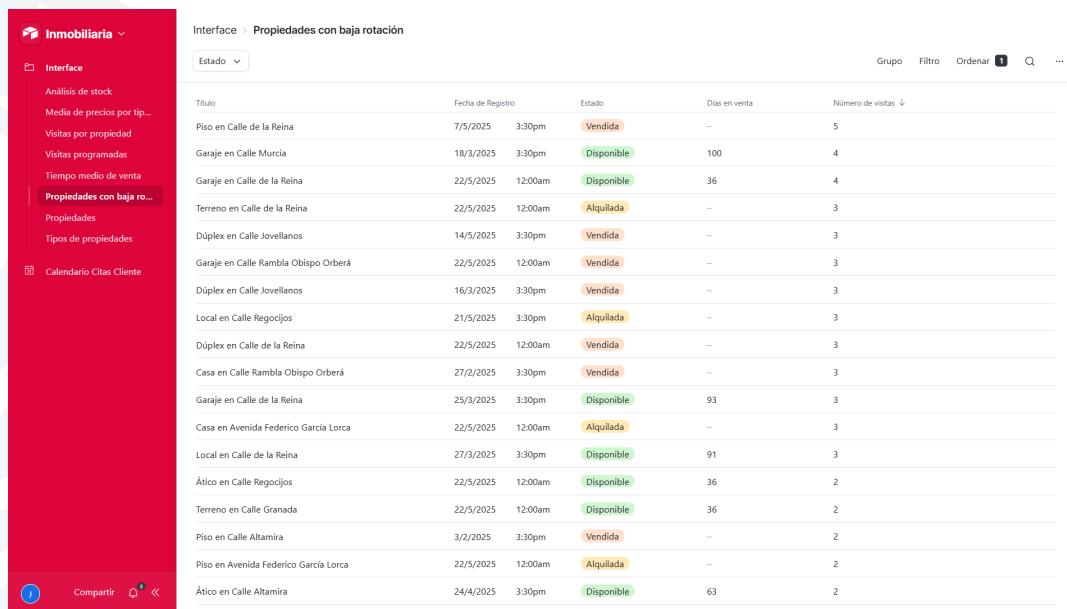
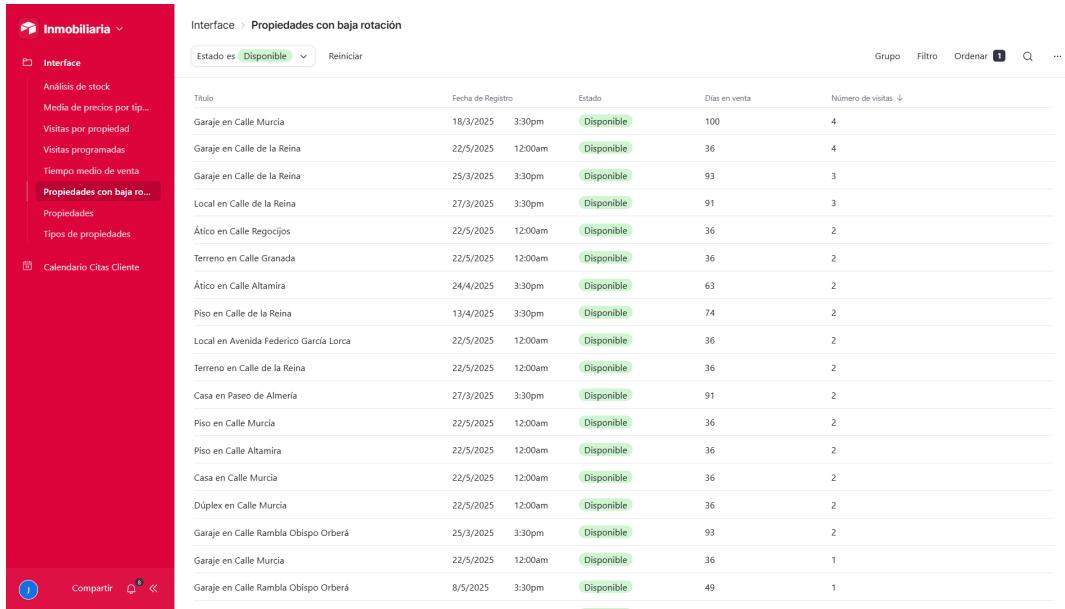


Figura 4.62: Interfaz de Propiedades con Baja Rotación en AirTable

#### 4.4. DISEÑO E IMPLEMENTACIÓN



Título	Fecha de Registro	Estado	Días en venta	Número de visitas
Garaje en Calle Murcia	18/3/2025	3:30pm	Disponible	100
Garaje en Calle de la Reina	22/5/2025	12:00am	Disponible	36
Garaje en Calle de la Reina	25/3/2025	3:30pm	Disponible	93
Local en Calle de la Reina	27/3/2025	3:30pm	Disponible	91
Ático en Calle Regocjos	22/5/2025	12:00am	Disponible	36
Terreno en Calle Granada	22/5/2025	12:00am	Disponible	36
Ático en Calle Altamira	24/4/2025	3:30pm	Disponible	63
Piso en Calle de la Reina	13/4/2025	3:30pm	Disponible	74
Local en Avenida Federico García Lorca	22/5/2025	12:00am	Disponible	36
Terreno en Calle de la Reina	22/5/2025	12:00am	Disponible	36
Casa en Paseo de Almería	27/3/2025	3:30pm	Disponible	91
Piso en Calle Murcia	22/5/2025	12:00am	Disponible	36
Piso en Calle Altamira	22/5/2025	12:00am	Disponible	36
Casa en Calle Murcia	22/5/2025	12:00am	Disponible	36
Dúplex en Calle Murcia	22/5/2025	12:00am	Disponible	36
Garaje en Calle Rambla Obispo Orberá	25/3/2025	3:30pm	Disponible	93
Garaje en Calle Murcia	22/5/2025	12:00am	Disponible	36
Garaje en Calle Rambla Obispo Orberá	8/5/2025	3:30pm	Disponible	49

Figura 4.63: Propiedades Disponibles con Baja Rotación en AirTable

#### 7. Propiedades

En las Figuras 4.64 y 4.65 muestra gráficos visuales que representan la proporción de propiedades en diferentes estados (ej., "Disponible", "Alquilada", "Vendida"). Esto proporciona una visión rápida de la liquidez y el estado general del inventario, complementando los gráficos por tipo de propiedad.

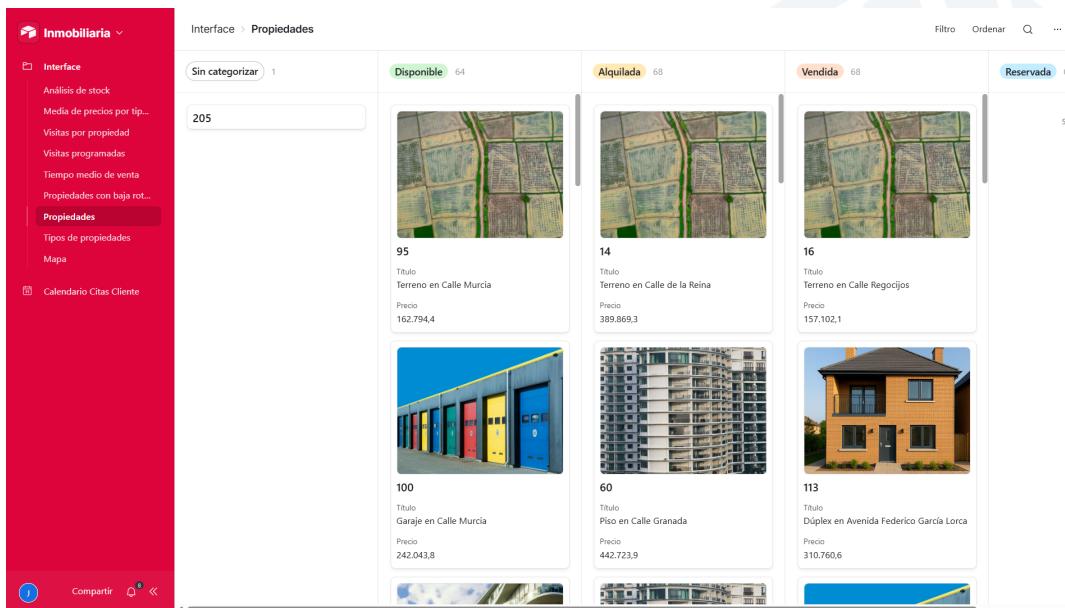
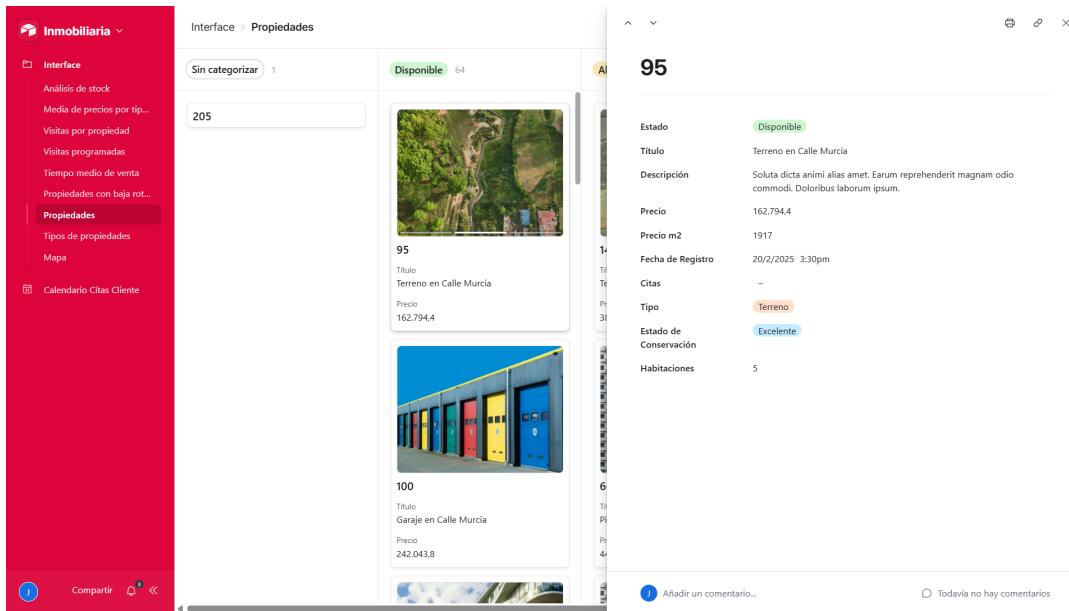


