

# Agro Asistencia

Pablo Rubio Nogales

Desarrollo de Aplicaciones Web

I.E.S Francisco Ayala

## Introducción

La agricultura moderna se enfrenta a constantes desafíos que exigen una gestión eficiente y optimizada de los recursos. Desde la planificación de tareas hasta el control de la maquinaria, los agricultores necesitan herramientas que les permitan tomar decisiones informadas y mejorar su productividad. Este Proyecto de Fin de Grado presenta **Agro Asistencia**, una aplicación web diseñada específicamente para satisfacer estas necesidades.

**Agro Asistencia** proporciona una solución integral para que los agricultores administren de manera efectiva sus trabajos y su maquinaria. La aplicación ha sido desarrollada utilizando un enfoque de arquitectura moderna, combinando **Laravel** para la construcción de una robusta API RESTful en el backend y **React** para la creación de una interfaz de usuario dinámica e intuitiva en el frontend. Esta sinergia tecnológica permite ofrecer una plataforma escalable, eficiente y fácil de usar, que contribuirá a la digitalización y optimización de las operaciones agrícolas. A lo largo de este documento, se detalla el proceso de desarrollo, las funcionalidades implementadas y los beneficios que **Agro Asistencia** aporta al sector agrícola.

<b>Introducción.....</b>	<b>1</b>
<b>Justificación.....</b>	<b>3</b>
Características Generales.....	3
Aspectos a Cubrir y No Cubrir.....	4
<b>Justificación de la Tecnología Empleada.....</b>	<b>5</b>
Frontend: React.....	5
Backend: Laravel.....	6
Investigación.....	7
<b>Requerimientos Hardware y Software.....</b>	<b>8</b>
Requerimientos del Cliente (Tu Dispositivo).....	8
Requerimientos del Servidor.....	8
<b>Análisis y Diseño.....</b>	<b>9</b>
Casos de Uso.....	9
Diagrama de Clases.....	10
<b>Manual de estilos.....</b>	<b>13</b>
Tipografía.....	13
Principios de Diseño y Componentes Clave.....	15

## Justificación

La necesidad de optimizar la gestión en el sector agrícola es más evidente que nunca. La complejidad de las operaciones, la variabilidad climática y la constante evolución tecnológica exigen soluciones que permitan a los agricultores controlar sus recursos de manera eficiente. **Agro Asistencia** surge como respuesta a esta demanda, buscando proporcionar una herramienta digital que simplifique y mejore la administración de las tareas y la maquinaria agrícola, facilitando la toma de decisiones y potenciando la productividad.

## Características Generales

**Agro Asistencia** es una aplicación web diseñada para ser una plataforma centralizada de gestión para el agricultor. Sus características principales se centran en la administración de dos pilares fundamentales de la actividad agrícola:

- **Gestión de Trabajos:** Permite al agricultor registrar, planificar y supervisar todas las tareas agrícolas. Se pueden asignar trabajos a distintas fechas para mantener un control de tu trabajo.
- **Gestión de Maquinaria:** Ofrece un apartado para el inventario y seguimiento de la maquinaria agrícola. Los usuarios pueden registrar datos de cada máquina (tipo, modelo, etc.).

La arquitectura de la aplicación, basada en **Laravel** para el backend y **React** para el frontend, garantiza una experiencia de usuario fluida y una alta capacidad de respuesta. Esto permite que la aplicación sea accesible desde cualquier dispositivo con conexión a internet, brindando flexibilidad y movilidad al agricultor.

## Aspectos a Cubrir y No Cubrir

**Agro Asistencia** ha sido concebida para cubrir aspectos esenciales de la gestión agrícola, con un enfoque en la usabilidad y la eficiencia.

