



# MANUAL TÉCNICO

DOCUMENTACIÓN TÉCNICA DE AGROMANAGER  
ANDRÉS MANUEL EGIDO FERNÁNDEZ

## Contenido

Objetivos del Proyecto.....	2
Requisitos del Sistema .....	2
Descripción .....	3
Stack tecnológico .....	4
Parcelas.....	5
Atributos comunes a todas las parcelas (agrícolas y ganaderas): .....	5
Atributos Específicos por Tipo de Explotación .....	6
Administradores .....	7
Guía de estilos .....	8
Prototipo.....	9
Flujo de navegación .....	11
Listado de los ficheros de la aplicación .....	12
Despliegue .....	14
Prerrequisitos .....	14
Despliegue Web.....	14
Despliegue Android .....	14
Incidencias encontradas, pruebas realizadas y resolución.....	15
Posibles ampliaciones y mejoras .....	15

## Objetivos del Proyecto

Optimizar la gestión agropecuaria, permitiendo a los productores administrar cultivos, ganado e insumos desde una plataforma digital.

- Facilitar la toma de decisiones basada en datos en tiempo real.
- Reducir costos y mejorar la productividad mediante el control eficiente de recursos.
- Brindar acceso a la tecnología a pequeños y medianos productores con una interfaz intuitiva.

## Requisitos del Sistema

### 1. Requisitos Funcionales

- Registro y gestión de usuarios.
- Gestión de parcelas.
- Administración de cultivos y ganado.
- Registro de costos y reportes financieros.
- Accesibilidad web y móvil.

### 2. Requisitos No Funcionales

- Seguridad: Autenticación de usuarios y protección de datos.
- Rendimiento: Respuesta rápida y eficiente en la carga de datos.
- Compatibilidad: Soporte para dispositivos Android y navegadores web.

### 3. Requisitos de Interfaz

- Diseño intuitivo y minimalista con navegación sencilla.
- Interfaz responsive, adaptable a móviles y computadoras.
- Uso de colores asociados a la agroindustria.

## Descripción

La aplicación en la que voy a trabajar se denomina “AgroManager”. La intención de esta app es facilitar las tareas agropecuarias de una finca tradicional.

¿Qué tareas deseamos facilitar? Las tareas en las que nos centramos son las de inventariado de explotaciones ya sea de reses o cultivos, inventariado de tareas que se le deben realizar a dichas explotaciones, también facilitaremos la adquisición de los costes de que tiene dicha explotación, contabilizando la cantidad de productos necesarios para cada explotación y su coste ya sean estos fijos o variables.

¿Cómo facilitaremos estas tareas? Las facilitaremos de modo que toda la información necesaria a la hora de gestionar una finca la tengamos al alcance de la mano. Ya que queremos volcar toda esta información en una sola app. Con esta información nos referimos a extensiones de terreno que disponemos para su explotación. En las que podremos realizar diferentes labores ya sean de ganadería o agricultura.

¿De qué manera vamos a mostrar dichas extensiones de terreno? Dichas extensiones de terreno las cuales denominaremos parcelas, serán mostradas en un mapa de forma poligonal, previa configuración del usuario. En dicha configuración previa el usuario deberá indicar las coordenadas geográficas de cada uno de los vértices de dicha finca, un nombre para la parcela, tipo de explotación (agrícola o ganadera). Dependiendo del tipo que seleccione se le harán distintas preguntas.

En caso de que sea una parcela agrícola necesitaremos la siguiente información:

- Tipo de cultivo.
- Frecuencia de riego.
- Frecuencia de desparasitación.
- Frecuencia de fertilización.
- Tipo de explotación (perenne o temporal).
- Fecha de recolección (si es temporal).

En el caso de que sea una parcela para ganado necesitaríamos la siguiente información:

- Especie que habitará.
- Cantidad de cabezas de ganado.
- Frecuencia de alimentación.
- Frecuencia de hidratación.
- Frecuencia de desparasitación.
- Frecuencia de supervisión veterinaria.
- Tipo de explotación (sacrificio producción continua).