Figura 4.64: Interfaz de Propiedades en AirTable

#### 4.4. DISEÑO E IMPLEMENTACIÓN



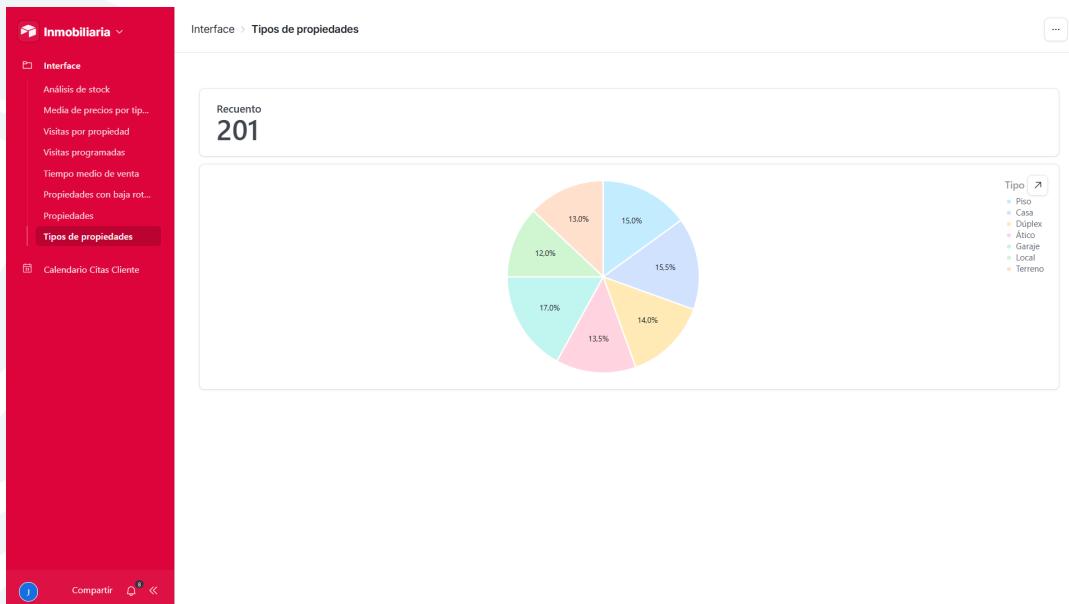
The screenshot shows the AirTable interface for property management. On the left, a sidebar menu for 'Inmobiliaria' includes options like 'Interface', 'Propiedades', and 'Tipos de propiedades'. The main view displays two property cards: one for a plot of land ('Terreno en Calle Murcia') and another for a garage ('Garaje en Calle Murcia'). To the right, a detailed view of a specific property (ID 95) is shown, including its title, description, price, registration date, and condition. A pie chart at the bottom indicates the distribution of property types.

Estado	Disponible
Título	Terreno en Calle Murcia
Descripción	Soluta dicta animi alias amet. Earum reprehenderit magnam odio commodi. Doloribus laborum ipsum.
Precio	162.794,4
Precio m <sup>2</sup>	1917
Fecha de Registro	20/2/2025 3:30pm
Citas	-
Tipo	Terreno
Estado de Conservación	Excelente
Habitaciones	5

**Figura 4.65:** Detalle de Interfaz de Propiedades Disponibles en AirTable

#### 8. Tipo de Propiedades

En las Figuras 4.66 y 4.67 muestra gráficos que muestran la composición del inventario por categorías (ej. Apartamentos, Casas), lo que ayuda a la estrategia de adquisición y venta.



**Figura 4.66:** Interfaz de Tipo de Propiedades en AirTable

#### 4.4. DISEÑO E IMPLEMENTACIÓN

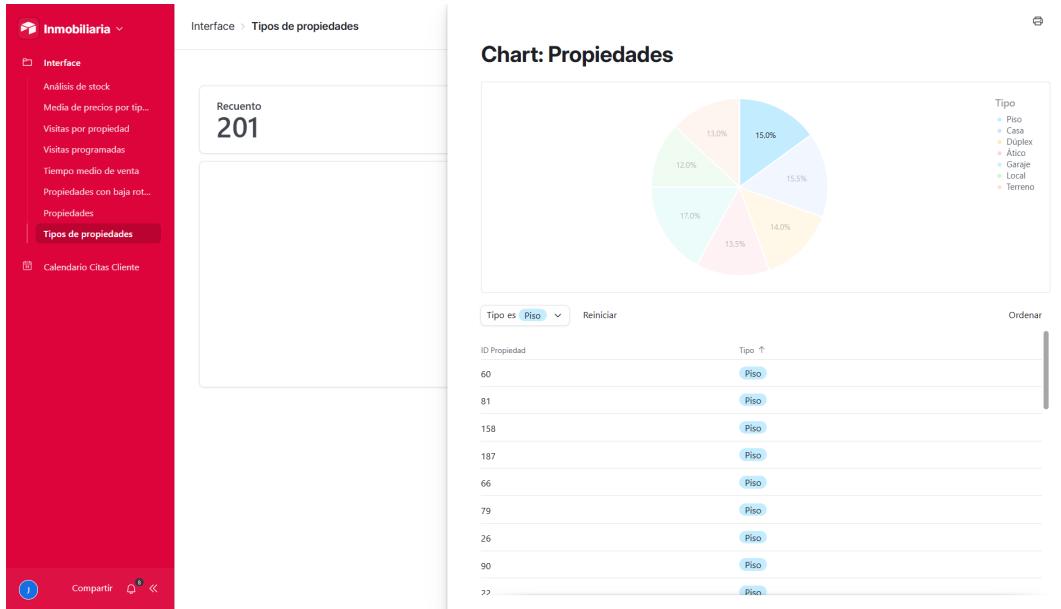


Figura 4.67: Detalle de Interfaz de Tipo de Propiedades en AirTable

##### 4.4.2.6 Módulo: Automatizaciones (Visión General)

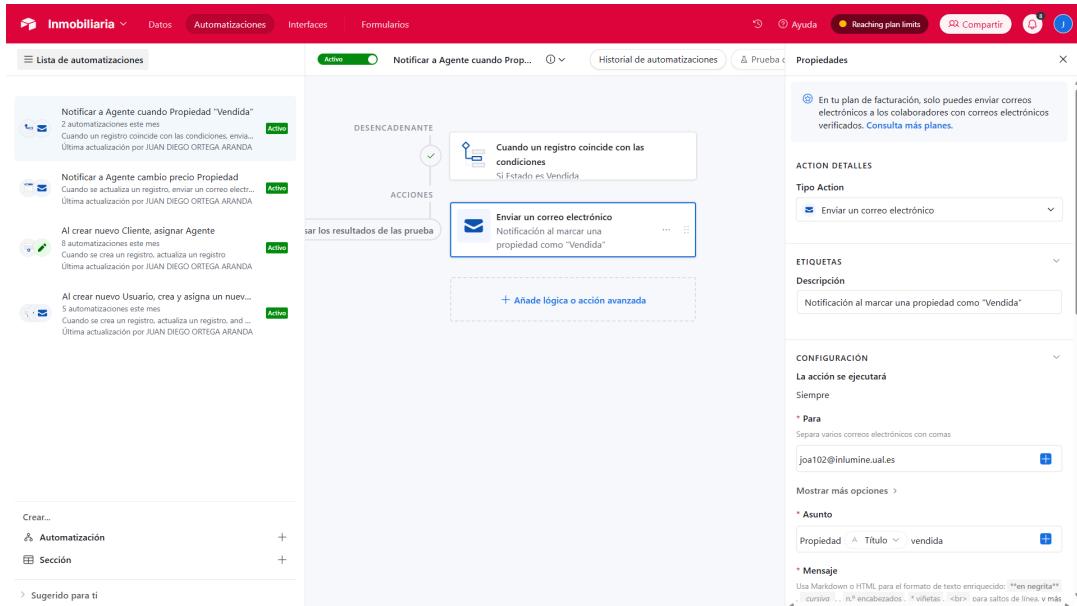
Este módulo describe la capa de automatización de flujos de trabajo que complementa las interfaces, operando en segundo plano para aumentar la eficiencia del sistema y mejorar la comunicación. Corresponde a los requisitos funcionales **RF 3.3, RF 6.2**.

Las automatizaciones configuradas directamente en Airtable son fundamentales para optimizar las operaciones diarias de los agentes inmobiliarios, reducir tareas repetitivas y garantizar una comunicación y un seguimiento eficientes. Aprovechan las capacidades low-code de Airtable para delegar lógica compleja, reduciendo la necesidad de desarrollo de código personalizado en el backend y frontend.

La interfaz de automatizaciones de Airtable permite definir flujos de trabajo mediante disparadores (triggers), condiciones y acciones, de forma visual. Se han implementado varias automatizaciones clave:

- Notificar a Agente cuando Propiedad "Vendida":** El propósito es asegurar que el agente responsable de una propiedad reciba una notificación inmediata cuando el estado de la misma cambia a "Vendida", permitiendo una respuesta y un cierre de gestión ágiles.