### Aspectos a Cubrir:

- **Planificación y Seguimiento de Tareas:** Proporcionar una visión clara de las actividades agrícolas.
- **Inventario y Gestión de Maquinaria:** Ofrecer un registro detallado y la capacidad de una vista detallada para la flota de maquinaria.
- **Inventario y Gestión de Fitosanitarios:** Ofrecer un registro detallado y la capacidad de una vista detallada de fitosanitarios.
- **Inventario y Gestión de Fincas:** Ofrecer un registro detallado con geolocalización y la capacidad de una vista detallada para todas las fincas.
- **Análisis de Fitosanitarios:** Ofrece un listado de los tres fitosanitarios más usado por los usuarios en nuestra aplicación
- **Interfaz de Usuario Intuitiva:** Facilitar la interacción del agricultor con la plataforma, incluso para aquellos con conocimientos tecnológicos limitados.
- **Modularidad y Escalabilidad:** Diseñar una arquitectura que permita futuras ampliaciones de funcionalidades y la integración con otros sistemas si fuera necesario.

### Aspectos No Cubiertos (Alcance del Proyecto):

Para mantener el enfoque y la viabilidad del proyecto de fin de grado, se han delimitado ciertos aspectos que no serán abordados en esta primera versión de **Agro Asistencia**:

- **Gestión Financiera Avanzada:** No incluirá funcionalidades complejas de contabilidad, nóminas o gestión de ingresos/gastos.
- **Análisis de Datos Agronómicos Profundos:** No ofrecerá herramientas de análisis de datos geoespaciales o predicciones basadas en modelos agronómicos complejos.
- **Mercados o Comercialización:** No incluirá funcionalidades relacionadas con la compra-venta de productos agrícolas.

## Justificación de la Tecnología Empleada

La elección de las tecnologías para el desarrollo de **Agro Asistencia** se basó en la búsqueda de robustez, escalabilidad, eficiencia en el desarrollo y una experiencia de usuario óptima. Tras un análisis de las opciones disponibles y considerando las características del proyecto, se optó por una arquitectura basada en **Laravel** para el backend y **React** para el frontend.

### Frontend: React

Para el desarrollo del frontend de **Agro Asistencia**, se seleccionó **React**, una biblioteca de JavaScript declarativa, eficiente y flexible para construir interfaces de usuario. Las razones clave para esta elección incluyen:

- **Rendimiento y Experiencia de Usuario (UX):** React permite construir **Single Page Applications (SPAs)**, lo que resulta en una experiencia de usuario fluida y rápida. Al cargar la aplicación una sola vez y actualizar sólo los componentes necesarios, se reduce significativamente el tiempo de carga y la latencia, lo cual es crucial para una herramienta de gestión donde la inmediatez es valorada.
- **Componentización:** Su enfoque basado en componentes facilita la creación de interfaces de usuario complejas a partir de piezas pequeñas y reutilizables. Esto mejora la **modularidad del código**, la **mantenibilidad** y la **escalabilidad** del proyecto, permitiendo añadir nuevas funcionalidades de manera más sencilla en el futuro.
- **Amplia Comunidad y Ecosistema:** React cuenta con una de las comunidades de desarrolladores más grandes y activas, lo que se traduce en una vasta cantidad de recursos, librerías, herramientas y soporte en línea. Esto agiliza el proceso de desarrollo y la resolución de posibles problemas.
- **Popularidad y Demanda en el Mercado Laboral.**

## Backend: Laravel

Para la construcción de la API RESTful que sirve de base a **Agro Asistencia**, se eligió **Laravel**, un framework de PHP de código abierto. La justificación de esta elección se basa en:

- **Productividad y Velocidad de Desarrollo:** Laravel es conocido por su **sintaxis expresiva y elegante**, así como por su vasta colección de herramientas y paquetes pre-construidos (ORM Eloquent, Artisan CLI, autenticación integrada, etc.). Esto permite a los desarrolladores construir aplicaciones robustas de manera más rápida y eficiente, reduciendo los tiempos de desarrollo.
- **Ecosistema Robusto para APIs:** Laravel está excelentemente equipado para la creación de **APIs RESTful**, ofreciendo funcionalidades como routing, middleware, controladores y validación de peticiones HTTP de forma nativa.
- **Seguridad:** Laravel incorpora características de seguridad robustas de forma predeterminada, como protección contra inyección SQL, scripting entre sitios (XSS) y falsificación de peticiones entre sitios (CSRF), lo que contribuye a la creación de una aplicación más segura y confiable.
- **Escalabilidad:** Gracias a su diseño modular y su capacidad para integrarse con sistemas de cacheo y colas, Laravel es adecuado para construir aplicaciones que pueden crecer en tamaño y demanda.
- **Comunidad y Documentación:** Al igual que React, Laravel goza de una gran comunidad y una documentación oficial excepcionalmente buena y completa, lo que facilita el aprendizaje y la resolución de dudas durante el desarrollo.