Esta información nos ayudará a la hora de calcular los costes de la explotación ya que si por ejemplo el usuario crea una parcela de cultivos de secano deberá poner que la

frecuencia de riego es cero y con ello en la app sabremos que no debemos contemplar el agua como un coste.

## Stack tecnológico

Las tecnologías elegidas para este proyecto son las siguientes:

- Frontend: Angular + Ionic + Capacitor
- UI: Componentes Ionic + Angular Material (para web)
- Mapas: Google Maps API + @capacitor/geolocation
- Gráficos: Chart.js
- Backend: Firebase (Firestore, Auth)

¿Por qué queremos implementar este stack?

- Angular + Ionic + Capacitor: esta combinación te permite desarrollar una sola vez y desplegar en Android y web con alta calidad.
- Angular (Framework Frontend): es un framework de JavaScript/TypeScript para construir aplicaciones web dinámicas y escalables.
- Ionic (Framework UI para Apps Móviles y Web): es un framework que permite desarrollar interfaces móviles usando tecnologías web (HTML, CSS, JavaScript) con componentes prediseñados.
- Capacitor: es una herramienta que convierte una aplicación web (Angular + Ionic) en una app nativa para Android, iOS y web.
- Firebase: elimina la necesidad de un backend complejo y nos aporta una base de datos no SQL.
- Google Maps: es la solución más factible para dibujar parcelas con polígonos.

# Modelo de datos

## Parcelas

Atributos comunes a todas las parcelas (agrícolas y ganaderas):

Campo	Tipo	Descripción	Ejemplo
id	string	ID único generado automáticamente.	"MISlzdVHXFnk7u4a3YlV"
tipoExplotacion	string	Tipo de explotación: "agricola" o "ganadera".	"ganadera"
name	string	Nombre descriptivo de la parcela.	"Andrés agrícola"
vertices	Array<{lat: number, lng: number}>	Coordenadas geográficas que delimitan la parcela (polígono).	[{lat: 37.9229, lng: -6.8280}, ...]
id Usuario	string	ID del usuario propietario.	"yb1kMVrcgpfv7wXRjXioT4XGhu63"
correo	string	Correo asociado al usuario.	"andres@gmail.com"
fechaCreacion	timestamp	Fecha de creación de la parcela.	"2024-04-01T00:00:00.000Z"
finalidad	string	Objetivo principal (ej: "sacrificio", "temporal").	"temporal"

## Atributos Específicos por Tipo de Explotación

### 1. Parcela Agrícola

Campo	Tipo	Descripción	Ejemplo
especie	string	Cultivo principal (ej: "Tomates").	"Tomates"
fecha_recoleccion	timestamp	Fecha estimada de cosecha.	"2025-07-31T17:25:00"
gastosFijos	Object	Costos recurrentes predefinidos:	
→ fertilizacion	{precio_vez: number, veces_ano: number}	Gasto en fertilizantes.	{precio_vez: 12, veces_ano: 3}
→ riego	{litrosPorVez: number, precio_litro: number, veces_semana: number }	Costos de riego.	{litrosPorVez: 12, precio_litro: 0.3, veces_semana: 7}
gastosVariables	Array< {tipo: string, importe: number, fecha: timestamp}>	Gastos imprevistos.	[{tipo: "Alambrada", importe: 200, fecha: "2025-05-27T21:46:59.422Z" }]

### 2. Parcela Ganadera

Campo	Tipo	Descripción	Ejemplo
especie	string	Tipo de ganado (ej: "vacas").	"vacas"
cabezasGanado	number	Número de animales.	120
fecha_sacrificio	timestamp	Fecha programada para sacrificio (si aplica).	"2025-09-06T17:26:00"
gastosFijos	Object	Costos fijos de mantenimiento:	
→ alimentacion	{veces_semana: number, precio_kilo: number, kilosPorVez: number}	Costos de alimento.	{veces_semana: 12, precio_kilo: 12, kilosPorVez: 23}
→ veterinario	{precio_vez: number, veces_ano: number}	Gastos veterinarios.	` {precio_vez:

