



Universidad de Valladolid

Escuela de Ingeniería Informática (SG)

**Grado de Ingeniería Informática de Servicios
y Aplicaciones.**

AgroPlan

Desarrollo de una aplicación para la Planificación Agrícola

Alumno: Pablo Llorente Senín

Tutor: Fernando Díaz Gómez

Resumen

El presente proyecto, AgroPlan, es una herramienta desarrollada para ayudar a agricultores en lo que se refiere a la planificación, gestión y prevención de anomalías de sus cultivos. AgroPlan proporciona información personalizada y detallada sobre cada etapa del proceso del cultivo de plantas, desde el labrado hasta la cosecha de los cultivos, pasando por el tratamiento de enfermedades y plagas.

El sistema contará con una base de datos escalable que cubre los cultivos más comunes en España con proyección de expansión futura para abarcar nuevas siembras, regiones y condiciones climáticas. Además el sistema mostrará al usuario historiales y antecedentes en sus tierras, para prevenir nuevos riesgos y contar con un seguimiento.

AgroPlan tiene como objetivo reconocerse como un nuevo paso hacia la modernización del sector primario, y haciendo así que la ingeniería se sitúe más accesible a los pequeños agricultores mediante una solución útil y adaptable.

Palabras clave: Agricultura, cultivos, tierras, gestión, planificación, ingeniería, semilla.

Índice de contenidos

Capítulo 1

1.1 Estructura de la documentación.....	12
1.2 Introducción.....	14
1.3 Motivación.....	15
1.4 Objetivos.....	16
1.5 Entorno de la aplicación.....	17
1.6 Tecnologías utilizadas.....	17
Java 8.....	17
PostgreSQL.....	18
ReactJS.....	18
Github.....	18
Trello.....	19
Draw.io.....	19
Visual Studio Code.....	19
Node js.....	20
Npm.....	20

Capítulo 2

2.1 Herramientas existentes.....	21
Agroptima.....	21
AGRIVI.....	21
Kaampo.....	22
Efemis.....	22
Farmable.....	22
2.2 Comparativa.....	24

Capítulo 3

3.1 Metodología de trabajo.....	25
Ventajas e inconvenientes.....	26
3.2 Planificación temporal.....	28
3.3 Presupuesto económico.....	31
3.4 Balance final.....	33

Capítulo 4

4.1 Alcance del sistema.....	34
4.2 Actores del sistema.....	34
4.3 Requisitos de usuario.....	35
4.4 Diagrama de casos de uso.....	37

Especificación de casos de uso.....	38
4.5 Requisitos funcionales.....	39
4.6 Requisitos no funcionales.....	41
Usabilidad.....	41
Eficiencia.....	41
Mantenibilidad.....	41
Seguridad.....	42
Disponibilidad.....	42
Portabilidad.....	43
Escalabilidad.....	43
4.7 Requisitos de información.....	44
4.8 Modelo E-R.....	45
4.9 Diccionario de datos.....	46
Capítulo 5	
5.1 Estructura de la información.....	50
5.2 Modelo lógico de datos.....	53
5.3 Arquitectura física.....	55
5.4 Arquitectura lógica detallada.....	56
5.5 Arquitectura lógica no detallada.....	57
5.6 Tipo de modelo.....	58
5.7 Diagrama de secuencia.....	60
Capítulo 6	
6.1 Backend (capa de negocio).....	62
6.2 Frontend (lado cliente).....	65
Capítulo 7	
7.1 Pruebas de caja negra.....	67
7.2 Pruebas de caja blanca.....	70
Capítulo 8	
8.1 Manual de instalación.....	72
8.2 Manual de usuario.....	75
Capítulo 9	
9.1 Futuras ampliaciones AgroPlan:.....	89

Índice de figuras

Capítulo 1

1.2 Introducción.....	11
Imagen: logo AgroPlan.....	11
1.6 Tecnologías utilizadas.....	14
Imagen: Logo Java.....	15
Imagen: Logo PostgreSQL.....	15
Imagen: Logo ReactJS.....	15
Imagen: Logo Github.....	16
Imagen: Logo Trello.....	16
Imagen: Logo Draw.io.....	16
Imagen: Logo Visual Studio Code.....	17
Imagen: Logo Node js.....	17

Capítulo 2

2.1 Herramientas existentes.....	18
Imagen: Logo “Agroptima”.....	18
Imagen: Logo “Agrivi”.....	19
Imagen: Logo “Kaampo”.....	19
Imagen: Logo “Efemis”.....	19
Imagen: Logo “Farmable”.....	20

Capítulo 3

3.1 Metodología de trabajo.....	22
Imagen: Estructura Agile.....	22
Imagen: Tablero “Agroplan-trello”, Trello.....	22
Imagen: Tablero “Agroplan-trello-documentación”, Trello.....	23

Capítulo 4

Diagrama de características.....	31
4.4 Diagrama de casos de uso.....	34
Diagrama de casos de uso de AgroPlan.....	34
4.8 Modelo E-R.....	42
Modelo entidad relación.....	42

Capítulo 5

5.2 Modelo lógico de datos.....	50
Diagrama: Modelo lógico de datos.....	51
5.3 Arquitectura física.....	52
Diagrama físico.....	52
5.4 Arquitectura lógica detallada.....	53
Diagrama lógico detallado.....	53
5.5 Arquitectura lógica no detallada.....	54

Diagrama lógico no detallado.....	54
5.6 Tipo de modelo.....	55
Imagen: Estructura del entorno de la aplicación.....	55
5.7 Diagrama de secuencia.....	57
Diagrama de secuencia “Iniciar sesión”.....	57
Diagrama de secuencia “Sembrar tierra”.....	58
Diagrama de secuencia “Comprobación de Temp y pH”.....	58

Capítulo 6

6.1 Backend (capa de negocio).....	59
Imagen: Carpeta del proyecto.....	59
Imagen: paquetería Entities y Repositories.....	60
Imagen: Repository ejemplo de AgroPlan.....	60
Imagen: Paquetería Controller.....	61
Imagen: Paquetería operations.....	61
Imagen: Paquetería Utils.....	61
6.2 Frontend (lado cliente).....	62
Imagen: Estructura del front.....	62
Imagen: Uso de Axios.....	62

Capítulo 8

8.2 Manual de usuario.....	72
Imagen: Manual de usuario “Login”.....	72
Imagen: Formularios Inicio sesión y registro.....	72
Imagen: Listado de labores del agricultor.....	73
Imagen: Barra navegación listado labores.....	73
Imagen: Listado “Sembradas” labores.....	74
Imagen: Listado “Próximas a sembrar” labores.....	74
Imagen: Listado “Cosechadas” labores.....	74
Imagen: Fila labor, botón cosechar.....	75
Imagen: Barra navegación lateral.....	75
Imagen: Listado propiedades agricultor.....	75
Imagen: Modal registro nueva propiedad.....	76
Imagen: Modal edición propiedad.....	76
Imagen: Indicador “tareas por hacer”.....	76
Imagen: Modal “tareas por hacer”.....	77
Imagen: Modal confirmación “Eliminar tierra”.....	78
Imagen: Modal “Tierra no eliminada”.....	78
Imagen: Botones detectar plaga y enfermedad.....	78
Imagen: Modal detección de plagas.....	79
Imagen: Modal detección de enfermedades.....	79
Imagen: Listado antecedentes tierra.....	80
Imagen: Listado historial tierra.....	80
Imagen: Formulario comprobar estado tierra.....	81

Imagen: Resultado comprobar estado tierra.....	81
Imagen: Listado semillas.....	82
Imagen: Título detalle semilla.....	83
Imagen: Detalle información acerca de la semilla.....	83
Imagen: Periodos de siembra y cosecha semilla.....	83
Imagen: Información sobre posibles plagas y enfermedades.....	84
Imagen: Modal sembrar tierra.....	85

Índice de tablas

Capítulo 1

1.4 Objetivos.....	16
Tabla objetivos de Agroplan.....	16

Capítulo 2

2.2 Comparativa.....	24
Tabla: Comparativa Estado del Arte.....	24

Capítulo 3

3.3 Estimación del esfuerzo (puntos de función).....	30
3.4 Presupuesto económico.....	31

Capítulo 4

4.2 Actores del sistema.....	34
Tabla: Actores del sistema.....	35
4.3 Requisitos de usuario.....	35
Tabla: Requisitos de usuario no logueado.....	35
Tabla: Requisitos de usuario logueado.....	36
4.4 Diagrama de casos de uso.....	37
Tabla: Especificación Caso de uso “Sembrar semilla”.....	38
Tabla: Especificación Caso de Uso “Comprobar pH y Temperatura”.....	38
Tabla: Especificación Caso de Uso “Detectar plaga o enfermedad”.....	39
4.5 Requisitos funcionales.....	39
Tabla: Requisitos funcionales.....	41
4.6 Requisitos no funcionales.....	41
Tabla: Requisitos no funcionales de usabilidad.....	41
Tabla: Requisitos no funcionales de eficiencia.....	41
Tabla: Requisitos no funcionales de mantenibilidad.....	42
Tabla: Requisitos no funcionales de seguridad.....	42
Tabla: Requisitos no funcionales de disponibilidad.....	43
Tabla: Requisitos no funcionales de portabilidad.....	43
Tabla: Requisitos no funcionales de escalabilidad.....	43
4.7 Requisitos de información.....	44
Tabla: Requisitos de información.....	44
4.9 Diccionario de datos.....	46
Diccionario de datos: Entidad agricultor.....	46
Diccionario de datos: Entidad tierra.....	46
Diccionario de datos: Entidad planta.....	47
Diccionario de datos: Entidad semilla.....	47
Diccionario de datos: Entidad necesidad planta.....	47
Diccionario de datos: Entidad tierra sembrada.....	48
Diccionario de datos: Entidad abono.....	48

Diccionario de datos: Entidad labrado.....	48
Diccionario de datos: Entidad minerales.....	49
Diccionario de datos: Entidad enfermedades.....	49
Diccionario de datos: Entidad plagas.....	49
Diccionario de datos: Entidad fertilizantes.....	49
Diccionario de datos: Entidad labor.....	50

Capítulo 7

7.1 Pruebas de caja negra.....	67
Tabla: Pruebas caja negra “Registro nuevo usuario”	67
Tabla: Pruebas caja negra “Inicio sesión”	68
Tabla: Pruebas caja negra “Agricultor crea tierra”	68
Tabla: Pruebas caja negra “Agricultor siembra tierra”	68
Tabla: Pruebas caja negra “Agricultor cosecha tierra”	69
Tabla: Pruebas caja negra “Comprobar temperatura y ph”	69
Tabla: Pruebas caja negra “Detectar plaga o enfermedad”	70

Capítulo 1: Descripción de Agroplan

1.1 Estructura de la documentación

Se detalla aquí en este apartado, la estructura de este documento separado por capítulos.