## Investigación

La elección de **React** y **Laravel** no fue aleatoria, se realizó un proceso de investigación y análisis comparativo. Se evaluaron otras opciones populares en el mercado, como Vue.js y Node.js.

La investigación se centró en:

- **Curva de aprendizaje:** Se consideró la facilidad con la que se podrían adquirir los conocimientos necesarios para utilizar eficientemente cada tecnología. React y Laravel ofrecían un equilibrio óptimo entre potencia y una curva de aprendizaje manejable para un proyecto de fin de grado.
- **Rendimiento:** Se comparó el potencial de rendimiento de cada tecnología para manejar la carga de datos y las interacciones de usuario esperadas en una aplicación de gestión.
- **Ecosistema y Soporte:** La disponibilidad de librerías, paquetes, herramientas de depuración y la actividad de la comunidad fueron factores determinantes para asegurar un desarrollo fluido y el soporte a largo plazo.

La combinación de **Laravel y React** fue seleccionada por su capacidad para ofrecer un **rendimiento óptimo**, una **excelente experiencia de usuario**, **velocidad en el desarrollo** y una **sólida base para la escalabilidad** futura de **Agro Asistencia**. Esta elección estratégica garantiza que el proyecto no solo cumpla con los requisitos funcionales actuales, sino que también esté preparado para futuras evoluciones y mejoras.



## Requerimientos Hardware y Software

Para el correcto funcionamiento de **Agro Asistencia**, tanto en el lado del cliente (usuario final) como en el lado del servidor (donde reside la aplicación), se necesitan cumplir con ciertos requisitos de hardware y software. Estos requerimientos están pensados para garantizar un rendimiento óptimo y una experiencia de usuario fluida.

### Requerimientos del Cliente (Tu Dispositivo)

Solo necesitas un navegador web actualizado, ya que la aplicación funciona online.

- **Hardware:** Cualquier ordenador o dispositivo móvil moderno (smartphone o tablet) y una pantalla de resolución media.
- **Conexión a Internet:** Una conexión a internet **estable y rápida**.
- **Software:** Un **navegador web moderno y actualizado** como Google Chrome, Mozilla Firefox, Microsoft Edge o Apple Safari.

### Requerimientos del Servidor

El servidor es donde vive la aplicación y la base de datos. Para un proyecto de esta escala:

- **Hardware:**
  - **CPU:** Al menos 2 núcleos de procesador.
  - **RAM:** Mínimo 4 GB de RAM, idealmente 8 GB o más para entornos de producción.
  - **Almacenamiento:** Mínimo 20 GB de espacio en disco (preferiblemente SSD).
  - **Red:** Conexión de red de al menos 100 Mbps.
- **Software:**
  - **Sistema Operativo:** Una distribución de **Linux** (como Ubuntu LTS).
  - **Servidor Web:** Apache.
  - **PHP:** Versión **8.1 o superior** con las extensiones necesarias para Laravel.
  - **Base de Datos:** **MySQL 8.0+**.
  - **Herramientas:** **Composer** (para gestionar dependencias de PHP en Laravel) y **Node.js con npm/Yarn** (para construir el frontend de React).

## Análisis y Diseño

Esta sección detalla el análisis de los requisitos funcionales de Agro Asistencia y el diseño de su arquitectura. Se presentarán los casos de uso principales, un diagrama de clases que ilustra la estructura de la aplicación y el diseño de la base de datos, elementos fundamentales para la construcción de una aplicación robusta y escalable.