## Administradores

Campo	Tipo	Descripción	Ejemplo
Id	string	ID único generado automáticamente.	"MISlzdVHXFnk7u4a3YIV"
uid	String	ID del usuario que es administrador	"tHAMQdFC9GNRBcuvJwqd7KUCuM83"



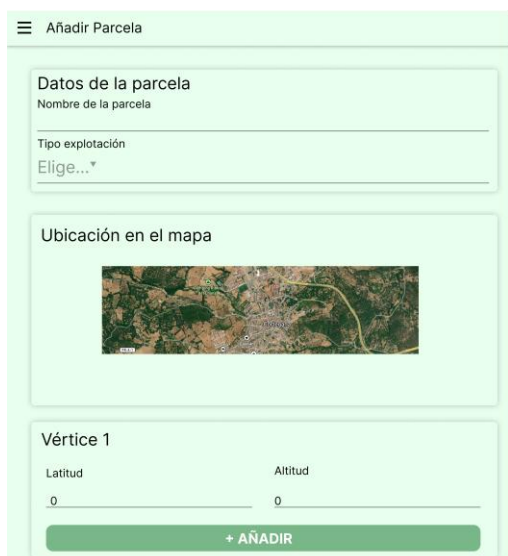
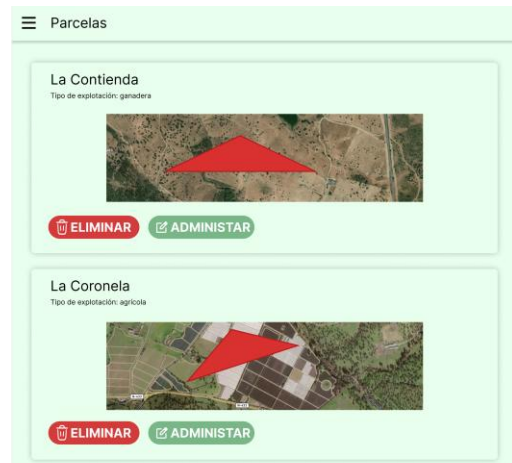
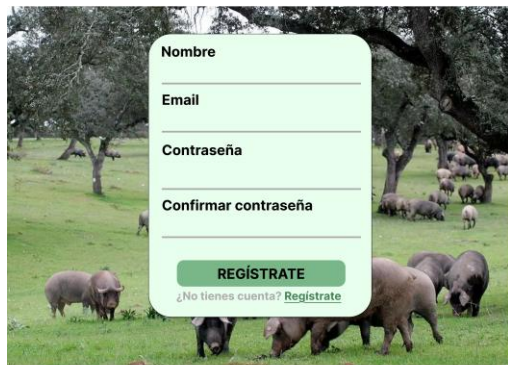
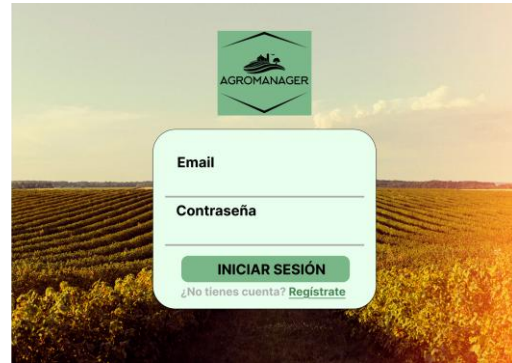
## Guía de estilos



Para mejor visualización:

<https://www.figma.com/design/QlieHURnu7K11gaOREsFY4/Untitled?node-id=0-1&t=nORFs93Srpmpnrfl-1>