- Capítulo 1: Descripción del proyecto

Se detalla una visión general del proyecto y su motivo de existir. Se explican los objetivos y el alcance del mismo, es decir, aquí se explica el problema que se resolverá.

- Capítulo 2: Estado del arte

Investigación y enumeración de soluciones similares a AgroPlan, que ya existen en el mercado, por lo tanto, competencia. Explicación de las funcionalidades de cada proyecto y finalmente una visual comparativa con nuestro proyecto.

- Capítulo 3: Planificación del proyecto

Se detalla el plan de organización del desarrollo de AgroPlan. Planificación de tiempo, de esfuerzo y por último de coste.

- Capítulo 4: Análisis del sistema

Análisis de los requisitos que definen las diferentes funcionalidades del proyecto, también los actores involucrados y casos de uso.

- Capítulo 5: Diseño del sistema

Especificación de la arquitectura del sistema y diseño de la base de datos. Diccionario de datos y estructura interna de AgroPlan.

- Capítulo 6: Implementación

Se explica cómo se lleva a cabo el desarrollo del sistema y su despliegue. Explicadas las herramientas utilizadas y los lenguajes de programación.

- Capítulo 7: Pruebas

Pruebas realizadas en el sistema, así como los resultados obtenidos y correcciones.

- Capítulo 8: Manuales

Manual de usuario y manual técnico. Instrucciones claras para el uso de AgroPlan, instalación y mantenimiento del sistema.

- Capítulo 9: Conclusiones

Comparativa explicada de los resultados esperados y obtenidos. Cumplimiento de objetivos, dificultades encontradas y mejoras o ampliaciones futuras.

Anexo: Referencias

Lista de todas las fuentes de información utilizadas para el desarrollo del proyecto. Fuentes de información para conformar la base de datos.

1.2 Introducción

La tecnología y, más precisamente, la ingeniería, es lo que nos hace a las personas ser mejores, con menos esfuerzo, y es ese ingenio el que nos lleva a crear herramientas que nos hagan las tareas más llevaderas.

Existe ingeniería en cualquier actividad en que nos fijamos, en un puente para cruzar un río, en una silla de oficina, en una carretera, etc.

Estoy seguro de que muchas personas, cuando escuchan hablar de sector primario, tanto de agricultura como de ganadería, piensan en métodos antiguos, “de pueblo”, o rudimentario. Pero se debe de saber que al igual que en la mayoría de los sectores, hay ingenieros e ingenieras haciendo que hasta las tareas “de toda la vida”, se simplifiquen descubriendo métodos que perfeccionan y aceleran procesos.

Presento así una herramienta llamada **AgroPlan**, pensado para el sector de la agricultura, que busca ayudar al agricultor/a a gestionar sus labores de cultivo, permitiéndoles toda la información necesaria acerca de la gestión de sus tierras. El agricultor/a dispondrá en cada etapa de su cultivo de toda la información necesaria acerca del siguiente paso a dar: labrado, siembra, tratamiento ante enfermedades o plagas, cosecha, etc.

Esta herramienta tendrá que contener información de todos los cultivos posibles (inicialmente en España), periodos de siembra, periodos de cosecha, información acerca de calidades de la tierra, enfermedades, plagas o abonos recomendados.

El objetivo con esta propuesta, es garantizar al agricultor una mayor eficiencia en sus labores, y que ante cualquier adversidad, se subsane en el menor tiempo posible.



Imagen: logo AgroPlan.

1.3 Motivación

Durante la carrera, en los meses de verano he trabajado en el campo como peón de agricultor. Lo que sembramos era principalmente zanahorias y puerros, y aún teniendo maquinaria avanzada, técnicas de siembra y de cuidado avanzados, siempre pensaba que seguro había algo que se pudiera mejorar, en lo que respecta a la organización de las actividades propias del agricultor para cada cultivo.

Siempre imaginé, que sería posible crear una herramienta que de alguna forma, permitiese tener el control sobre el estado de las fincas y no solo guardar registro de todas las decisiones tomadas sobre la tierra (abonos, riegos, semillas...), sino también, que contando con la información necesaria, anticiparse y estar preparados ante cualquier contratiempo (enfermedades, plagas, etc).

El jefe de la empresa donde trabajaba, me contó que también tenían cultivos de zanahorias en Cáceres (Extremadura), y las condiciones climáticas allí eran distintas. Asimismo, otro factor a tener en cuenta en el desarrollo de esta herramienta para que se pueda gestionar el cultivo de la tierra, indiferentemente de donde se localiza el agricultor.

1.4 Objetivos

El proyecto de AgroPlan, busca crear una aplicación web de gestión, en la que contendrá suficiente información como para poder brindar al agricultor, una experiencia resolutive y de mínimo esfuerzo en la planificación de las labores que tenga que llevar a cabo el agricultor. Aunque este sea el objetivo inicial de la aplicación, también es que en un futuro, siga creciendo “conceptualmente la herramienta”, contando con servicios web y con una base de datos mucho más completa que la actual, y para así resolver más incidencias de las que se tratarán en esta primera versión.

OBJ-01	Desarrollar una herramienta que permita al agricultor gestionar los cultivos, proporcionando información y consejos sobre la siembra, cosecha, tratamiento de enfermedades, plagas, calidad del suelo, etc.
OBJ-02	Crear una base de datos lo suficientemente completa, como para poder dar salida a todas las incidencias del agricultor en cualquier punto del cultivo. (Con capacidad para su crecimiento)
OBJ-03	Hacer que el agricultor optimice sus recursos y que sobre todo, sean lo más efectivos posibles gracias a las recomendaciones dadas por la aplicación.
OBJ-04	El funcionamiento de la aplicación está pensado para todas las regiones de España, teniendo en cuenta la toma de decisiones diferenciada para cada región.
OBJ-05	Permitir el registro y análisis de datos anteriores que influyan en las decisiones futuras basadas en esas experiencias previas.

Tabla objetivos de Agroplan.

1.5 Entorno de la aplicación

A continuación se describe brevemente el entorno donde se desarrollará AgroPlan así como las tecnologías seleccionadas para garantizar alto rendimiento, escalabilidad y facilidad de mantenimiento.

En el lado servidor, será el encargado de interactuar con el sistema de bases de datos, **PostgreSQL**, que gestionará una base de datos relacional. En este lado se utilizará el *framework* **Spring Boot**, que creará unas clases entidad y unas clases repositorio, donde se describirán las operaciones (*queries*) que interactúan con la base de datos. Además Spring Boot, cuenta con las llamadas clases controlador, que utilizaremos de *endpoints*, es decir, determinan los puntos de entrada a la API desarrollada con las operaciones accesibles desde el lado cliente.

El lado cliente recibirá un **JSON**, que procesa e interpreta para componer las vistas para la interfaz con el usuario. A su vez, el lado cliente, utilizará Axios y Fetch para comunicarse con el lado servidor, haciendo llamadas de tipo HTTP, (GET, POST o PUT.)

Con esto se quiere dejar claro, que el lado servidor no generará ningún HTML como en frameworks como Java Server Faces, únicamente una respuesta, que el lado cliente procesa posteriormente y generará ahora sí, una vista. Es por esta razón por la que en el patrón MVC la faceta de controlador está repartida entre *backend* y *frontend*.

Para ver más detalle sobre el entorno y arquitectura del proyecto, se recomienda consultar el apartado de *Diseño del sistema* (capítulo 5).

1.6 Tecnologías utilizadas

Se enumeran aquí las principales tecnologías utilizadas en el desarrollo de AgroPlan.

Java 8

Lenguaje de programación orientado a objetos y multiplataforma. La programación orientada a objetos (POO) es un paradigma de programación el cual organiza las entidades en clases, que instanciados son objetos. Cada objeto representa una entidad del problema a resolver.



Imagen: Logo Java.

PostgreSQL

Sistema de bases de datos relacional y de código abierto utilizado en AgroPlan. Garantiza la integridad de datos, permite transacciones sin bloqueos, etc.



Imagen: Logo PostgreSQL.

ReactJS

Biblioteca de JavaScript utilizada en AgroPlan, como Controlador y generador de la Vista del proyecto. Comúnmente utilizada para construir interfaces de usuario (IU), y especialmente utilizada en aplicaciones SPA (Single Page Applications).

Utiliza componentes que pueden ser reutilizables y que representan partes de la interfaz, como por ejemplo un formulario, un navbar, etc.