### Casos de Uso

Los casos de uso describen las interacciones entre los actores (el **Agricultor** y el **Sistema**) y las funcionalidades que **Agro Asistencia** ofrecerá.

- Registro y visualización de trabajos agrícolas diarios.
- 
- Gestión de agricultores (visibles para otros usuarios).
- 
- Registro y edición de fincas personales
- 
- Consulta y sugerencia de productos fitosanitarios
- 
- Control de maquinaria y recursos
- 
- Sistema personalizado de trabajos agrícolas
- 
- Clima integrado con mapa en portada.

## Diagrama de Clases

El diagrama de clases representa la estructura estática de la aplicación, mostrando las clases principales, sus atributos, métodos y las relaciones entre ellas. A continuación, se describen las clases esenciales que conformarán el modelo de datos y la lógica de negocio de **Agro Asistencia**.

```
class agricultores {  
    +id_agricultor: bigint  
    +DNI: varchar  
    +nombre: varchar  
    +apellidos: varchar  
    +telefono: varchar  
    +localidad: varchar  
    +created_at: timestamp  
    +updated_at: timestamp  
}
```

```
class usuarios {  
    +id_usuario: bigint  
    +id_agricultor: bigint  
    +nombre: varchar  
    +direccion: varchar  
    +email: varchar  
    +telefono: varchar  
    +contraseña: varchar  
    +created_at: timestamp  
    +updated_at: timestamp  
}
```

```
class maquinaria {  
    +id_maquinaria: bigint  
    +id_usuario: bigint  
    +nombre: varchar  
    +tipo: varchar  
    +año: varchar  
    +marca: varchar  
    +modelo: varchar  
    +numero_serie: varchar  
    +created_at: timestamp  
    +updated_at: timestamp  
}
```

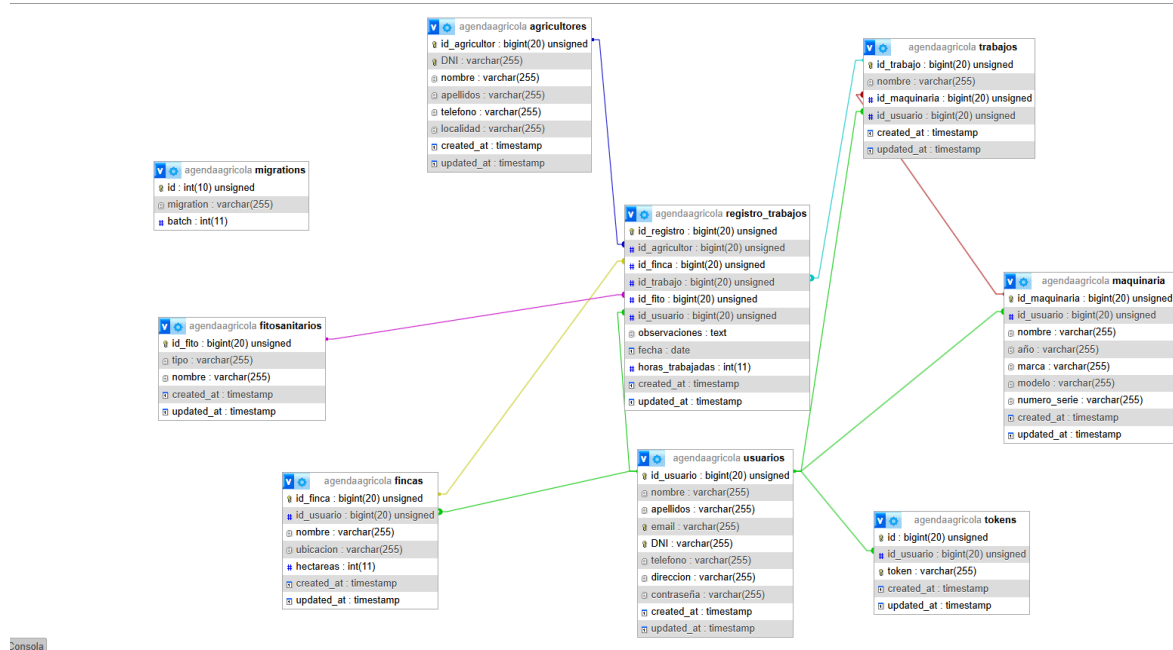
```
class trabajos {  
    +id_trabajo: bigint  
    +nombre: varchar  
    +tipo: varchar  
    +descripcion: varchar  
    +id_maquinaria: bigint  
    +created_at: timestamp  
    +updated_at: timestamp  
}
```

```
class fincas {  
  +id_finca: bigint  
  +id_usuario: bigint  
  +nombre: varchar  
  +ubicacion: varchar  
  +hectareas: int  
  +created_at: timestamp  
  +updated_at: timestamp  
}
```

```
class fitosanitarios {  
  +id_fito: bigint  
  +tipo: varchar  
  +nombre: varchar  
  +created_at: timestamp  
  +updated_at: timestamp  
}
```

```
class registro_trabajos {  
  +id_registro: bigint  
  +id_agricultor: bigint  
  +id_finca: bigint  
  +id_trabajo: bigint  
  +id_fito: bigint  
  +id_usuario: bigint  
  +observaciones: text  
  +fecha: date  
  +horas_trabajadas: int  
  +created_at: timestamp  
  +updated_at: timestamp  
}
```