## Prototipo



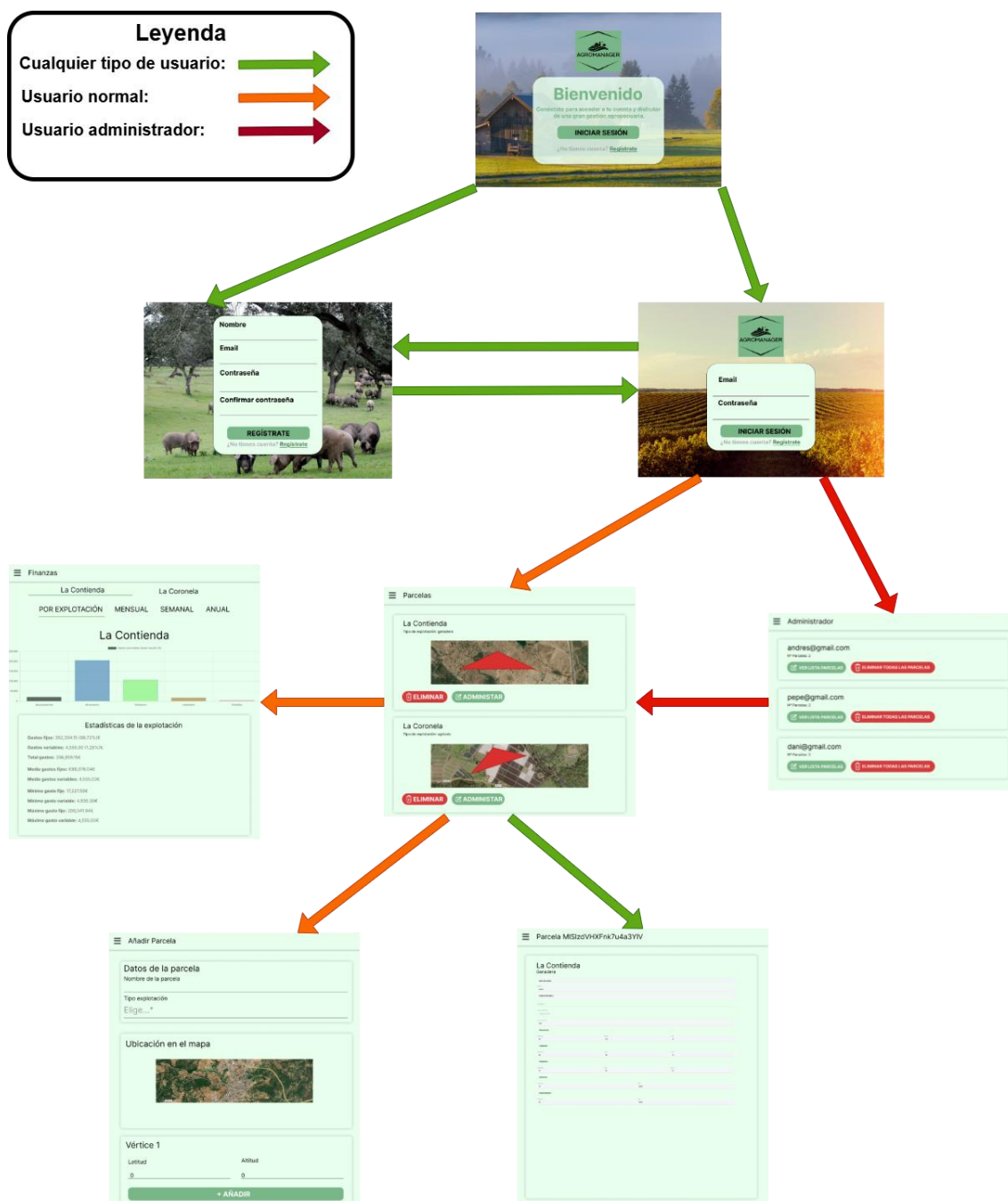
[illegible]

Administrador	
andres@gmail.com	Nº Parcelas: 2
<a href="#">VER LISTA PARCELAS</a>	<a href="#">ELIMINAR TODAS LAS PARCELAS</a>
pepe@gmail.com	Nº Parcelas: 2
<a href="#">VER LISTA PARCELAS</a>	<a href="#">ELIMINAR TODAS LAS PARCELAS</a>
dani@gmail.com	Nº Parcelas: 2
<a href="#">VER LISTA PARCELAS</a>	<a href="#">ELIMINAR TODAS LAS PARCELAS</a>

Para mejor visualización:

<https://www.figma.com/design/7iE8uNiIPdcVQg6VrUitMX/Untitled?node-id=0-1&t=I3kBy45hNquSAyFf-1>

## Flujo de navegación



Para mejor visualización: <https://drive.google.com/file/d/1f9INxpN-6DW05DxrWYah0os8FuiLOk9s/view?usp=sharing>

## Listado de los ficheros de la aplicación

Ya que Angular es un framework el cual crea muchos ficheros y la lista sería interminable voy a nombrar los componentes, páginas y servicios que yo he creado, junto con los ficheros que crea el propio angular y si se modifican.