Imagen: Logo ReactJS.

Github

Sistema de control de versiones, que ha permitido en AgroPlan, almacenar, gestionar y asegurar los documentos que contienen el proyecto, guardando el historial de cambios en el código, así como los documentos asociados a ello. Permite además colaborar entre más desarrolladores, pudiendo trabajar simultáneamente en el mismo proyecto.



Imagen: Logo Github.

Trello

Herramienta utilizada para la gestión de proyectos que utiliza un tablero muy visual de lista de tareas para organizar el trabajo. Permite organizar las tareas por características, por ejemplo, tareas por hacer, en proceso y realizadas.



Imagen: Logo Trello.

Draw.io

Herramienta gratuita utilizada para crear diagramas UML. Utilizado para realizar los diagramas a lo largo de toda la etapa del diseño de AgroPlan se definen.



Imagen: Logo Draw.io

Visual Studio Code

Entorno de desarrollo integrado (IDE, *Integrated Development Enviroment*), que facilita la creación de aplicaciones de todo tipo, que soporta múltiples lenguajes de programación. Incluye herramientas de depuración, de edición de código, pruebas, control de versiones, etc.



Imagen: Logo Visual Studio Code

Node js

Entorno de ejecución para JavaScript, permite ejecutar JavaScript fuera del navegador, principalmente en el lado servidor. Permite construir aplicaciones web rápidas y escalables.



Imagen: Logo Node js

Npm

Gestor de paquetes predeterminado de Node.js. Gestiona dependencias y bibliotecas de

proyectos de JavaScript, tanto en el *frontend* como en el *backend*.



Imagen: Logo npm

Capítulo 2: Estado del arte

Como en todas las actividades siempre existen otras herramientas que ya existen que se convierten en competencia a la herramienta que desarrollamos. Se presentan ahora las diferentes plataformas y aplicaciones que ya existen en el mercado similares a AgroPlan. Aplicaciones similares que ofrecen distintas funcionalidades que algunas comparte AgroPlan. La idea de AgroPlan es ofrecer funcionalidades nuevas, que no se contemplen en el mercado y hacer de AgroPlan una herramienta que cada vez más agricultores necesiten para sus campos.

2.1 Herramientas existentes

Agroptima

Aplicación de gestión agrícola la cual permite al agricultor controlar y registrar las actividades de sus explotaciones de manera ágil e intuitiva. Facilita planificación y distribución de labores agrícolas. [\[Agroptima\]](#)

- Proporciona al agricultor información sobre plagas y enfermedades comunes y ofrece consejos para su identificación y tratamiento.
- Ofrece también orientación sobre la elección de diferentes fertilizantes o abonos en función de la necesidad del suelo.



Imagen: Logo “Agroptima”

AGRIVI

Aplicación de gestión agrícola utilizado globalmente, en la que se ofrece información en tiempo real de la cadena de valor agroalimentaria. Ofrece herramientas para planificación, seguimiento y análisis de todas las actividades agrícolas. [\[Agrivi\]](#)

- Planificación y seguimiento de cultivos.
- Informa al agricultor mediante algoritmos avanzados sobre riesgos de plagas o enfermedades.



Imagen: Logo "Agrivi"

Kaampo

Software agrícola que integra la gestión de parcelas, tratamientos fitosanitarios y fertilizantes, labores de campo, operarios, producción agrícola y facturación. Además contiene herramientas para hacer al agricultor cumplir con las normativas vigentes del sector, como por ejemplo el cuaderno de campo o producción ecológica. [\[Kaampo\]](#)

- Utiliza drones equipados con sensores especiales para monitorear los cultivos identificando plagas, enfermedades y demás necesidades de la planta.



Imagen: Logo "Kaampo"

Efemis

Efemis es una plataforma digital avanzada que gestiona la actividad de campo de forma integral e inteligente. Genera el cuaderno de campo automáticamente. [\[Efemis\]](#)

- Planificación y control de actividades
- Permite registrar presencia de plagas y enfermedades y asociarlas a tierras.



Imagen: Logo "Efemis"

Farmable

Plataforma de gestión agrícola que permite digitalizar fincas, facilitando la planificación y seguimiento de actividades de campo. Ofrece funcionalidades de registrar datos, monitorear cultivos y analizar información relevante para la toma de decisiones. [\[Farmable\]](#)

- Facilita el seguimiento de plagas y enfermedades a lo largo del tiempo evaluando la efectividad de los diferentes tratamientos aplicados.
- Gestión de inventario de productos.



Imagen: Logo "Farmable"

2.2 Comparativa

Se muestra en la tabla, la comparativa de las diferentes funcionalidades más reseñables de entre las aplicaciones. Señalando con un ✓, aquellas aplicaciones que cuenten con esa funcionalidad.

	Agroptima	AGRIVI	Kaampo	Efemis	Farmable	Agroplan
Gestión de cultivos	✓	✓	✓	✓	✓	✓
Gestión de maquinaria						✗
Predicción meteorológica	✓	✓		✓		✗
Control de riego		✓	✓			✗
Gestión de tareas de campo	✓	✓	✓	✓	✓	✓
Programación de siembras	✓	✓	✓	✓	✓	✓
Detección plagas/enfermedades	✓	✓	✓		✓	✓
Recomendaciones sobre calidad suelo (minerales, abonos, etc...)	✓			✓		✓
Gestión de siembras en función de la región	✓	✓		✓		✓

Tabla: Comparativa Estado del Arte.

Capítulo 3: Planificación del proyecto

3.1 Metodología de trabajo

Aquí se describe la metodología de trabajo que se llevará a cabo para desarrollo del proyecto. En este caso se opta por una metodología Ágil. El desarrollo del proyecto se dividirá en 3 sprints, cada uno de los cuales tendrá una duración de 4 semanas.

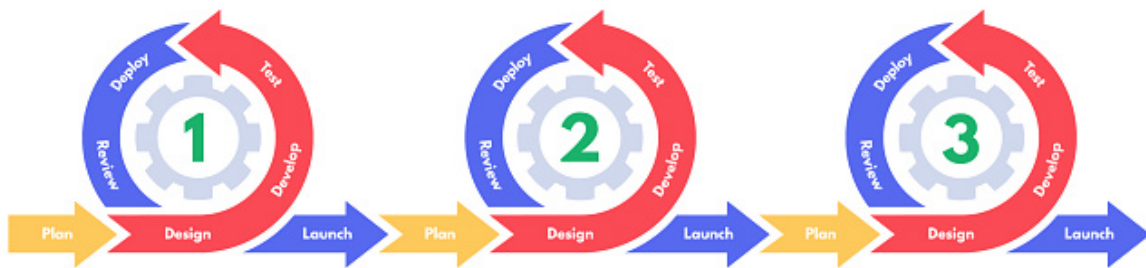


Imagen: Estructura Ágil.

Para llevar a cabo la gestión de las tareas e ítems a realizar en cada sprint del desarrollo del proyecto, se utilizará **Trello**, donde se definirán las tareas por hacer, en proceso y realizadas.

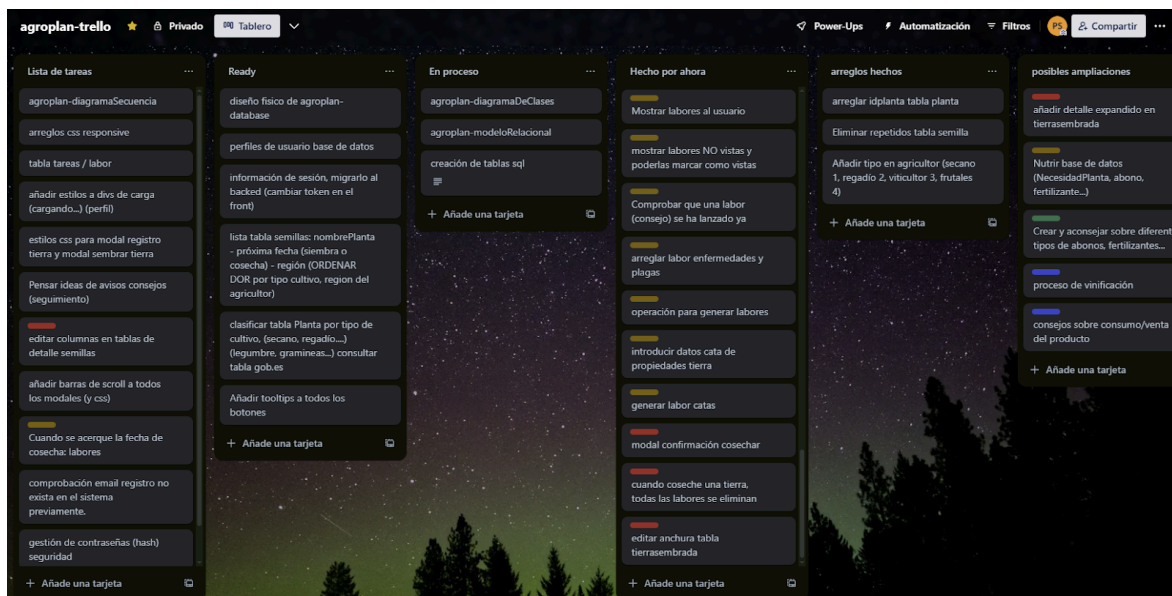


Imagen: Tablero "Agroplan-trello", Trello.