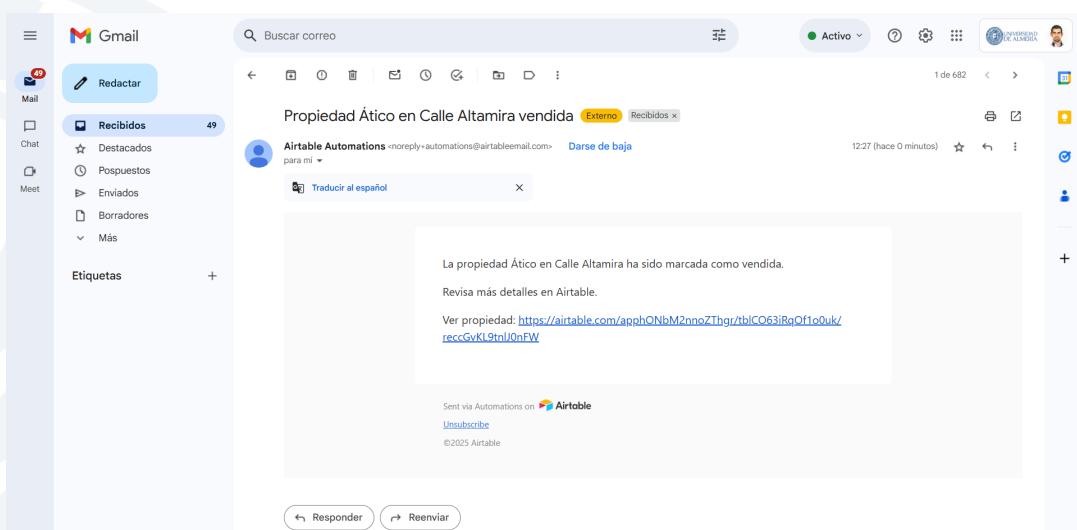
#### 4.4. DISEÑO E IMPLEMENTACIÓN



The screenshot shows the AirTable Automations interface. On the left, there's a list of existing automations, including "Notificar a Agente cuando Propiedad Vendida" which is active. The main panel displays the configuration for this specific automation. It consists of a trigger section ("DESENCADENANTE") and an action section ("ACCIONES"). The trigger is "Cuando un registro coincide con las condiciones" (If State is 'Vendida'). The action is "Enviar un correo electrónico" (Send an email) to the assigned agent. The configuration details include the recipient's email (joa102@inlumine.ua.es), subject ("Propiedad vendida"), and message content. A note on the right states: "En tu plan de facturación, solo puedes enviar correos electrónicos a los colaboradores con correos electrónicos verificados. Consulta más planes."

**Figura 4.68:** Automatización Notificar Agente Propiedad Vendida en AirTable

La automatización se activa "Cuando un registro en la tabla Propiedades coincide con ciertas condiciones" (disparador). La condición específica es que el campo Estado de una propiedad sea establecido a "Vendida". La acción es "Enviar un correo electrónico" al email del Agente Asignado de la propiedad. La Figura 4.68 muestra la configuración visual de esta automatización en Airtable.



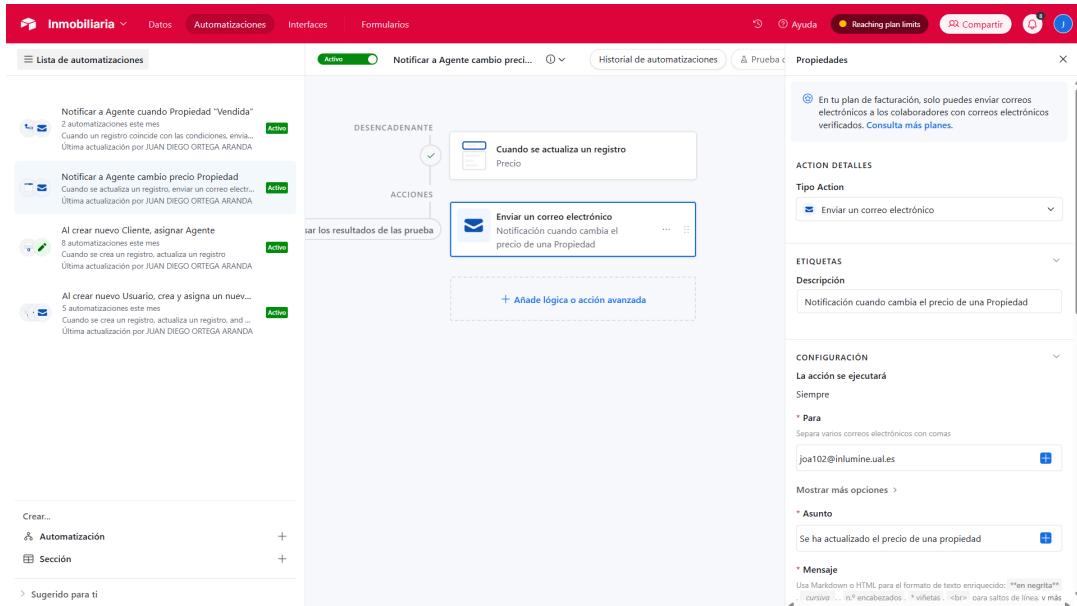
The screenshot shows a Gmail inbox with an email from "Airtable Automations <no-reply>automations@airtablemail.com>" with the subject "Propiedad Ático en Calle Altamira vendida". The email body contains a message: "La propiedad Ático en Calle Altamira ha sido marcada como vendida. Revisa más detalles en Airtable. Ver propiedad: <https://airtable.com/appbONbM2nnoZThgr/tb1C063iRqOf1o0uk/reccGvKL9tnlJ0nFW>". At the bottom, it says "Sent via Automations on Airtable" and "Unsubscribe".

**Figura 4.69:** Email de Automatización Notificar Agente Propiedad Vendida en AirTable

Y la Figura 4.69 muestra el correo electrónico enviado al Agente notificando que la Propiedad ha sido vendida.

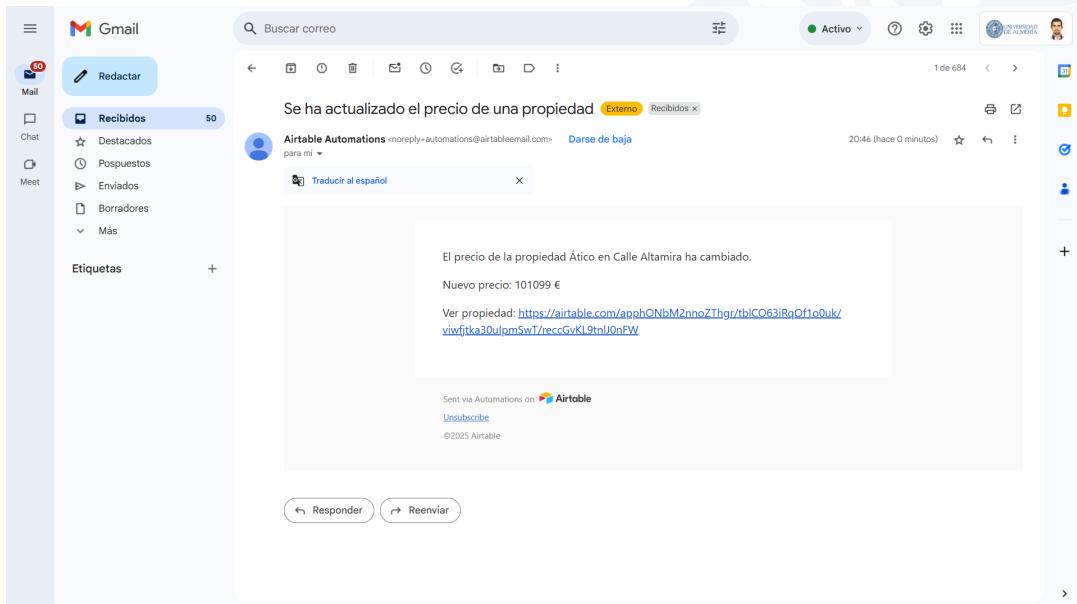
#### 4.4. DISEÑO E IMPLEMENTACIÓN

**2. Notificar a Agente cambio precio Propiedad:** El propósito es informar proactivamente al agente asignado sobre cualquier modificación en el precio de una propiedad bajo su gestión, garantizando que siempre manejen la información más actualizada.



**Figura 4.70:** Automatización Notificar Agente Cambio Precio Propiedad en AirTable

El disparador es "Cuando un registro en la tabla Propiedades es actualizado", con la condición de que el campo Precio "ha cambiado". La acción subsiguiente es "Enviar un correo electrónico" al Agente Asignado. La Figura 4.70 ilustra esta configuración.

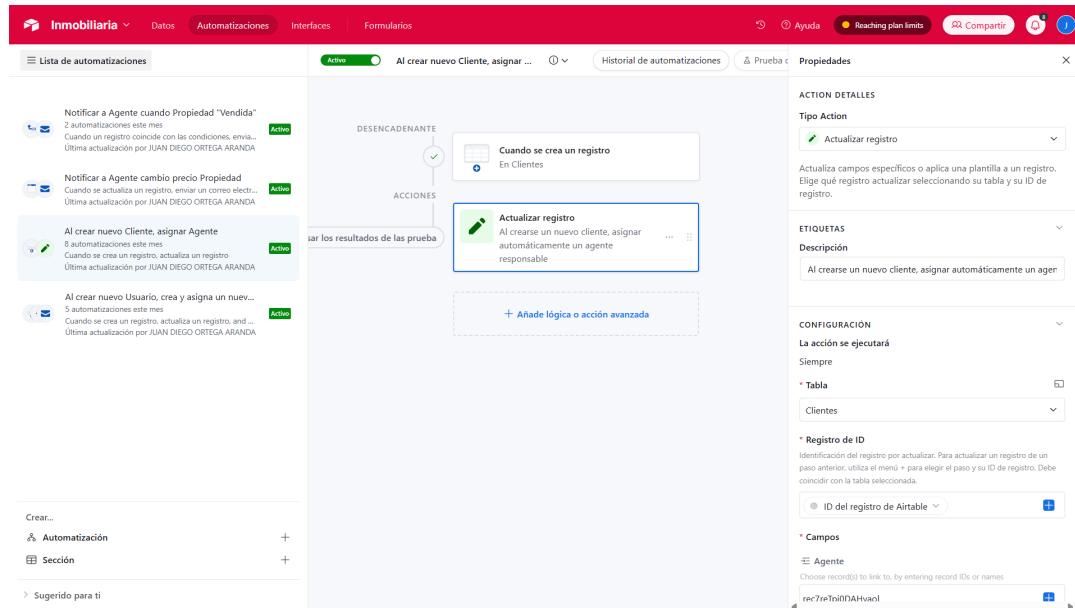


**Figura 4.71:** Email de Automatización Notificar Agente Cambio Precio Propiedad en AirTable

#### 4.4. DISEÑO E IMPLEMENTACIÓN

Y la Figura 4.71 muestra el correo electrónico enviado al Agente notificando el cambio de precio de una Propiedad.