## Base de datos



El diseño de la base de datos se basa directamente en tu esquema provisto, utilizando MySQL como gestor. Se describen las tablas y sus relaciones clave para la gestión de la agro asistencia.

## Manual de estilos

### Tipografía

La tipografía es fundamental para la legibilidad y la experiencia del usuario. Se ha optado por un conjunto de fuentes que aseguran claridad en diversos tamaños de pantalla.

- **Fuentes Principales:**
  - 'Segoe UI' (como fuente principal)
  - 'Arial' (como fuente secundaria, especialmente en formularios)
  - Tahoma, Geneva, Verdana, sans-serif (como fuentes de respaldo)
- **Jerarquía y Uso:**
  - **Títulos (h2, h3):** Se utilizan tamaños de fuente más grandes y pesos semi gruesos (`font-weight: 600`) para establecer una clara jerarquía y llamar la atención sobre secciones importantes, como títulos de formularios o tablas.
    - **Ejemplos de tamaños:** 2.5rem (escritorio), 2.2rem (tablet), 1.8rem (móvil) para títulos principales; 2em (escritorio), 1.8em (tablet), 1.6em (móvil) para títulos de formularios.
  - **Etiquetas de Formulario (label):** `font-weight: bold` o 500 con un tamaño de fuente de 14px (escritorio) para asegurar que sean claramente visibles y fáciles de asociar con sus campos correspondientes.
  - **Contenido General y Entradas (input, select, textarea, texto de tabla):** 16px o 1em (escritorio) para una óptima legibilidad del cuerpo de texto. Se ajusta dinámicamente en tamaños de pantalla más pequeños.
  - **Mensajes (éxito/error):** Tamaños ligeramente ajustados para destacar, pero sin ser intrusivos.

### Paleta de Colores

La paleta de colores ha sido seleccionada para ofrecer una interfaz atractiva y fácil de usar, con énfasis en la claridad y la retroalimentación visual.

- **Colores Primarios / de Interacción:**
  - **Azul (#007bff, #42a5f5, #0056b3, #308ec0):** Utilizado para botones de acción principal, enlaces y estados de enfoque (`focus`) en campos de formulario. Transmite confianza y usabilidad.
  - **Verde (#4CAF50, #45a049, #388E3C, #5cb85c):** Predominante en botones de éxito, mensajes de éxito y estados de enfoque. Simboliza éxito, aprobación y acción positiva.