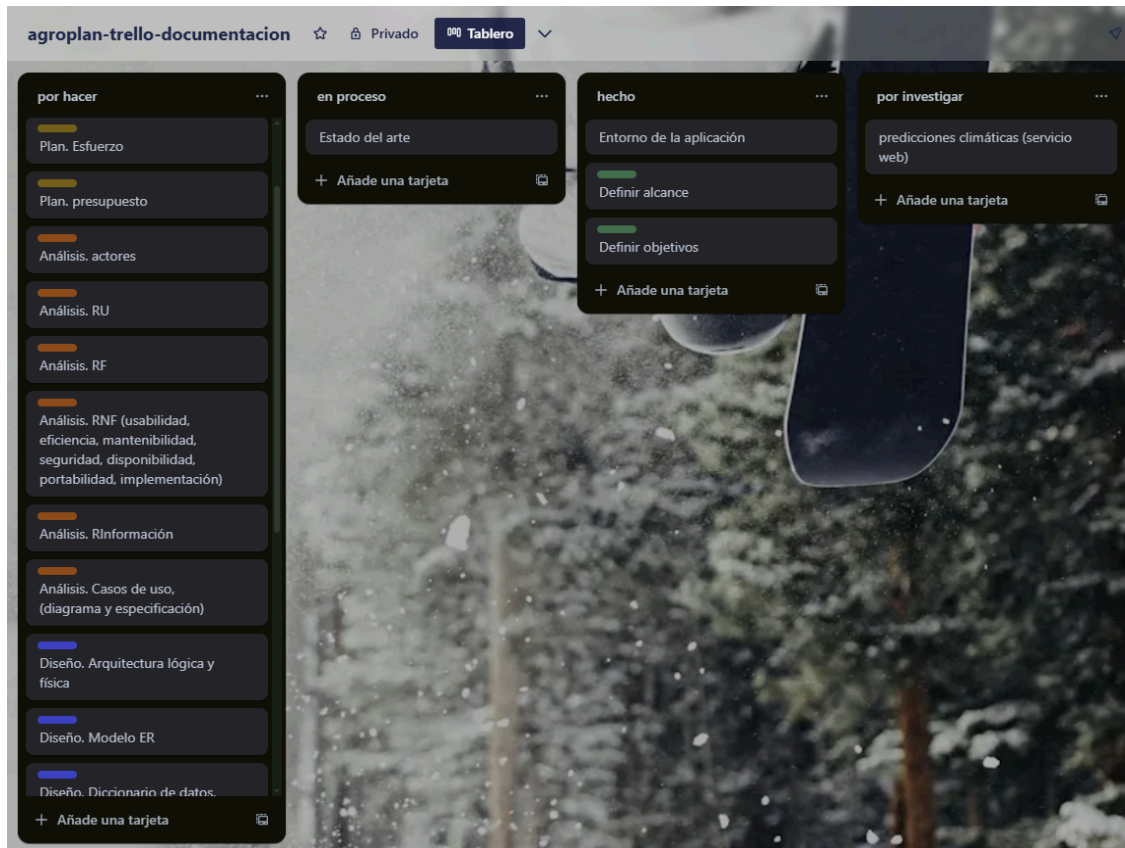


Imagen: Tablero “Agroplan-trello-documentación”, Trello.

Ventajas e inconvenientes

La metodología utilizada ha sido Agile, donde se ha dividido la carga de trabajo en 3 *sprints*, con entregas incrementales. Esta manera de organizar el trabajo permite tener una visión clara del progreso y, sobre todo útil, para respetar el alcance y no salirse de él. Se elige también esta metodología porque es útil a la hora de introducir cambios en el proceso de desarrollo del proyecto.

Al igual que en cualquier otro proyecto organizado con otra metodología, uno de los inconvenientes ha sido la falta de personal, ya que, contando con más desarrolladores, se podría haber llegado a un amplio tablero de ideas, la cual Agile suele fomentar.

Otro inconveniente, relacionado con el seguimiento, ha surgido como consecuencia del proyecto, ha sido el de no tener reuniones diarias o semanales, entre los desarrolladores. Sin embargo al final de cada *sprint* y algunas veces incluso más a menudo, se han realizado reuniones de control con el tutor de este proyecto.

3.2 Planificación temporal

Se describe aquí la planificación temporal para el desarrollo del sistema AgroPlan. Se ha seguido un método de planificación Agile, y se ha dividido la carga de trabajo en 3 bloques. Se espera que dadas las condiciones del alumno, nos situamos a finales del primer cuatrimestre del 5º año, terminando las últimas asignaturas, y se espera un parón en Navidades por tema de estudio para los debidos exámenes. Se prevé que después de enero, el desarrollo del proyecto continúe adecuadamente.

Como bien se explica anteriormente, se divide la carga de trabajo en 3 *sprints*. Para el primer *sprint*, se calcula que durará 4 semanas, compaginándolo con clases de universidad.

En este bloque se abordará la construcción del proyecto en su base, análisis del objetivo y alcance y definición de los primeros requisitos

Sprint 1:

Desarrollar la estructura básica del sistema y la conexión entre *frontend*, *backend* y base de datos.

- Definición de los primeros requisitos funcionales y no funcionales. Requisitos de funcionalidades en la aplicación. (semana 1)
- Definición también los requisitos de usuario. (semana 1)
- Configuración del entorno (java, reactjs y postgresql). (semana 2)
- Diseño de la estructura de la base de datos, Modelo entidad relación. (semanas 2-3)
- Creación de clases entidad y creación de API Rest en backend. (semana 3)
- Diseño de interfaz de usuario, creación de lógica para login y registro de agricultores. (semana 4)

Para el segundo *sprint*, se prevé que la mayor parte del tiempo se va a dedicar a investigar y a la búsqueda de información para la base de datos de AgroPlan. Y por consiguiente también se realizarán las funcionalidades para gestionar y mostrar la información nueva de la base de datos (semillas).

Sprint 2:

Desarrollo de las primeras funcionalidades del sistema:

- Definición de los siguientes requisitos a realizar en este sprint. (nuevas funcionalidades). (semana 5)
- Proceso de “nutrir” la base de datos con la información de las semillas organizadas por región. (semanas 5-6)
- Funcionalidades de creación (crear, modificar y eliminar) de tierras por el agricultor. (semana 6)

- Funcionalidad de mostrar toda la información acerca de las semillas de la base de datos.(semanas 6-7)
- Funcionalidad de sembrar y cosechar tierras. (semanas 7-8)

Para el tercer *sprint*, será parecido al 2º, pero ampliado en investigación y en implementación de funcionalidades. Se llevará a cabo la investigación para las necesidades de cada planta, y por consiguiente labores para aplicar a los cultivos.

Sprint 3:

Implementación de funcionalidades de planificación y consejos de Agroplan

- Investigación información sobre necesidades planta (semana 5)
- Desarrollo de funcionalidades para mostrar información (semana 6)
- Generación de operaciones y labores para el agricultor (semana 7)
- Pruebas y corrección de errores general (semana 8 en adelante)

3.3 Presupuesto económico

A continuación, se muestra el presupuesto económico para el proyecto de AgroPlan. Contamos con el trabajo de un solo trabajador el cual es estudiante mientras la realización de éste, y teniendo en cuenta la organización temporal descrita previamente para el desarrollo de este proyecto. Se estima que la jornada de trabajo es de 5 horas diarias (media jornada).

Duración total (primera estimación): 8 semanas

Equipo de desarrollo: 1 desarrollador (interno)

Actividad	Horas estimadas	Coste por hora €	Total
Análisis y requisitos	10 h	20€	200€
Diseño arquitectura inicial	6 h	20€	120€
Login, registro y configuración <i>backend</i> y <i>frontend</i>	18 h	20€	360€
Total sprint 1			680€

Actividad	Horas estimadas	Coste por hora €	Total
Investigación BD	15 h	20€	300€

Diseño y creación de tablas en BD	10 h	20€	200€
Desarrollo de funcionalidades “Semillas”	16 h	20€	320€
Correcciones + <i>frontend</i>	8 h	20 €	160€
Total sprint 2	980€		

Actividad	Horas estimadas	Coste por hora €	Total
Investigación de necesidades por planta	18 h	20€	360€
Desarrollo <i>frontend/backend</i> visualización	14 h	20€	280€
Generación de operaciones para el agricultor	16 h	20€	320€
Pruebas y correcciones generales	12 h	20€	240€
Total sprint 3	1200€		

Estimación de costes adicionales:

Concepto	Coste total €	Uso %	Total €
Computadora portátil	1100€	5%	55€
Monitor	200€	2%	4€
Ratón	30€	1%	0,3€
Total uso	59,30€		

Concepto	Total
Hosting y dominio	42€
Licencias y API externas (ver ampliaciones)	40€

Uso de material (Computador, monitor, etc.)	59,30€
Subtotal adicional	141,30€

Resumen final del presupuesto	Total
Desarrollo total	2860€
Costes adicionales	141,30€
Total	3001,30€

3.4 Balance final

El proceso de planificación y desarrollo de AgroPlan ha sido una experiencia completa y enriquecedora, desde el principio en la búsqueda de la idea, mientras se recopilan los requisitos hasta que se iban implementando las primeras funcionalidades. A lo largo del desarrollo se han ido completando los objetivos, más o menos, en tiempo.

Se quiere aclarar que la distribución del tiempo no ha sido específica para el desarrollo de este proyecto, sino que el alumno se encontraba en el inicio trabajando en empresa (Delonia SW), donde hizo las prácticas curriculares y estudiando. Y después en las siguientes fases del proyecto siguió trabajando en dicha empresa a la vez que en el proyecto.

Considerando el planteamiento inicial de AgroPlan y donde más tiempo se ha dedicado, ha sido en el desarrollo de la base de datos. AgroPlan tenía como objetivo principal, contar con una fuente de información interna, que le pudiese valer al agricultor para todo tipo de contingencias, En un futuro, con la expansión de la aplicación, se dedicará aún más cantidad de tiempo a completar la base de datos.