3. **Al crear nuevo Cliente, asignar Agente:** El propósito es simplificar la asignación inicial de responsabilidades a los nuevos clientes captados, garantizando que cada nuevo lead tenga un agente de referencia.

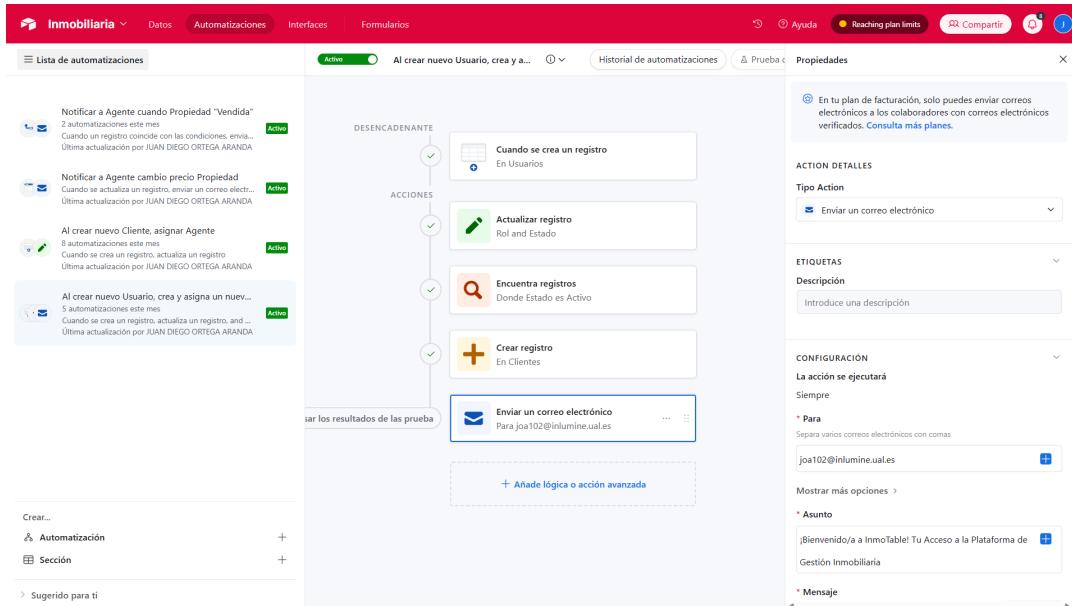


**Figura 4.72:** Automatización Asignar Agente al Crear Nuevo Cliente en AirTable

La automatización se dispara "Cuando se crea un registro en la tabla Clientes. La acción es "Actualizar registro" en la misma tabla Clientes, asignando automáticamente un Agente predefinido o siguiendo una lógica de asignación (ej., ronda, menos asignaciones). La Figura 4.72 muestra la configuración.

4. **Al crear nuevo Usuario, crea y asigna un nuevo Cliente:** El propósito es sincronizar automáticamente el registro de usuarios de la aplicación web con la base de datos de clientes de la inmobiliaria, facilitando la gestión integral del perfil del cliente.

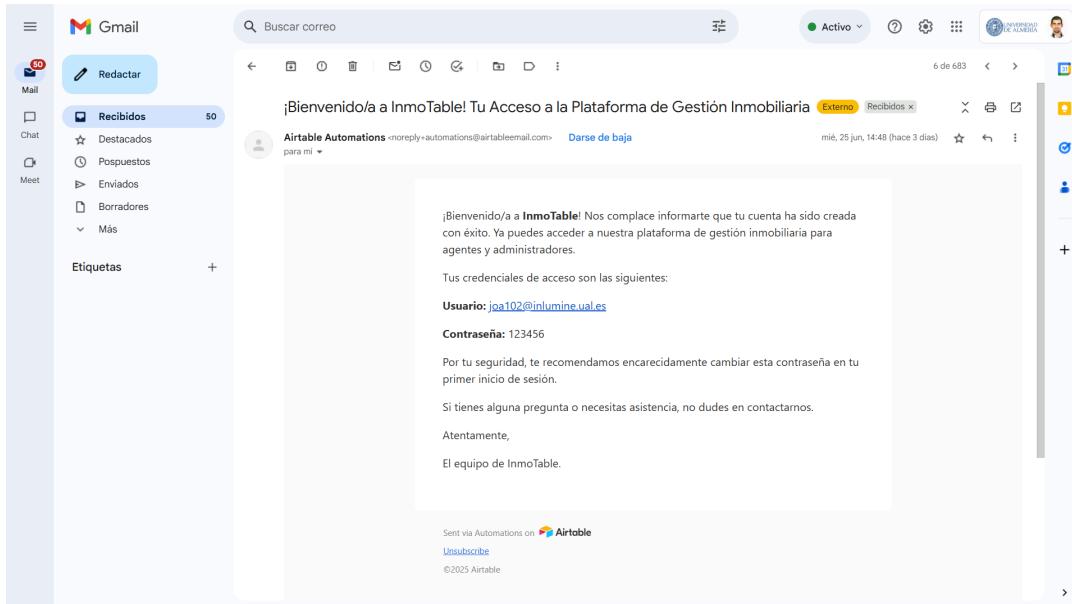
#### 4.4. DISEÑO E IMPLEMENTACIÓN



**Figura 4.73:** Automatización Asigna nuevo Cliente al Crear nuevo Usuario en AirTable

Esta automatización se activa "Cuando se crea un registro en la tabla Usuarios (el registro proviene del formulario de registro incrustado en la web, Figura 4.28). Las acciones subsecuentes incluyen "Actualizar registro" (posiblemente para establecer un Rol o Estado por defecto en la tabla Usuarios), "Crear registro" en la tabla Clientes con los datos del nuevo usuario, y "Enviar un correo electrónico" al usuario registrado (para la bienvenida o activación de cuenta, tal como se discutió previamente, preferiblemente con un enlace seguro para establecer contraseña). La Figura 4.73 ilustra este flujo.

#### 4.4. DISEÑO E IMPLEMENTACIÓN



**Figura 4.74:** Email de Automatización Asigna nuevo Cliente al Crear nuevo Usuario en AirTable

Y la Figura 4.74 muestra el correo electrónico enviado al usuario que se ha registrado.

##### 4.4.2.7 Módulo: Gestión de Cuentas de Usuario y Roles de Acceso (Airtable)

Este módulo es administrado principalmente por el Administrador a través de Airtable, controlando los accesos del personal. Se centra específicamente en la gestión de las credenciales de acceso y los niveles de permiso de los colaboradores que utilizan la base de datos de Airtable para la gestión interna de la inmobiliaria.

Colaborador/grupo (checkbox)	Permisos	Invitado por	Invitado ↑
<input type="checkbox"/> Paco García frangarc@gmail.com	ESPACIO DE TRABAJO Creador	J	Hace 7 días
<input type="checkbox"/> Juandi juandioa90@gmail.com	ESPACIO DE TRABAJO Solo lectura	J	12 de mayo
<input type="checkbox"/> Juandi juandi_1990@hotmail.com	Editor	J	12 de mayo
<input type="checkbox"/> JUAN DIEGO ORTEGA ARANDA joa102@inlumine.ual.es	ESPACIO DE TRABAJO Propietario	J	28 jun 2024

**Figura 4.75:** Gestión de Cuentas de Usuario y Roles de Acceso en Airtable

Los administradores utilizan la interfaz de gestión de colaboradores de Airtable (visible en la Figura 4.75) como su herramienta principal para controlar estos registros. Esta interfaz muestra a los usuarios internos por su email, nombre, última actividad y, crucialmente, su Rol asignado dentro de Airtable.

**Roles en Airtable:** La gestión de usuarios se basa en los roles nativos de Airtable, que definen sus capacidades:

- **Administrador (ACT-04):** En Airtable, estos usuarios se corresponden con roles de alto nivel como **Propietario (Owner)** o **Creador (Creator)**. Tienen control total sobre la base de datos, incluyendo la capacidad de añadir/eliminar colaboradores, modificar la estructura de las tablas, gestionar interfaces y automatizaciones, y acceder a todos los datos.
- **Agente (ACT-03):** Se corresponden con el rol de **Editor** en Airtable. Pueden ver y modificar datos en las tablas a las que se les ha dado acceso, añadir registros, usar formularios e interfaces, pero no tienen control sobre la estructura de la base o la administración de usuarios y configuración avanzada.

#### 4.4.3 Flujo de la aplicación

En este apartado se va a analizar el funcionamiento interno de algunas de las funcionalidades de nuestro sistema, centrándose en el flujo de cargar el formulario de cita, ya que este es fundamental para mostrar la comunicación de la interfaz de la aplicación web con la API de Airtable, el flujo de actualizar usuario para comprender cómo se relaciona el frontend con el backend. Y, además, se analizará la funcionalidad de añadir una propiedad a favoritos.

##### 4.4.3.1 Flujo de cargar el Formulario de Cita

Este flujo es un ejemplo destacado de cómo la aplicación InmoTable aprovecha la flexibilidad de Airtable para funcionalidades orientadas al cliente, minimizando el desarrollo de código personalizado. Se centra en el proceso que un cliente sigue para solicitar una cita para una propiedad.

- **Carga del Formulario:** Cuando un **Cliente Registrado** (ACT-02) navega a la página de detalle de una propiedad y decide solicitar una cita, la aplicación web (Frontend Angular) no renderiza un formulario Angular personalizado. En su lugar, construye dinámicamente una URL para un formulario incrustado de Airtable. Esta URL se configura para precargar automáticamente los campos Propiedad (utilizando el Record \_ ID de la propiedad actual) y Cliente (utilizando el Record \_ ID del cliente logueado) mediante parámetros `prefill_` en la URL (ej. [https://airtable.com/.../form?prefill\\_Propiedad=reciFQKpKH26ZbDd5&prefill\\_Cliente=recrq9z2GFijDLWIe](https://airtable.com/.../form?prefill_Propiedad=reciFQKpKH26ZbDd5&prefill_Cliente=recrq9z2GFijDLWIe)). Estos campos se ocultan en el formulario para el usuario final, garantizando que los datos críticos se asocien correctamente sin intervención manual.
- **Visualización e Interacción:** La aplicación web incrusta este formulario de Airtable (visible en la Figura 4.76). El cliente interactúa con los campos visibles del formulario, como la fecha, hora y el tipo de cita, así como cualquier comentario adicional.
- **Envío del Formulario:** Al hacer clic en el botón 'Guardar' (o 'Enviar'), el formulario de Airtable envía los datos directamente a la base de datos de Airtable. La interfaz de Airtable gestiona la persistencia de los datos, registrando la nueva cita en la tabla Citas.