- **Verde Teal (#10a88f):** Usado para el encabezado de las tablas, dando un toque distintivo y profesional.
- 
- **Colores de Fondo y Superficie:**
    - **Blanco (#ffffff):** Fondo principal de las tarjetas, formularios y modales, proporcionando un lienzo limpio para el contenido.
    - **Gris Claro (#f9fafb, #fdfdfd, #f2f2f2):** Utilizado para fondos de contenedores y filas alternas en tablas, mejorando la distinción visual sin ser abrumador.
- 
- **Colores de Texto:**
    - **Gris Oscuro (#333, #2c3e50):** Para títulos y texto principal, asegurando una excelente legibilidad sobre fondos claros.
    - **Gris Medio (#555, #666):** Para etiquetas de formulario y texto secundario, suavizando la interfaz.
- 
- **Colores de Retroalimentación y Estado:**
    - **Rojo (#FF5252, #dc3545, #a71d2a, #c62828, #d9534f):** Utilizado para mensajes de error y botones de acción "peligrosa" (ej. eliminar), alertando al usuario de problemas o acciones irreversibles.
    - **Amarillo/Naranja (#f0ad4e, #eea236):** Para botones de cancelar o acciones de advertencia en modales.
- 
- **Bordes y Sombras:**
    - **Grises Suaves (#ddd, #e0e0e0, #cfd8dc):** Utilizados para bordes de campos de entrada, tarjetas y divisores de tabla, aportando estructura sin ser demasiado llamativos.
    - **Sombras (rgba(0, 0, 0, 0.1), rgba(0, 0, 0, 0.2)):** Sombras sutiles se aplican a contenedores como formularios y modales para dar una sensación de profundidad y separación del fondo.
-

## Principios de Diseño y Componentes Clave

El diseño CSS refleja un enfoque que prioriza la usabilidad y la adaptabilidad.

- **Diseño Fluido y Responsive (Mobile First):**

- La mayoría de los elementos comienzan con estilos optimizados para dispositivos móviles (tamaños de fuente, padding, márgenes reducidos; elementos apilados en lugar de en fila).
- Mediante **Media Queries**, los estilos se adaptan progresivamente para tabletas y escritorios, restaurando o aumentando tamaños, padding y la disposición de los elementos (ej. `flex-direction: row` para contenedores en pantallas más grandes).
- Las tablas en móvil se hacen desplazables horizontalmente (`overflow-x: auto`) para asegurar la visibilidad de todas las columnas.

- **Espaciado y Consistencia:**


- Se utiliza un sistema de espaciado consistente (ej. `gap` en flexbox, `margin-bottom` en grupos de formulario) para asegurar una interfaz ordenada y predecible.
- El `padding` en los contenedores se ajusta para maximizar el espacio útil en pantallas pequeñas y ofrecer una estética más "aireada" en pantallas grandes.

- **Elementos Interactivos:**

- **Botones:** Presentan un `padding` generoso, `border-radius` sutil y transiciones suaves (`background-color`, `transform`) al interactuar, proporcionando una retroalimentación visual clara.
- **Campos de Formulario:** Tienen `padding` adecuado y bordes discretos que cambian de color y añaden una sombra suave al enfocar, guiando al usuario y mejorando la usabilidad.
- **Selects Personalizados:** Se les ha dado una apariencia más moderna eliminando la flecha por defecto y añadiendo una SVG personalizada.

- **Feedback al Usuario:**



- 
- Mensajes de **éxito** y **error** claramente diferenciados por color de texto y fondo, con bordes para hacerlos prominentes y fáciles de identificar.
- **Modales:**
    - Centrados en la pantalla con un `transform: translate(-50%, -50%)` para una posición consistente.
    - Utilizan un `overlay-bloqueo` oscuro para enfocar la atención del usuario en el contenido del modal.
    - Los formularios dentro de modales también siguen los principios de diseño responsive y usan el `box-sizing: border-box` para un control preciso del tamaño.
- 

Este conjunto de estilos crea una experiencia de usuario cohesiva y adaptable, diseñada para funcionar eficazmente en una amplia gama de dispositivos.