Se espera de AgroPlan ser una herramienta útil y, por supuesto escalable, de modo que de forma continua se desarrollen nuevas funcionalidades para proveer al usuario nuevas soluciones a sus labores en el campo.

Capítulo 4: Análisis del sistema

Se describe el análisis del sistema, actores, requisitos, diagramas, etc, necesarios para la identificación de todas las características necesarias del proyecto a realizar.

4.1 Alcance del sistema

Se expone el árbol de características, donde se observa cuales son las tres principales funcionalidades del sistema y sus subcaracterísticas. Gestión de tierras, gestión de labores y semillas.

El árbol de características es útil para tener una visión de las partes de las que dispone un producto, en este caso Agroplan.

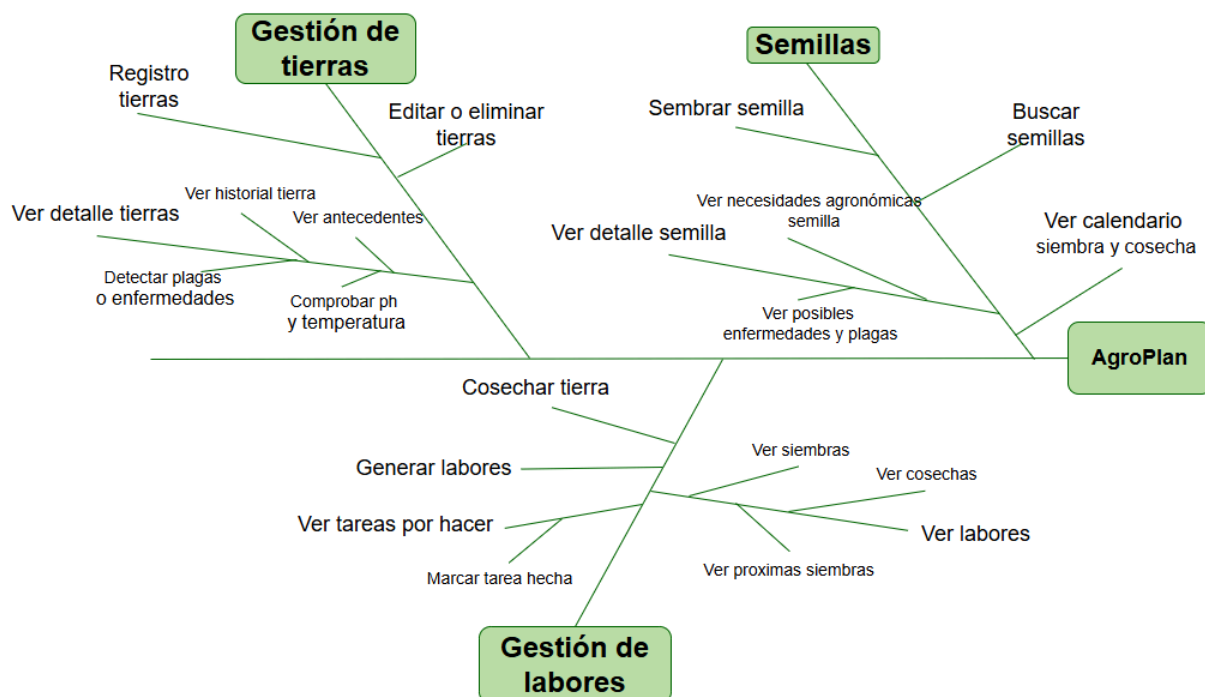


Diagrama de características.

4.2 Actores del sistema

Habrà un administrador que será el encargado de actualizar, añadir o eliminar información en base de datos. El objetivo de AgroPlan es crecer, y cada vez añadir nuevas funcionalidades, por lo tanto para generar más labores al agricultor, se necesitará en un futuro más

información. En un futuro se contempla realizar un proyecto interno, para dotar al administrador, de una interfaz sencilla para gestionar toda la información de AgroPlan así como la gestión de usuarios. Por ahora al no desarrollarse esta funcionalidad, con el contexto de este proyecto, no se incluye como actor del sistema a desarrollar, limitándose a los usuarios finales del sistema, los agricultores.

Usuarios agricultores:

AC-01	Usuario no logueado	Usuario que se encuentra en la página de inicio de sesión y puede o bien registrarse o bien iniciar sesión.
AC-02	Usuario logueado	Usuario que accede a todas las funcionalidades del sistema una vez ha iniciado sesión con éxito.

Tabla: Actores del sistema.

4.3 Requisitos de usuario

Funcionalidades y actividades que se les permitirá a los usuarios Agricultores en la aplicación AgroPlan.

Usuario no logueado

id	Descripción
CU-01	El usuario podrá introducir sus datos de inicio de sesión o en caso de no tener cuenta, introducir sus datos en el formulario de registro.

Tabla: Requisitos de usuario no logueado.

Usuario logueado

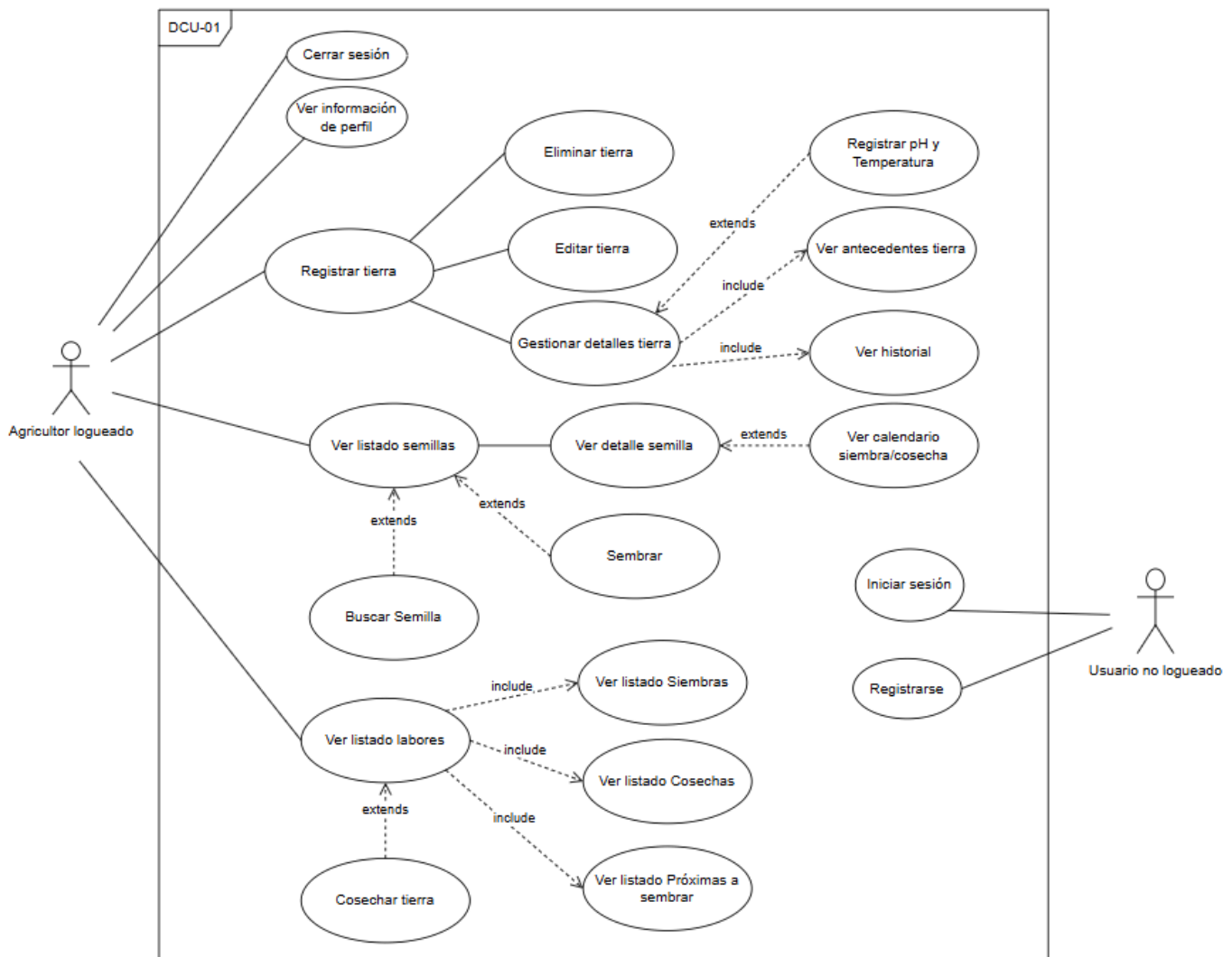
id	Descripción
Información de perfil de usuario	
CU-02	El usuario podrá acceder a ver su información personal.
CU-03	El usuario podrá cerrar sesión.
Gestión de tierras	
CU-04	El usuario podrá registrar nuevas propiedades en el formulario de registro de

	tierras.
CU-05	El usuario podrá ver, modificar y eliminar todas las propiedades que tiene registradas.
CU-06	El agricultor podrá ver labores pendientes y marcarlas como realizadas.
CU-07	El usuario podrá acceder a ver el historial de cada tierra, así como los antecedentes de enfermedades o plagas.
CU-08	El usuario podrá introducir datos de ph y temperatura para comprobar si están dentro de los valores recomendados.
CU-09	El usuario podrá visualizar un listado de plagas y enfermedades posibles a la semilla seleccionada.
CU-10	El usuario podrá detectar una plaga o enfermedad de esos listados identificándose por los síntomas.
Gestión de semillas	
CU-11	El usuario podrá acceder a toda la información de todas las semillas registradas en la aplicación.
CU-12	El usuario podrá sembrar una tierra seleccionando la semilla en la lista de semillas.
CU-13	El usuario podrá buscar semillas por el nombre en un buscador.
CU-14	El usuario podrá ver el calendario de siembra y cosecha de cada semilla.
CU-15	El usuario podrá acceder a información sobre posibles plagas o enfermedades de cada semillas.
Gestión de labores	
CU-16	El usuario podrá cosechar una tierra cuando ésta esté sembrada.
CU-17	El usuario podrá acceder al listado de todas las tierras sembradas, no sembradas y próximas a sembrar.
CU-18	El usuario podrá ver tareas o consejos que genere AgroPlan para él.