- **Confirmación:** Airtable devuelve una respuesta de éxito al formulario incrustado, y el cliente recibe una confirmación visual en la interfaz web.

#### Justificación de la Implementación:

La decisión de utilizar un formulario incrustado de Airtable para la solicitud de citas por parte del cliente (RF 1.7) se basa en la rapidez y simplicidad de implementación. Permite una integración directa con la base de datos de Airtable para la captura de datos, evitando la necesidad de programar un formulario complejo en Angular, la lógica de validación y la comunicación con la API de Laravel y, a su vez, con la API de Airtable para la inserción del registro. Esto contrasta con el esfuerzo requerido para los formularios nativos de Angular como "Mi Perfil". Este enfoque demuestra la eficiencia en el desarrollo y el ahorro de tiempo al delegar funcionalidades de recolección de datos a una plataforma *low-code*.

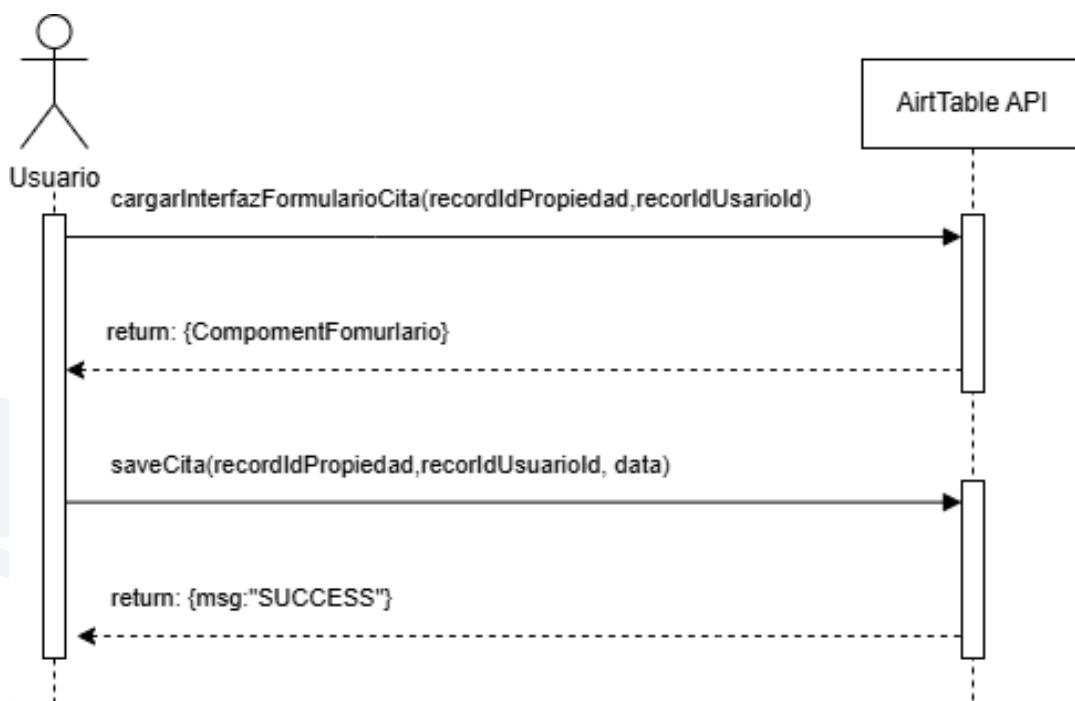


Figura 4.76: Diagrama Secuencias - Formulario Pedir Cita AirTable

#### 4.4.3.2 Flujo de Actualización de Datos de Usuario (Cliente)

Este flujo ilustra la comunicación bidireccional entre el frontend, el backend y la capa de datos de Airtable para la gestión de la información personal de un cliente autenticado.

- **Inicio de la Edición:** Un Cliente Registrado (ACT-02), después de iniciar sesión, accede a la sección "Mi Perfil" en la aplicación web. Esta interfaz (`user-form.component.html`) (visible en la Figura 4.77), construida con componentes Angular nativos, le permite visualizar y modificar sus datos personales (nombre, teléfono, email y contraseña).
- **Envío de Datos al Backend:** Al hacer clic en 'Guardar Cambios', la aplicación web (Frontend Angular) envía una petición HTTP (PATCH o PUT) al endpoint `/api/user/id` del Backend (Laravel API), incluyendo el ID del usuario y los datos modificados.

- **Validación y Lógica del Backend:** El Backend (Laravel API) recibe la petición. Primero, realiza una validación exhaustiva de los datos recibidos. Si la validación es exitosa, el backend consulta la tabla Usuarios de Airtable (a través del AirtableService.php) para verificar la existencia del usuario y obtener su Record ID en Airtable.
- **Actualización en Airtable:** Con el Record ID de Airtable, el Backend (Laravel API) procede a enviar una petición PATCH a la API de Airtable para actualizar la información del usuario en su registro correspondiente en la tabla Usuarios. Esto incluye la actualización del password \_ hash si la contraseña fue modificada.
- **Respuesta del Sistema:** Si la operación de actualización en Airtable se completa correctamente, Airtable responde con éxito al Backend (Laravel API). El backend, a su vez, devuelve una respuesta al Frontend Angular indicando success: true. En caso de que el usuario no exista o la validación de datos falle, el backend enviará una respuesta success: false o un mensaje de error detallado.

#### **Justificación de la Implementación:**

Este flujo demuestra el rol crítico del backend de Laravel en la seguridad (validación, autenticación con Passport) y la orquestación de las operaciones CRUD con Airtable para datos sensibles como los de usuario. A diferencia del formulario de registro incrustado, la gestión del perfil requiere un control más granular sobre la lógica de negocio y la validación en el servidor, lo que justifica la implementación de un formulario nativo en Angular y la lógica de backend en Laravel para esta funcionalidad.

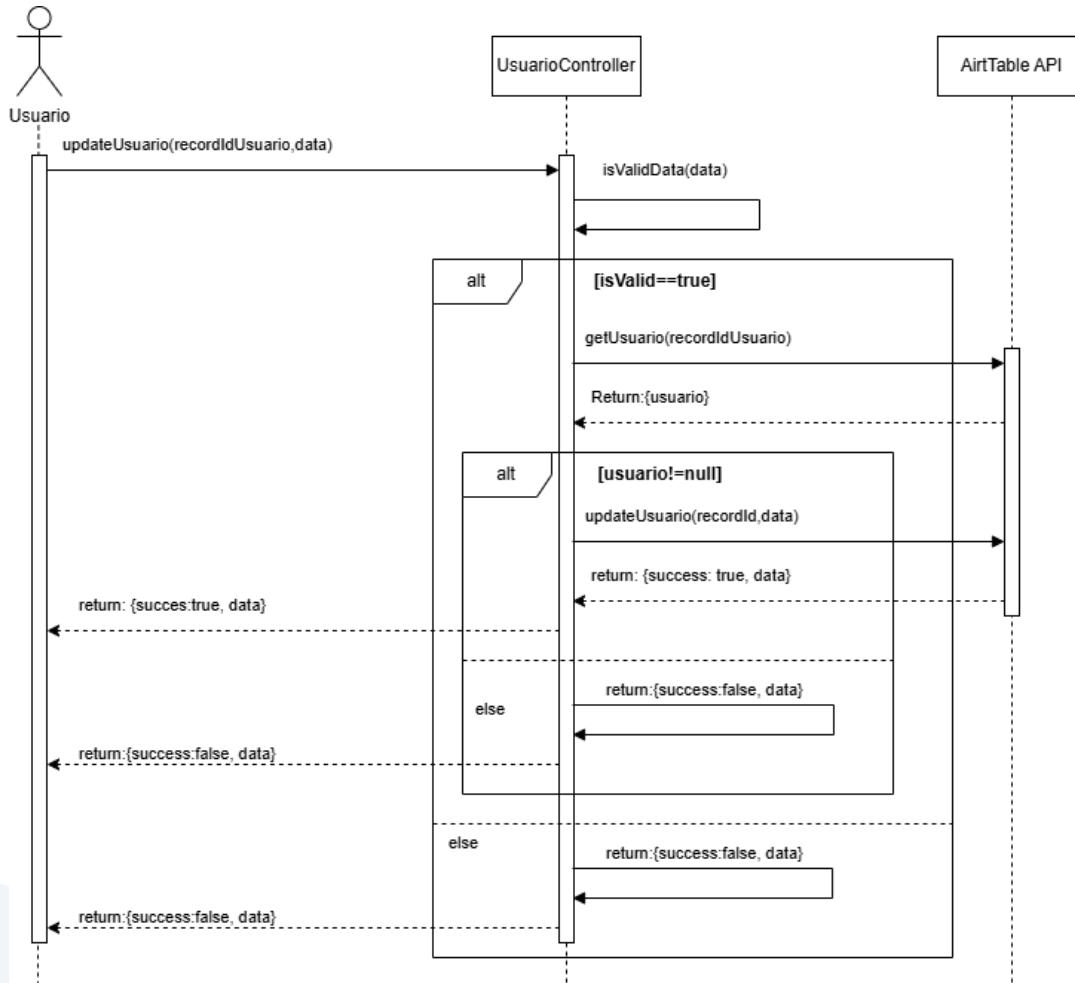


Figura 4.77: Diagrama Secuencias - Formulario Editar Perfil Usuario

#### 4.4.3.3 Flujo de Gestión de Propiedades Favoritas (Cliente)

Este flujo detalla cómo un **Cliente Registrado** (ACT-02) puede gestionar su lista de propiedades favoritas desde la aplicación web, una funcionalidad que se refleja directamente en la base de datos de Airtable para la visibilidad del agente.