- `app-routing.module.ts`: su función es definir las rutas de la aplicación y cargar los componentes.
- `app.module.ts`: es el módulo raíz de la aplicación. Aquí se configuran los componentes, servicios, módulos y dependencias globales que estarán disponibles en toda la app.
- `components.module.ts`: es un módulo de Angular/Ionic diseñado para agrupar y gestionar componentes reutilizables que pueden ser usados en diferentes partes de la aplicación.
- Componente administrador-parcelas: es un componente el cual carga los datos de una parcela determinada y permite su edición, a parte de añadir en los gastos variables para luego actualizar la base de datos.
- Componente lista-parcelas: es un componente el cual recibe una lista de parcelas y muestra cada una de ella con opciones e eliminar o administrar.
- Componente map: implementa un mapa interactivo usando Google Maps API, permitiendo visualizar parcelas (como polígonos) y marcadores, con funcionalidades para centrar el mapa en la ubicación del usuario o en coordenadas externas, dibujar parcelas almacenadas para el usuario autenticado, y agregar nuevos marcadores mediante clics (cuando está en modo edición), emitiendo eventos cuando se crean.
- Componente menu: es un menú lateral reutilizable en Ionic que muestra opciones configurables y un botón de logout, gestionando automáticamente la navegación y el cierre de sesión.
- `auth.guard.ts`: define una guardia de ruta llamada `privateGuard` que, al intentar acceder a una ruta protegida, verifica si el usuario está autenticado mediante un servicio (`AuthService`); si no lo está, lo redirige a la página de login y bloquea el acceso, y si sí lo está, permite el acceso.
- `AddParcelaPage`: permite a los usuarios añadir parcelas agrícolas o ganaderas definiendo sus vértices en un mapa y especificando distintos parámetros según el tipo de explotación, como especie, consumo de agua, alimentación o fertilización.

- AdminPage: es un panel de administración en Angular e Ionic donde se muestran usuarios que tienen parcelas registradas. Cada tarjeta muestra el correo del usuario, la cantidad de parcelas que posee y ofrece dos botones: uno para ver la lista de parcelas asociadas al usuario y otro para eliminarlas todas. Si no hay usuarios con parcelas, se muestra un mensaje indicando que no hay registros, junto con un botón para volver a la página de inicio. La información se carga al iniciar la página desde un servicio (ParcelasService).
- FinanzasPage: está diseñada para proporcionar una visualización y análisis financiero de una o varias explotaciones agrícolas.
- HomePage: es la página de inicio la cual nos puede dirigir al login o a registrarnos.
- LoginPage: es la página en la que se intentará iniciar sesión y en la cual se hará una llamada al servicio de login para que nos verifique si el Login es correcto o no. En caso de ser correcto nos llevará ParcelasPage en caso de ser un usuario corriente, en caso de ser un administrador te llevará a AdminPage y de no ser correcto mostrará un mensaje de error.
- ParcelasPage: en esta página se muestran todas las parcelas del usuario logueado y en cada una de ellas se da la opción de modificarlas o eliminarlas.
- RegistroPage: es la página en la cual se solicita la información necesaria para crear una cuenta en la app. Si la información es correcta redirecciona a login y si no lo es muestra un mensaje de error.
- AuthService: servicio el cual maneja la autenticación de usuarios usando Firebase Authentication y Firestore.
- GeolocalizacionService: utiliza la API de geolocalización de Capacitor para obtener la ubicación actual del dispositivo.
- ParcelasService: gestiona operaciones CRUD sobre las parcelas agrícolas en Firestore. Incluye métodos para agregar parcelas asociadas al usuario autenticado, obtener parcelas filtradas por usuario o todas las parcelas disponibles, eliminar parcelas individuales o por usuario, y actualizar información específica de una parcela.
- AuthStateService: servicio encargado de gestionar el estado de autenticación del usuario usando Firebase Authentication. Proporciona un observable "authState\$" para monitorear cambios en el estado de autenticación en tiempo real.
- variables.scss: contiene definiciones globales de variables CSS que se usan en toda la aplicación para mantener estilos consistentes y facilitar la personalización.
- global.scss: es el archivo global de estilos para una aplicación Ionic, que importa las bases necesarias para Ionic funcionar correctamente y añade estilos específicos para diversas páginas y componentes personalizados de la app.