Tabla: Requisitos de usuario logueado.

4.4 Diagrama de casos de uso

A continuación se muestran los diagramas de casos de uso, que describen las interacciones entre un actor externo (en este caso el usuario no logueado o usuario agricultor de Agroplan) y el sistema.



CU-01: Diagrama de casos de uso de AgroPlan.

Especificación de casos de uso

id: CU-12	Sembrar semilla
Descripción	El usuario podrá registrar una siembra seleccionando una semilla y una tierra.
Actores	Agricultor logueado
Precondiciones	- La tierra no puede estar sembrada
Flujo normal	- Agricultor solicita ver el listado de la semilla - El sistema muestra el detalle de la semilla - El sistema solicita la selección de una tierra disponible - El agricultor selecciona una tierra - El sistema solicita la fecha de siembra - El agricultor introduce la fecha - El sistema permite confirmar la siembra de la tierra (si está disponible) - El agricultor confirma la siembra
Postcondiciones	Aparecerá una nueva línea en el listado de labores con el detalle de la siembra.
Importancia	ALTA

Tabla: Especificación Caso de uso "Sembrar semilla".

id: CU-08	Registrar pH y Temperatura
Descripción	El usuario podrá introducir los datos de pH y de Temperatura de la tierra para conocer si es adecuado con el cultivo sembrado.
Actores	Agricultor logueado
Precondiciones	- El agricultor debe de tener tierras registradas
Flujo normal	- Agricultor accede a detalle de la tierra (detalles) - El agricultor introduce los valores de pH o de Temperatura o ambos, para realizar la comprobación - Envía los datos y se le responde con un mensaje explicativo.
Postcondiciones	Aparecerá un resultado de respuesta de la aplicación donde se verá si los datos introducidos están correctos o no.
Importancia	MEDIA

Tabla: Especificación Caso de Uso “Comprobar pH y Temperatura”.

id: CU-10	Registrar Plaga o Enfermedad
Descripción	El usuario podrá registrar una plaga o una enfermedad, reconociendo los síntomas al detectarla.
Actores	Agricultor logueado
Precondiciones	- El agricultor debe de tener tierras registradas
Flujo normal	- Agricultor accede al detalle de la tierra y selecciona el control de plagas y enfermedades (detalles) - El sistema muestra las diferentes enfermedades o plagas posibles para ese cultivo y mostrará los síntomas para cada una. - El agricultor identifica el problema y lo selecciona - El sistema añade una nueva tarea al buzón de labores y quedará la plaga o enfermedad registrada en los antecedentes. <u>Excepciones</u> Si la enfermedad o plaga no es reconocida termina el caso de uso.
Postcondiciones	Aparecerá una nueva línea en el listado de antecedentes en el detalle de la tierra.
Importancia	ALTA

Tabla: Especificación Caso de Uso “Detectar plaga o enfermedad”.

4.5 Requisitos funcionales

id	Descripción
Inicio de sesión	
RF-01	El sistema comprobará los datos de inicio de sesión de un usuario que quiera loguearse.
RF-02	El sistema registrará los datos a un nuevo usuario.
RF-03	El sistema comprobará que los datos de registro de un nuevo usuario son correctos.
RF-04	El sistema permitirá recuperar la información registrada en su cuenta personal.
Listado semillas	
RF-05	El sistema mostrará un mensaje emergente de aviso, si la semilla seleccionada por el usuario para sembrar no está en periodo óptimo de siembra.
RF-06	El sistema no permitirá sembrar una tierra si ya se encuentra sembrada, además mostrará un mensaje de error informando al agricultor.
RF-07	El sistema mostrará un detalle completo de la semilla (calidad de la tierra necesaria, minerales, abonos recomendados, posibles enfermedades o plagas, y un calendario especificando el periodo de siembra y cosecha recomendado)
Listado de tierras del agricultor	
RF-08	El sistema mostrará un listado de las tierras registradas del agricultor, mostrando en cada fila información importante sobre la tierra, como nombre, hectáreas... y calculará el estado de la tierra: (sembrado, barbecho, cosechado o próximo a sembrar)
RF-09	El sistema no permitirá eliminar una tierra si está sembrada.
RF-10	El sistema pedirá una segunda confirmación antes de eliminar una tierra.
RF-11	El sistema permitirá editar la tierra al usuario.
RF-12	En cada fila, habrá un icono que aparecerá cuando el agricultor tenga alguna tarea pendiente. Una vez el agricultor la marque como vista/realizada, desaparecerá.
Componente labores	
RF-13	El sistema permitirá navegar por una barra de navegación al usuario

	mostrándole todo el listado de siembras, cosechas y próximas a sembrar.
RF-14	El sistema pedirá una segunda confirmación antes de cosechar una tierra.
Detalle tierra	
RF-15	El sistema mostrará un detalle completo de la siembra de la tierra: - Historial de todas las siembras de la tierra - Antecedentes de plagas o enfermedades)
RF-16	El sistema comprobará los datos de pH y Temperatura introducidos por el usuario en un pequeño formulario, y mostrará un mensaje dependiendo si está correcto o si hay que hacer alguna corrección.
Más	
RF-17	El sistema mostrará en la parte superior derecha en todo momento la región del agricultor y la fecha del día.
RF-18	Todos los listados están paginados en función del alto de la pantalla.

Tabla: Requisitos funcionales.

4.6 Requisitos no funcionales

Usabilidad

id	Descripción
RNF-01	La interfaz debe ser accesible e intuitiva para el agricultor. No le debe de costar más de 10 minutos saber manejar la aplicación con fluidez.
RNF-02	Un usuario no experimentado debería de poder realizar cualquier operación en menos de 2 minutos.
RNF-03	El sistema deberá de proporcionar mensajes de errores claros cuando ocurra cualquier evento erróneo.

Tabla: Requisitos no funcionales de usabilidad.

Eficiencia

id	Descripción
RNF-04	Cualquier acción del usuario deberá de tener una respuesta visual en un tiempo inferior a 500 ms.
RNF-05	El sistema deberá soportar al menos 10000 usuarios concurrentes sin

	degradación significativa del rendimiento.
RNF-06	Las consultas a base de datos no deben de tardar más de 100 ms en ejecutarse.
RNF-07	El tráfico de datos entre el cliente y el servidor no superará los 200 kb por cada solicitud para evitar latencias.
RNF-08	La aplicación deberá iniciarse completamente en menos de 5 sg.

Tabla: Requisitos no funcionales de eficiencia.

Mantenibilidad

id	Descripción
RNF-09	El código deberá estar correctamente comentado y con documentación complementaria para hacer legible y entendible la totalidad de las operaciones.
RNF-10	Un error detectado en producción deberá ser identificado y corregido en menos de 4 horas.
RNF-11	Todo el código fuente estará gestionado por un sistema de control de versiones (Git), con los commits perfectamente explicados.
RNF-12	Se realizan revisiones periódicas cada 5 meses, para eliminar código obsoleto o mejorar su eficiencia.
RNF-13	El desarrollador nuevo no tardará más de 1 semanas en familiarizarse con el proyecto.

Tabla: Requisitos no funcionales de mantenibilidad.

Seguridad

id	Descripción
RNF-14	El sistema será resistente a ataques de tipo de inyección SQL, javascript...
RNF-15	Se limitará el tamaño y la frecuencia de las solicitudes para prevenir ataques de Denegación de servicio.
RNF-16	Cada 24 horas, se hará una copia de seguridad y será almacenada en una ubicación segura.

Tabla: Requisitos no funcionales de seguridad.

Disponibilidad

id	Descripción
RNF-17	El sistema estará disponible al menos el 99% del tiempo al mes.
RNF-18	Los usuarios deberán ser notificados en caso de alguna jornada de mantenimiento.
RNF-19	En caso de fallo en el servidor, el sistema debe reanudar su funcionamiento en menos de 5 min.
RNF-20	El sistema será capaz de estar disponible para mínimo 10000 usuarios simultáneamente.

Tabla: Requisitos no funcionales de disponibilidad.

Portabilidad

id	Descripción
RNF-21	La aplicación web será compatible para todos los sistemas (windows, linux, macOS...
RNF-22	La aplicación será completamente funcional en Google Chrome, Mozilla Firefox, Microsoft Edge y Safari.
RNF-23	La interfaz de usuario será responsive, y se adaptará perfectamente a todos los anchos de pantalla.

Tabla: Requisitos no funcionales de portabilidad.

Escalabilidad

id	Descripción
RNF-24	El sistema debe ser capaz de soportar un incremento de tamaño de la base de datos de un 300%
RNF-25	Las bases de datos deben de soportar hasta 100000 transacciones por minuto sin afectar al rendimiento.