- **Acción del Usuario:** Cuando un cliente autenticado selecciona la opción "Agregar a Favoritos" (o "Quitar de Favoritos") en el listado o detalle de una propiedad (visible en la Figura 4.78), la aplicación web (Frontend Angular) envía una petición HTTP (POST o PATCH) al endpoint `api/favorites` del Backend (Laravel API). Esta petición incluye el `clienteId` y el `propiedadId` relevantes.
- **Procesamiento del Backend:** El Backend (Laravel API) recibe la solicitud. Para gestionar el array de propiedades favoritas, realiza una petición GET a la API de Airtable para obtener el registro completo del cliente desde la tabla Clientes, incluyendo su lista actual de Propiedades \_ interesadas (que es un campo de "linked records" en Airtable).

- **Lógica de Actualización:** El backend verifica si la propiedadId ya se encuentra en el array de Propiedades \_ interesadas del cliente.
  - **Si la propiedad no está en la lista:** El backend añade el propiedadId al array.
  - **Si la propiedad ya está en la lista (para "Quitar de Favoritos"):** El backend elimina el propiedadId del array.
- **Actualización en Airtable:** Una vez modificado el array, el Backend (Laravel API) envía una petición PATCH a la API de Airtable para actualizar el campo Propiedades \_ interesadas en el registro del cliente.
- **Respuesta del Sistema:** Si la operación en Airtable se realiza correctamente, Airtable responde con éxito. El backend, a su vez, devuelve una respuesta al Frontend Angular (ej., success: true ), y la interfaz del cliente muestra una confirmación visual al usuario (como en Figuras 4.33 y 4.35). En caso de error (ej., propiedad ya agregada al intentar añadirla), se devuelve una respuesta adecuada (ej., success: false ).

#### Justificación de la Implementación:

Este flujo subraya la capacidad del backend de Laravel para manejar la lógica de negocio compleja y las interacciones avanzadas con la API de Airtable que van más allá de un simple CRUD directo. Airtable, con sus campos de "linked records", soporta directamente esta funcionalidad de "favoritos" en su capa de datos. El backend actúa como un orquestador, transformando las peticiones de los clientes en operaciones de datos coherentes y seguras en Airtable, y luego proporcionando una retroalimentación instantánea a la interfaz de usuario.

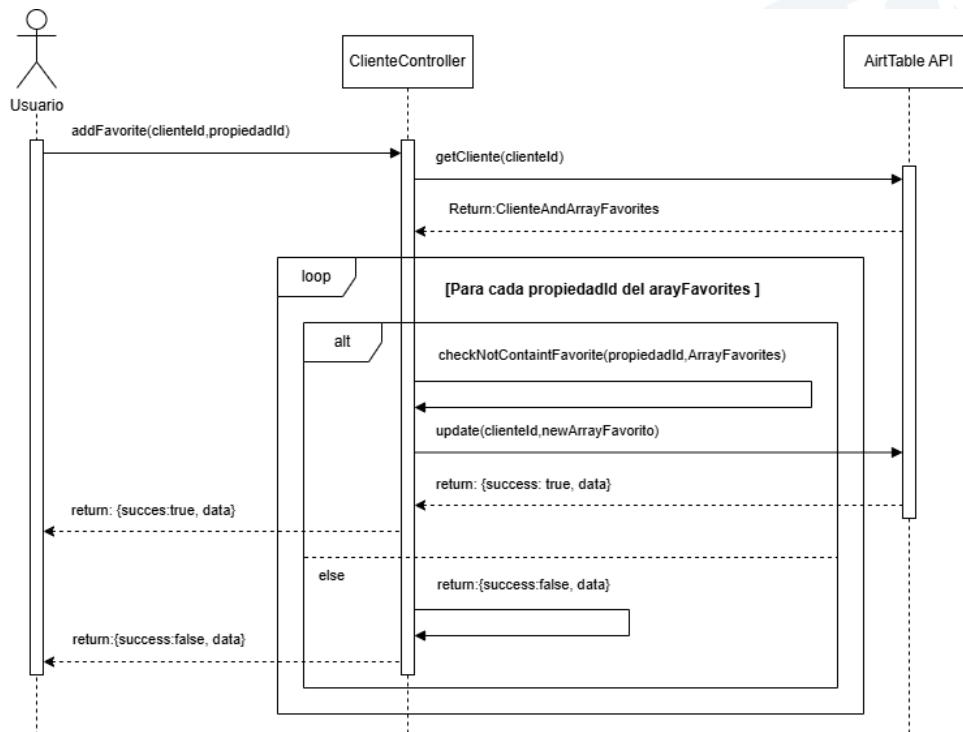


Figura 4.78: Diagrama Secuencias - Formulario Pedir Cita AirTable

#### 4.4.4 Implementación

La implementación de **InmoTable** se ha llevado a cabo siguiendo una arquitectura modular y basada en patrones de diseño probados, lo que garantiza la escalabilidad, mantenibilidad y eficiencia del sistema. Esta sección detalla la estructura interna de los componentes del frontend y backend, así como los patrones clave aplicados en su desarrollo.

##### 4.4.4.1 Estructura General del Proyecto

El proyecto **InmoTable** se organiza en un repositorio raíz que contiene las siguientes carpetas principales, reflejando la separación de las capas de la arquitectura:

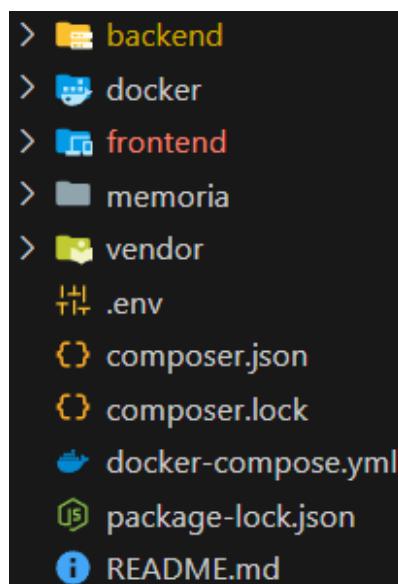


Figura 4.79: Estructura General del Proyecto

- **frontend/**: Contiene todo el código de la aplicación web del cliente, desarrollada en Angular.
- **backend/**: Aloja el código de la API RESTful, implementada con el framework Laravel.
- **memoria/**: Guarda los archivos relacionados con la documentación del Trabajo de Fin de Grado (ej., archivos LaTeX).
- **docker-compose.yml**: Define la configuración del entorno de desarrollo y despliegue de la aplicación, orquestando los servicios de frontend y backend en contenedores Docker.

#### 4.4.4.2 Implementación del Frontend (Aplicación Web Angular)

El frontend se ha construido como una Single Page Application (SPA) utilizando Angular, con un enfoque en la modularidad y la reutilización de componentes.

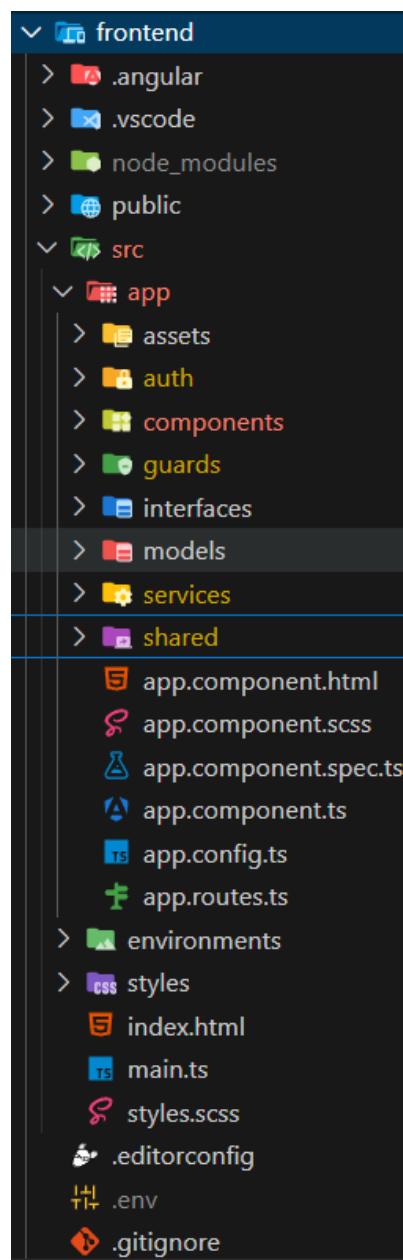
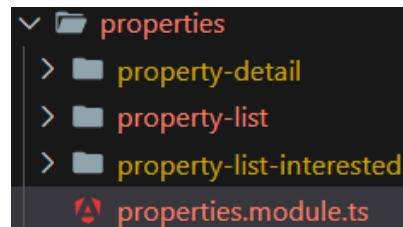


Figura 4.80: Estructura del Frontend

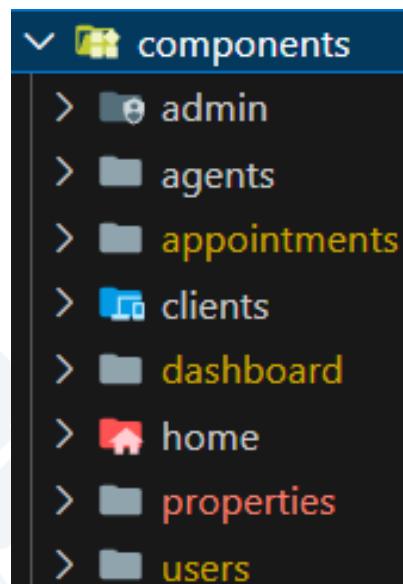
- **Arquitectura Modular y por Componentes:**

- La aplicación se divide en **módulos lógicos** (ej., properties.module.ts, auth.module.ts) que agrupan funcionalidades relacionadas, facilitando la organización y el lazy loading (carga perezosa) de módulos para optimizar el rendimiento.



**Figura 4.81:** Estructura de los módulos lógicos del Frontend

- Se sigue un patrón de **componentes** (.component.html, .component.ts, .component.scss) que encapsulan la lógica, la vista y los estilos de partes específicas de la interfaz de usuario. Esto promueve la reusabilidad (ej., un property-card.component utilizado en listados y favoritos) y la mantenibilidad.



**Figura 4.82:** Estructura de Componentes del Frontend

- **Capa de Servicios (Services Layer):** La lógica de negocio y la comunicación con el backend se centralizan en servicios inyectables (.service.ts). Ejemplos incluyen api.service.ts (para las peticiones HTTP generales), propiedades.service.ts, citas.service.ts, auth.service.ts (para la lógica de autenticación) y empresa.service.ts (para la configuración dinámica de la empresa). Esta separación de responsabilidades facilita las pruebas y la reutilización.