## Despliegue

### Prerrequisitos

Antes de iniciar el despliegue, asegúrese de tener instaladas las siguientes herramientas:

- Node.js y npm.
- Angular CLI (npm install -g @angular/cli).
- Ionic CLI (npm install -g @ionic/cli).
- Capacitor (npm install @capacitor/core @capacitor/cli).
- Android Studio.

### Despliegue Web

Para hacer el despliegue web debemos hacer lo siguiente:

1. Ejecutar el build de producción en la terminal: ionic build --prod
2. El contenido se generará en el directorio /www
3. Subir el contenido a Firebase Hosting: firebase deploy --only hosting:  
<https://agromanager-e2ec7.web.app>

### Despliegue Android

1. Ejecutar el build de producción en la terminal: ionic build --prod.
2. Sincronizar con Capacitor: npx cap sync Android
3. Abrir Android Studio: npx cap open Android
4. Generar APK EN Android Studio: Build > Build Bundle(s) / APK(s)
5. Distribuir el APK: <https://github.com/2DAM-andresegi295/apk-AgroManager>

## Incidencias encontradas, pruebas realizadas y resolución.

Esta es una lista de incidencias que han ido surgiendo a lo largo del desarrollo de la aplicación:

1. Si no hay parcelas creadas, la página ParcelasPage sale completamente en blanco, lo cual no se considera muy intuitivo, nos damos cuenta de esto probando la app desde 0 con un usuario nuevo. La solución que ponemos es mostrar un mensaje de que no hay parcelas y que debe crearlas.
2. Si no hay parcelas creadas, la página FinanzasPage sale completamente en blanco, lo cual no se considera muy intuitivo, nos damos cuenta de esto probando la app desde 0 con un usuario nuevo. La solución que ponemos es mostrar un mensaje de que no hay datos financieros y lo reubicamos en parcelas con un botón.
3. Hacer el diseño responsive tanto para dispositivos web como para móvil no ha sido fácil. Nos dimos cuenta de esto tras usar la versión Android y lo solucionamos aplicando estilos distintos dependiendo de cual fuera el dispositivo.
4. Usando el dispositivo móvil nos dimos cuenta que la barra de notificaciones se superponía a la app, para solucionar este inconveniente modificamos el MainActivity del proyecto Android para que Android mantenga estable el diseño del contenido cuando aparezcan o desaparezcan elementos del sistema.
5. Al cerrar sesión no se borraban bien los datos cargados por el anterior usuario, de esto nos dimos cuenta iniciando sesión y cerrando sesión con distintos usuarios. Lo solucionando implementando el método logOut del fichero AuthService el cual borra el almacenamiento local, el almacenamiento de sesión y a parte luego de navegar a home recarga la página. Para asegurar así que se cierre la sesión.

## Posibles ampliaciones y mejoras

Como posibles mejoras nos gustaría implementar sistemas de filtrado y búsquedas en cada una de las listas que tenemos ya que esto aumentaría la usabilidad de la aplicación. También nos gustaría mejorar el menú de navegación de la app.

Como posible ampliación está la idea de poder implementar el monitoreo de animales y cultivos mediante sensores. Pudiendo así controlar el nivel de humedad de la tierra, el movimiento que hace cada animal, agua en los abrevaderos, calor, humedad ambiental, etc. Todo ello con el objetivo de poder ayudar a obtener una mayor rentabilidad al sector agropecuario.