4.7 Requisitos de información

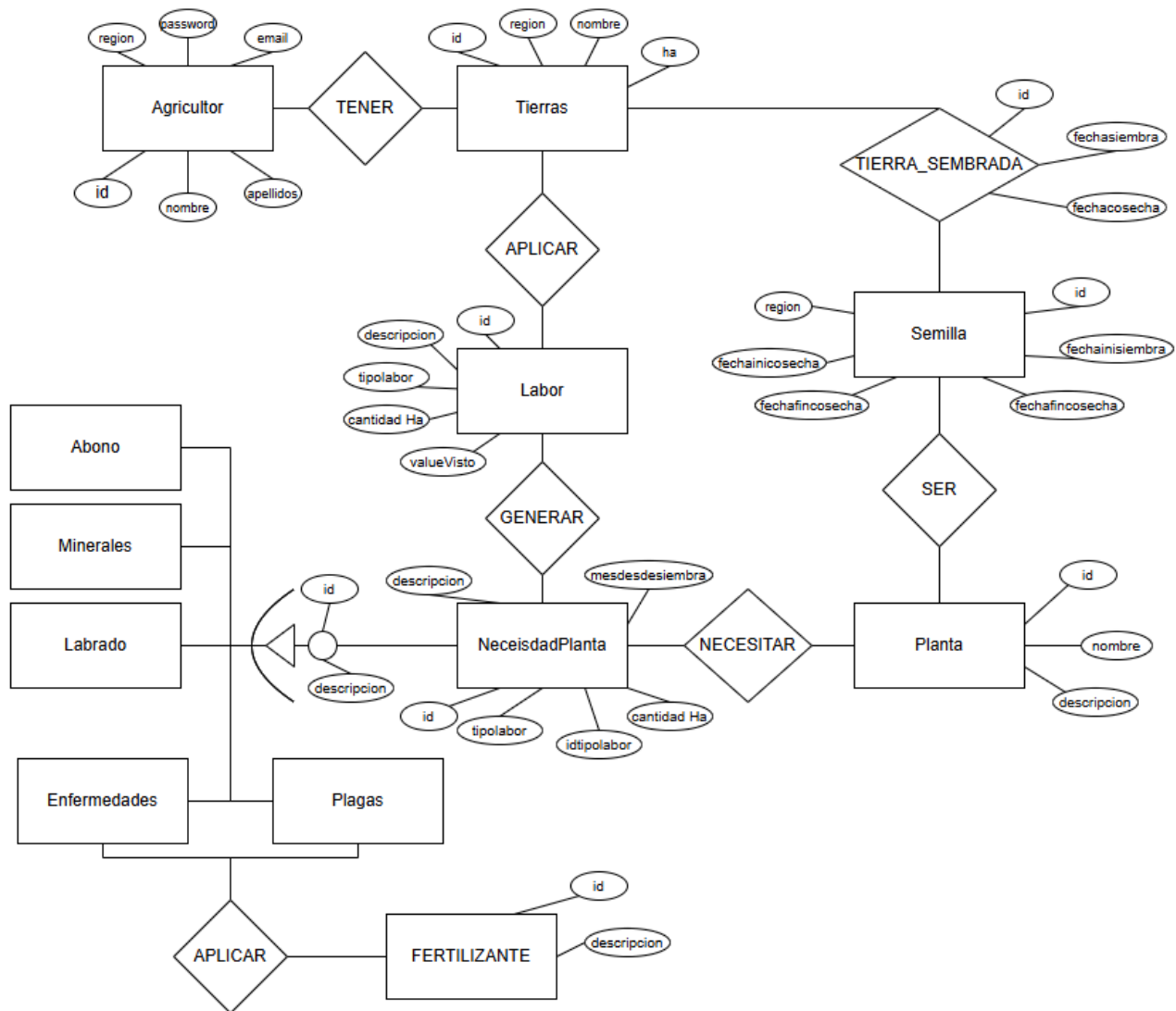
Se muestran ahora los requisitos de información, describen los datos que el sistema necesita almacenar, procesar, consultar o mostrar para cumplir con las funcionalidades y necesidades del agricultor.

id	Descripción
RI-01	El sistema deberá almacenar información de los usuarios agricultores (nombre y apellidos, email y región)
RI-02	El sistema deberá almacenar información de las semillas y necesidades de cada semilla. (periodos de siembra y cosecha por región, necesidades del suelo ...)
RI-03	El sistema deberá almacenar información sobre todas las tierras que registre el usuario. (nombre, región, hectáreas)
RI-04	El sistema deberá almacenar un historial de todas las acciones que se le apliquen a una tierra. (siembras y cosechas, aplicación de fitosanitarios, plagas o enfermedades, fechas de siembra y cosecha...)
RI-05	El sistema deberá almacenar información acerca de las labores que se le recomiende hacer al agricultor.
RI-06	El sistema consultará información en la base de datos de minerales, abonos, fertilizantes, labrados, plagas, enfermedades...

Tabla: Requisitos de información.

4.8 Modelo E-R

El modelo entidad relación, utilizada para diseñar la base de datos, nos ayuda a representar gráficamente las entidades involucradas en el proyecto y las relaciones entre ellas. Consigue mostrar a la perfección cómo se organiza la información en el sistema.



Modelo entidad relación.

4.9 Diccionario de datos

Especificación de los tipos de datos que se manejan en cada entidad de Agroplan. Restricciones, valores por defecto... Esencial para documentar el sistema y comprender la estructura de datos de Agroplan.

Entidad: Agricultor			
Atributo	Tipo	Único	Null
idagricultor	integer	si	no
namenombre	string	no	no
nameapellidos	string	no	si
valueemail	string	si	no
valueregion	string	no	no
password	string	no	no

Diccionario de datos: Entidad agricultor

Entidad: Tierra			
Atributo	Tipo	Único	Null
idtierra	integer	si	no
idagricultor	integer	no	no
valueregion	string	no	no
nombretierra	string	no	no
hectareas	double	no	no

Diccionario de datos: Entidad tierra

Entidad: Planta			
Atributo	Tipo	Único	Null
idplanta	integer	si	no
nombrecultivo	string	no	no

Entidad: Semilla			
Atributo	Tipo	Único	Null
idsemilla	integer	si	no
nombrecultivo	string	no	no
fechainisiembra	string	no	si
fechafinsiembra	string	no	si
fechainicosecha	string	no	si
fechafincosecha	string	no	si
región	string	no	no

*Que los periodos de siembra y cosecha puedan ser null, significa que puede haber regiones donde no se cultive un determinado cultivo.

Entidad: Necesidad planta			
Atributo	Tipo	Único	Null
idnecesidadplanta	integer	si	no
idplanta	integer	no	no
tipolabor	integer	no	no
cantidadHa	double	no	si
liminf	double	no	si
limsup	double	no	si
mesdesdesiembra	string	no	si
descripción	string	no	si
idtipolabor	integer	no	si

Entidad: Tierra sembrada			
Atributo	Tipo	Único	Null
idtierrasembrada	integer	si	no
idtierra	integer	no	no
idcultivo	integer	no	no
idagricultor	integer	no	no
fechasiembra	Date	no	no
fechacosecha	Date	no	no

Diccionario de datos: Entidad tierra sembrada

Entidad: Abono			
Atributo	Tipo	Único	Null
idabono	integer	si	no
descripcion	string	no	no

Diccionario de datos: Entidad abono

Entidad: Labrado			
Atributo	Tipo	Único	Null
idlabrado	integer	si	no
descripcion	string	no	no

Diccionario de datos: Entidad labrado

Entidad: Minerales			
Atributo	Tipo	Único	Null
idmineral	integer	si	no
descripcion	string	no	no

Entidad: Enfermedades			
Atributo	Tipo	Único	Null
idenfermedad	integer	si	no
descripcion	string	no	no
síntomas	string	no	no
prioridad	integer	no	no
idfertilizante	integer	no	si

Diccionario de datos: Entidad enfermedades

Entidad: Plagas			
Atributo	Tipo	Único	Null
idplaga	integer	si	no
descripcion	string	no	no
síntomas	string	no	no
prioridad	integer	no	no
idfertilizante	integer	no	si

Diccionario de datos: Entidad plagas

Entidad: Fertilizantes			
Atributo	Tipo	Único	Null
idfertilizante	integer	si	no
descripcion	string	no	no

Diccionario de datos: Entidad fertilizantes

Entidad: Labor			
Atributo	Tipo	Único	Null

idlabor	integer	si	no
descripcionlabor	string	no	no
idsemilla	integer	no	si
cantidadHa	double	no	si
idfertilizante	integer	no	si
tipolabor	integer	no	no
idtipolabor	integer	no	si
idtierra	integer	no	no
valuevisto	boolean	no	no

Diccionario de datos: Entidad labor

Capítulo 5: Diseño del sistema

En el diseño del sistema, se describen los aspectos más importante de la estructura de AgroPlan, arquitecturas utilizadas, arquitectura de datos.

5.1 Estructura de la información

Uno de los objetivos de AgroPlan es crear una base de datos completa y sobre todo escalable. AgroPlan necesita de suficiente información para poder estudiar cada cultivo, para poder brindar al agricultor los mejores consejos.

Y es necesario describir cómo esa información va a ser organizada para su comprensión.