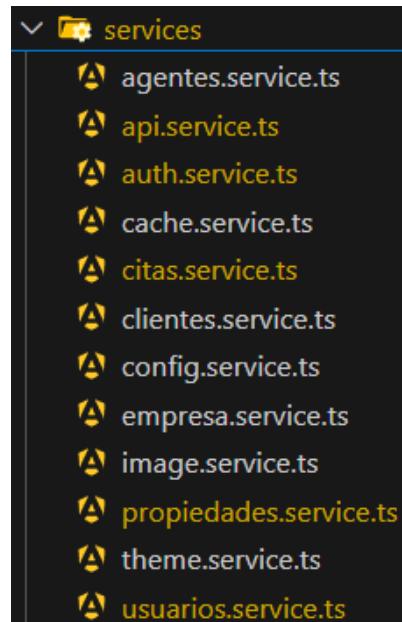


Figura 4.83: Estructura de Servicios del Frontend

- **Sistema de Rutas (Routing):** La navegación entre las diferentes vistas de la aplicación se gestiona mediante el módulo de routing de Angular (app.routes.ts, auth.routes.ts), permitiendo URLs limpias y una navegación sin recarga de página.

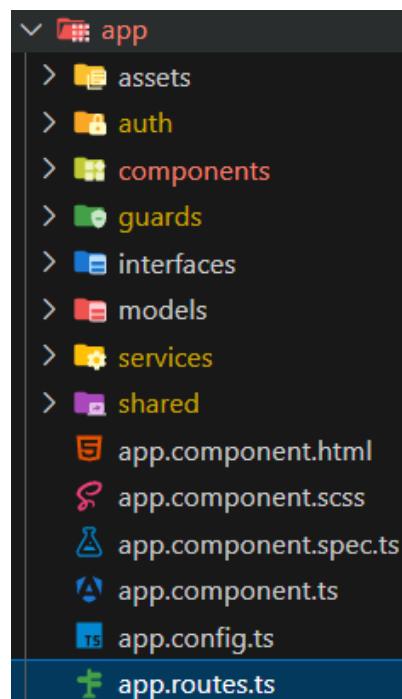


Figura 4.84: Estructura de Rutas del Frontend

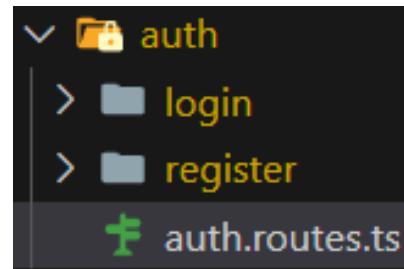


Figura 4.85: Estructura de Rutas Auth del Frontend

- **Guards de Autenticación y Autorización:** Se implementan AuthGuards (auth.guard.ts) y RoleGuards (role.guard.ts) para proteger rutas, asegurando que solo los usuarios autenticados o con roles específicos puedan acceder a ciertas secciones de la aplicación, reforzando la seguridad.

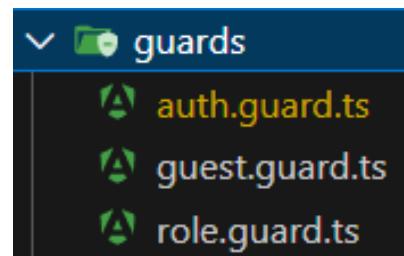


Figura 4.86: Estructura de Guards del Frontend

- **Estilos y Tematización Dinámica:** El styling se gestiona con SCSS (styles.scss) y variables CSS personalizadas (\_variables.scss). Un aspecto clave es la capacidad de la aplicación para configurar dinámicamente los colores globales y el logo/nombre de la empresa directamente desde los datos almacenados en la tabla Empresa de Airtable. Esto permite una personalización rápida y sin código de la marca para cada instancia de inmobiliaria, apoyando la portabilidad.

#### 4.4.4.3 Implementación del Backend (Laravel API)

El backend se ha construido sobre el framework Laravel para ofrecer una API RESTful robusta y segura, actuando como el principal punto de comunicación y orquestación.

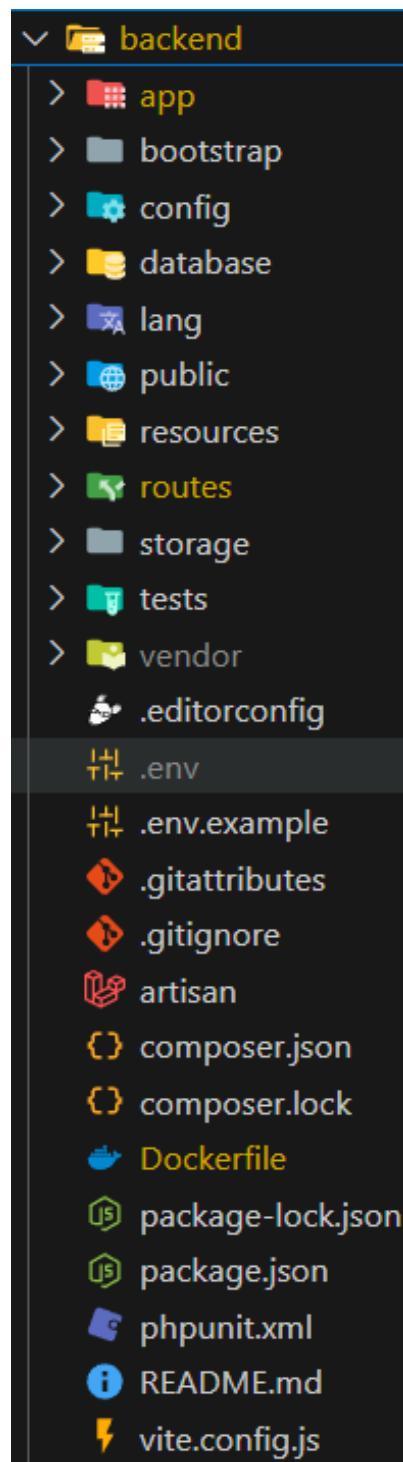
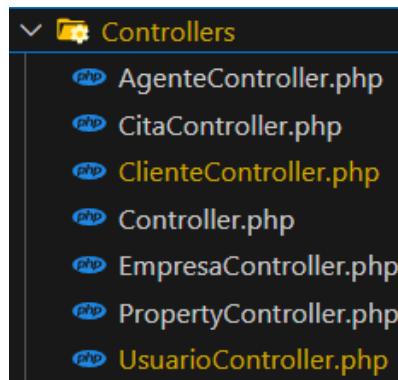


Figura 4.87: Estructura del Backend

■ **Patrón MVC (Modelo-Vista-Controlador) Adaptado a API:**

- Laravel sigue el patrón MVC. En este contexto API-céntrico, los Controladores (ej., `PropertyController.php`, `CitaController.php`, `UsuarioController.php`) actúan como los endpoints que reciben las peticiones HTTP del frontend.



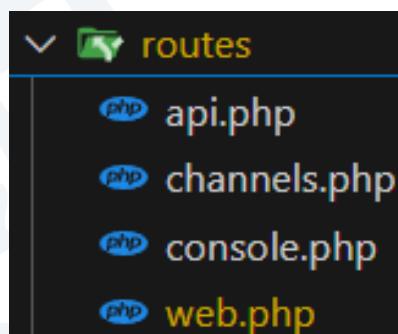
**Figura 4.88:** Estructura de los Controladores del Backend

- La **lógica de negocio** principal se encapsula en Servicios (ej., `AirtableService.php`), que son los responsables de interactuar con la API externa de Airtable y de aplicar las reglas de negocio. Esta separación (`Controller -> Service -> External API`) promueve un código limpio y fácil de mantener.



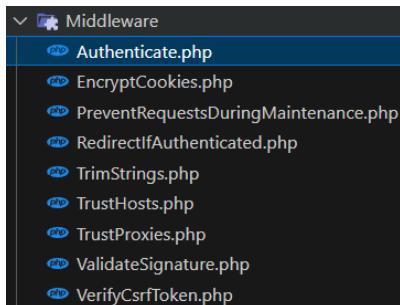
**Figura 4.89:** Estructura de los Servicios del Backend

- **Gestión de Rutas API:** Las rutas de la API se definen en `routes/api.php`, mapeando las URLs a los métodos de los controladores.



**Figura 4.90:** Estructura de las Rutas del Backend

- **Sistema de Autenticación y Autorización:** Se utiliza **Laravel Passport** para gestionar la autenticación de usuarios mediante tokens OAuth2. Esto permite un acceso seguro y stateless a la API, crucial para la comunicación con el frontend. Los middlewares de autenticación y roles (`Authenticate.php`, `RoleGuard.php`) en el backend y frontend garantizan el control de acceso a los recursos.



**Figura 4.91:** Estructura del Middleware del Backend

- **Validación de Datos:** Laravel proporciona un potente sistema de validación que asegura la integridad y consistencia de los datos recibidos de las peticiones HTTP antes de ser procesados o enviados a Airtable.

#### 4.4.4.4 Patrones de Integración con Airtable

La integración con Airtable se realiza siguiendo patrones específicos para optimizar el rendimiento, la facilidad de desarrollo y la funcionalidad:

- **API-First Approach:** El backend de Laravel interactúa con Airtable exclusivamente a través de su API RESTful. Esto permite que Airtable actúe como una base de datos centralizada y un "microservicio" de gestión de contenidos y lógica de negocio interna.
- **Orquestación Centralizada por Laravel:** El AirtableService en Laravel orquesta todas las operaciones de datos complejas. Esto incluye:
  - **Lectura/Escritura de Datos:** Recuperar propiedades, clientes, citas para el frontend, y enviar nuevas solicitudes o actualizaciones (ej., propiedades favoritas, actualización de perfil de usuario).
  - **Manejo de Campos Vinculados (Linked Records):** El backend gestiona las relaciones entre tablas de Airtable (ej., Propiedades vinculadas a Agentes, Clientes vinculados a Propiedades \_ interesadas) de forma programática.
  - **Procesamiento de Datos Complejos:** Realiza lógica adicional (ej., verificar si una propiedad ya está en favoritos antes de añadirla, como se ve en el Flujo de Gestión de Propiedades Favoritas).

- Delegación de UI/UX de Gestión a Airtable (Embedded Forms & Views):
  - Para formularios de entrada de datos simples y flujos de usuario directos a la base de datos (ej., **Registro de nuevos clientes**, **Formulario de Contacto**, **Solicitud de Cita**), la aplicación web **incrusta directamente formularios de Airtable**. Esto reduce drásticamente el tiempo de desarrollo del frontend, ya que se aprovecha la capacidad low-code de Airtable para generar y gestionar estos formularios.
  - Para la **visualización de datos personalizados para el cliente** (ej., el calendario de citas del cliente), la aplicación web también **incrusta vistas de Airtable** filtradas dinámicamente mediante parámetros en la URL (ej., `filter _ Cliente = ID _ Cliente`), demostrando la flexibilidad de Airtable.
- **Automatizaciones en Airtable:** Las automatizaciones (disparadores, condiciones y acciones) se configuran directamente en la interfaz de Airtable (ej., para la creación automática de clientes tras un registro web, o notificaciones). Esto externaliza la lógica de negocio secundaria de la programación y la gestión por código.

#### 4.4.4.5 Entorno de Desarrollo y Despliegue (Docker)

La implementación se ha apoyado en **Docker** para la contenerización del frontend (servicio por Nginx) y el backend (Laravel y PHP-FPM). Este patrón asegura la consistencia del entorno de desarrollo entre los colaboradores y facilita un despliegue homogéneo en producción, aislando las dependencias y garantizando la portabilidad del proyecto.

# 5 CONCLUSIONES Y TRABAJOS FUTUROS

## 5.1 CONCLUSIONES

El proyecto **InmoTable** ha culminado exitosamente con el desarrollo de una aplicación web integral para la gestión inmobiliaria, validando su propuesta de valor. La solución implementa una **arquitectura híbrida** que combina un frontend en Angular (para una interfaz intuitiva al cliente) y un backend en Laravel (para lógica de negocio y seguridad) con **Airtable** como base de datos central y potente plataforma *low-code* para la gestión interna de agentes y administradores. Este enfoque ha permitido alcanzar los objetivos de eficiencia, un bajo coste operativo y una notable capacidad de replicabilidad. Se ha demostrado que **InmoTable** facilita una gestión inmobiliaria digital, ágil y rentable, transformando la interacción con el cliente y optimizando las operaciones internas mediante formularios visuales, vistas especializadas, dashboards analíticos y automatizaciones personalizadas.

## 5.2 CONOCIMIENTOS ACADÉMICOS ADQUIRIDOS

La realización de este Trabajo de Fin de Grado ha consolidado y ampliado significativamente mis conocimientos académicos y habilidades en diversas áreas clave de la ingeniería informática:

- **Ingeniería de Software:** Aplicación práctica de metodologías ágiles (Scrum), gestión de requisitos, y diseño de arquitecturas (N-tier, híbrida).
- **Desarrollo Web Full-Stack:** Experiencia avanzada en frontend (Angular, UI/UX) y backend (Laravel API, autenticación OAuth2, validación de datos).
- **Modelado y Gestión de Datos:** Comprensión de modelos de datos y uso estratégico de plataformas low-code (Airtable) para gestión interna, formularios y analítica.
- **DevOps y Despliegue:** Contenerización con Docker y familiarización con plataformas de despliegue continuo (Vercel, Render).
- **Integración de Sistemas:** Habilidad para interconectar tecnologías dispares (Angular-Laravel-Airtable) y orquestar flujos de datos complejos.
- **Resolución de Problemas y Comunicación Técnica:** Desarrollo de pensamiento crítico y capacidad de documentar proyectos de software de manera clara.

### 5.3. TRABAJOS FUTUROS

#### 5.3 TRABAJOS FUTUROS

El potencial de InmoTable ofrece diversas líneas para futuras mejoras y expansiones:

- **Optimización de Airtable:** Migrar a una versión de pago de Airtable para eliminar limitaciones actuales y acceder a mayor capacidad, más automatizaciones y funcionalidades avanzadas (Extensions), lo que incrementaría la robustez del sistema.
- **Mejoras en la Experiencia del Cliente:** Incorporar un buscador de propiedades avanzado con mapa interactivo y notificaciones en tiempo real.
- **Ampliación de Gestión Interna:** Desarrollar módulos de gestión de tareas y seguimiento de leads más complejos, así como la gestión documental.
- **Integraciones Externas:** Sincronización automática con portales inmobiliarios (MLS) y la integración de pasarelas de pago.
- **Escalabilidad y Despliegue:** Implementar pipelines CI/CD avanzados y estrategias de caching más robustas para el entorno de producción.
- **Expansión de Plataforma:** Considerar el desarrollo de aplicaciones móviles nativas y un panel de administración central para múltiples inmobiliarias.

---

## BIBLIOGRAFÍA

---

- [1] IONOS, “Low code vs no code,” <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/low-code-vs-no-code/>, 2024.
- [2] AppMaster, “Las 10 principales ideas sobre aplicaciones inmobiliarias para crear tu propia aplicación,” <https://appmaster.io/es/blog/las-10-principales-ideas-sobre-aplicaciones-inmobiliarias-para-crear-tu-propia-aplicacion>, 2024.
- [3] S. 1, “Cuadro comparativo de metodologías tradicionales,” <https://www.studocu.com/es-mx/document/universidad-del-valle-de-mexico/modelos-y-metodos-de-desarrollo-de-software/actividad-1-cuadro-comparativo/111121564>, 2024.
- [4] S. 2, “Cuadro comparativo metodologías Ágiles,” <https://www.studocu.com/es-mx/document/instituto-tecnologico-de-pachuca/ingenieria-de-software/cuadro-comparativo-metodologias-agiles/23713846>, 2024.
- [5] C. Lasa Gómez, A. Álvarez García, and R. de las Heras del Dedo, *Métodos Ágiles. Scrum, Kanban, Lean.* RA-MA Editorial, 2021.
- [6] Atlassian, “Agile vs devops,” <https://www.atlassian.com/es/devops/what-is-devops/agile-vs-devops>, 2024.
- [7] Scribd, “Cuadro comparativo de arquitectura de software,” <https://es.scribd.com/document/587262812/Cuadro-Comparativo-de-Arquitectura-de-Software-1>, 2024.
- [8] Hiberus, “Angular vs react vs vue,” <https://www.hiberus.com/crecemos-contigo/angular-vs-react-vs-vue/>, 2024.
- [9] Back4App, “Los mejores marcos de backend en 2025,” <https://blog.back4app.com/es/los-mejores Marcos-de-backend-en-2025/>, 2024.
- [10] S. Lodos, “Notion vs airtable: ¿cuál es la mejor para ti?” <https://saralodos.com/notion-vs-airtable-cual-es-la-mejor-para-ti/>, 2024.
- [11] J. Secret, “Airtable vs smartsheet,” <https://www.joinsecret.com/compare/airtable-vs-smartsheet>, 2024.
- [12] Back4App, “Firebase vs airtable,” <https://blog.back4app.com/firebase-vs-airtable/>, 2024.
- [13] E. McMillan and T. Rowell, *Learning Airtable: Building Database-Driven Applications with No-Code.* O'Reilly Media, 2023.

- [14] I. 1, “Postgresql vs mysql,” <https://www.ionos.es/digitalguide/servidores/know-how/postgresql-vs-mysql/>, 2024.
- [15] I. 2, “Google sheets vs excel,” <https://www.ionos.es/digitalguide/online-marketing/vender-en-internet/google-sheets-vs-excel/>, 2024.
- [16] Figma, “Help center,” <https://help.figma.com/hc/es-es>, 2024.
- [17] GitHub, “Documentación oficial,” <https://docs.github.com/es>, 2024.
- [18] Composer, “Documentación oficial,” <https://getcomposer.org/doc/>, 2024.
- [19] npm, “Documentación oficial,” <https://docs.npmjs.com/>, 2024.
- [20] B. Cascales Salinas, *El libro de LaTeX*. Editorial Club Universitario, 2017.
- [21] Overleaf, “Documentación oficial,” <https://es.overleaf.com/learn>, 2024.
- [22] Vercel, “Documentación oficial,” <https://vercel.com/docs>, 2024.
- [23] Render, “Documentación oficial,” <https://render.com/docs>, 2024.
- [24] Angular, “Overview,” <https://angular.dev/overview>, 2024.
- [25] A. Material, “Angular material documentation,” <https://material.angular.dev>, 2024.
- [26] Bootstrap, “Bootstrap documentation,” <https://getbootstrap.com/docs>, 2024.
- [27] Laravel, “Documentación oficial,” <https://laravel.com/docs>, 2024.
- [28] A. 1, “Introduction to airtable basics,” <https://support.airtable.com/docs/introduction-to-airtable-basics>, 2024.
- [29] A. 2, “Airtable developers,” <https://airtable.com/developers>, 2024.
- [30] A. 3, “Support airtable,” <https://support.airtable.com/>, 2024.
- [31] Docker, “Documentación oficial,” <https://docs.docker.com/>, 2024.



El presente Trabajo de Fin de Grado describe el desarrollo de **InmoTable**, una innovadora aplicación web para la gestión inmobiliaria. La solución implementa una arquitectura híbrida que combina un **frontend en Angular** para una óptima experiencia del cliente y un **backend en Laravel** para la lógica de negocio y la integración API. Su principal diferenciador es el uso de **Airtable como base de datos central y potente plataforma low-code** para la gestión interna de agentes y administradores, incluyendo propiedades, clientes, citas y análisis de negocio. Este enfoque permite una alta eficiencia, un bajo coste operativo y una gran portabilidad, haciendo de InmoTable una solución replicable y ágil para diversas inmobiliarias.

This Final Degree Project describes the development of **InmoTable**, an innovative web application for real estate management. The solution implements a hybrid architecture combining an **Angular frontend** for an optimal client experience and a **Laravel backend** for business logic and API integration. Its main differentiator is the use of **Airtable as a central database and powerful low-code platform** for internal management by agents and administrators, including properties, clients, appointments, and business analytics. This approach enables high efficiency, low operational cost, and great portability, making InmoTable a replicable and agile solution for various real estate agencies.