En primer lugar deberá de existir una tabla para almacenar la información del **usuario**, así como el nombre, email, región y la contraseña.

idagricultor [PK] integer	namenombre character varying (150)	nameapellidos character varying (150)	valueemail character varying (150)	valueregion character varying (150)	password character varying (200)
------------------------------	---------------------------------------	--	---------------------------------------	--	-------------------------------------

El agricultor podrá registrar nuevas **tierras** y operar con ellas, por lo que será necesario crear una tabla para guardar registro de las tierras creadas. De cada tierra se guardará la siguiente información: el id del agricultor que la creó, el nombre, la región (importante) y las hectáreas.

idtierra [PK] integer	idagricultor integer	valueregion character varying (120)	nombretierra character varying (120)	hectareas real
--------------------------	-------------------------	--	---	-------------------

Una vez el agricultor registre una tierra, éste podrá sembrarla, y el sistema guardará información de cada siembra/cosecha que el agricultor realice de todas las tierras. Además guardará información de las plagas y enfermedades que haya sufrido esa tierra con la planta afectada.

idtierrasembrada [PK] integer	idtierra integer	idcultivo integer	idagricultor integer	fechasiembra date	fechacosecha date	datecreacion timestamp without time zone
----------------------------------	---------------------	----------------------	-------------------------	----------------------	----------------------	---

El sistema por defecto tendrá información ya incluida acerca de las semillas que puede el agricultor sembrar de las que se guardará la siguiente información. Nombre cultivo, y se guardará el periodo de siembra (fecha inicio siembra y fecha fin siembra) y el periodo de cosecha (fecha inicio cosecha y fecha fin cosecha) en función de la región. Se pretende así, que se tenga en cuenta la región donde vayamos a cultivar cierta semilla, ya que las condiciones principalmente climáticas no son iguales en diferentes puntos de España.

(Primera versión de agroplan, solamente para España península, incluidas Islas Canarias y Baleares).

idsemilla integer	nombrecultivo character varying (150)	fechainisiembra character varying (	fechafinsiembra character varying (	fechainicosecha character varying (	fechafincosecha character varying (	region character
----------------------	--	--	--	--	--	---------------------

Además de guardar registro de los periodos de siembra y de cosecha, Agroplan cuenta con información acerca de las necesidades de cada semilla, y se organizan por tipos de necesidades:

Tipo 1: Abonos recomendados.

Tipo 2: Minerales necesarios para cada planta.

Acerca de los abonos y los minerales AgroPlan podrá recomendar sobre la cantidad de cada producto por hectárea recomendado.

Tipo 3: Labrado del suelo.

Tipo 4: Recomendaciones como por ejemplo niveles de pH y Temperatura.

Tipo 5: Posibles enfermedades.

Tipo 6: posibles plagas.

Además en particular en las enfermedades y plagas, se guardará información acerca de los fertilizantes recomendables para paliarlas y los síntomas de cada una para que el agricultor pueda identificarlos sin duda.

De todas estas necesidades AgroPlan irá generando **labores**, que se le irán comunicando al agricultor en forma de tareas para el correcto cultivo.

5.2 Modelo lógico de datos

Describe cómo se organizan y relacionan las estructuras de datos del sistema. Independientemente del sistema de bases de datos que es donde se almacenan los datos de cada componente, aquí se detalla las uniones entre cada una de las entidades.

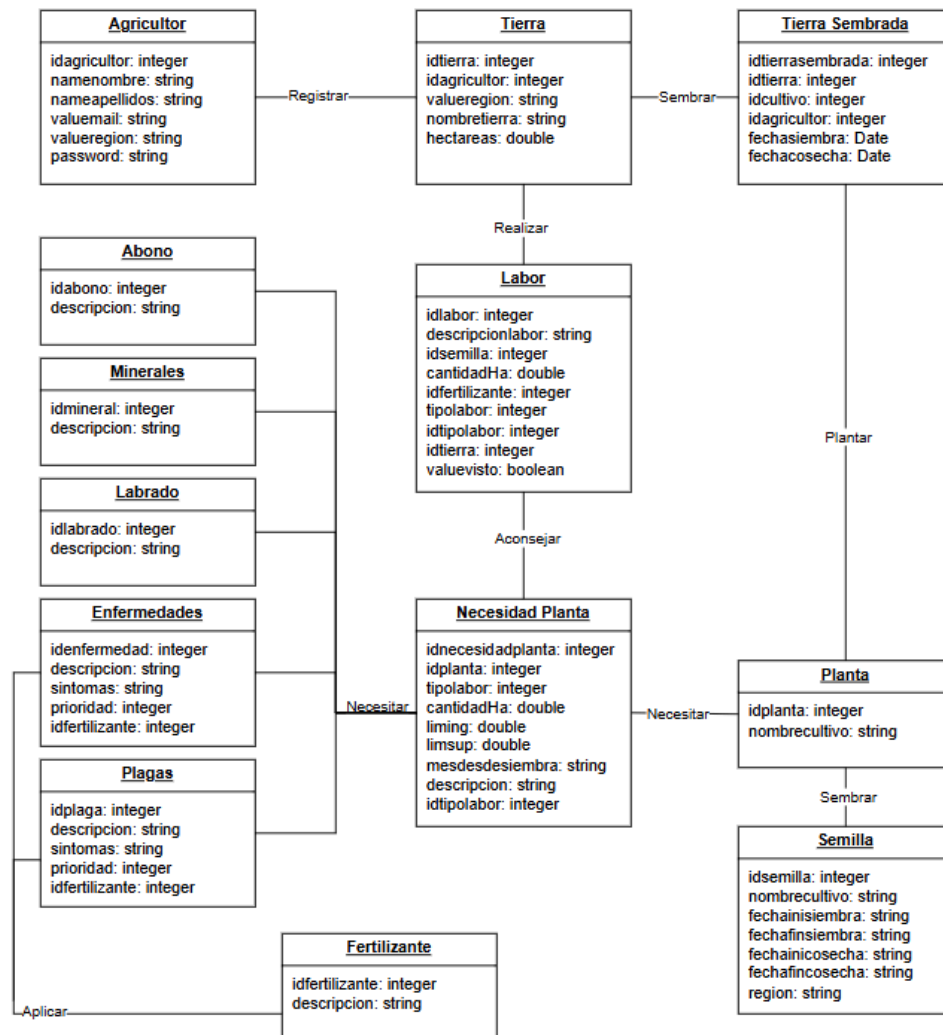


Diagrama: Modelo lógico de datos.

5.3 Arquitectura física

Se refiere a la estructura tangible del sistema. Ubicación de los componentes que componen el sistema y las conexiones entre ellos. Se describen aquí todos los módulos que permiten el funcionamiento correcto, completo y seguro del sistema:

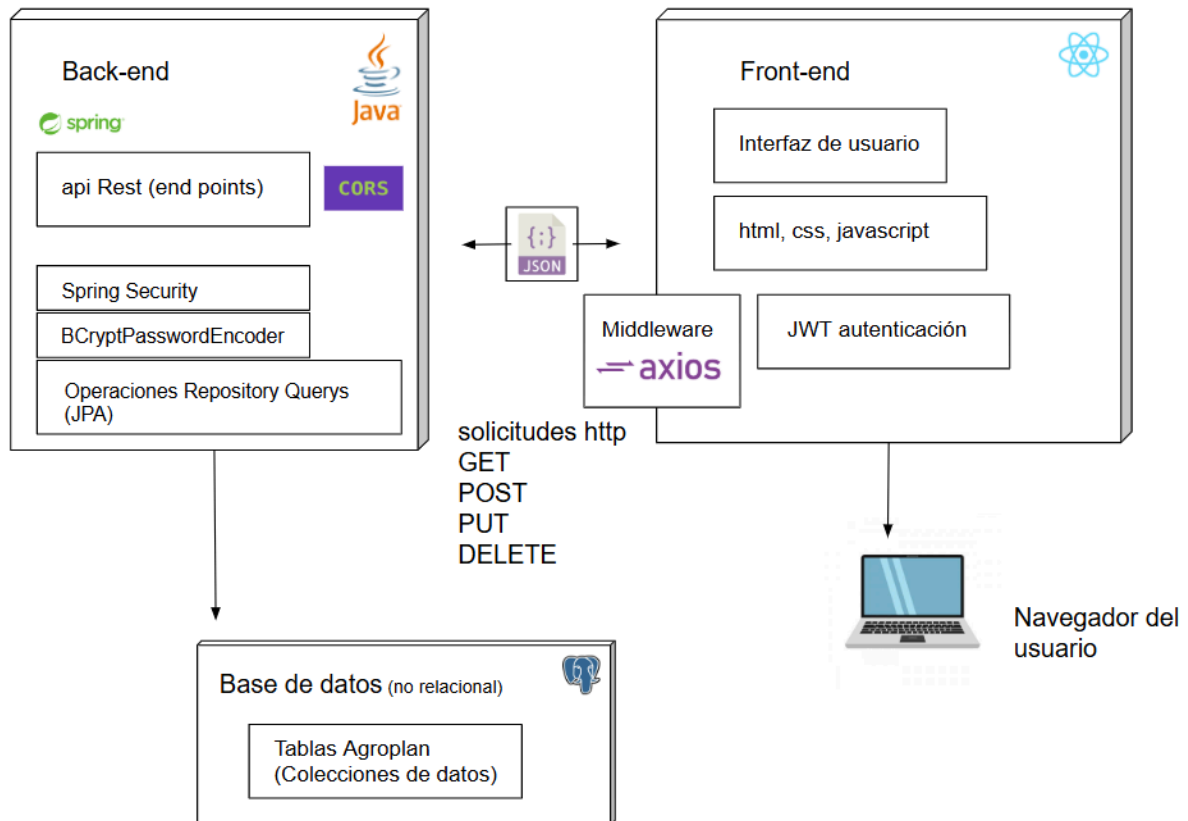


Diagrama físico

5.4 Arquitectura lógica detallada

En detalle a lo anterior, se describen aquí las funciones internas, los comportamientos específicos de cada componente y la lógica de negocio de Agroplan.

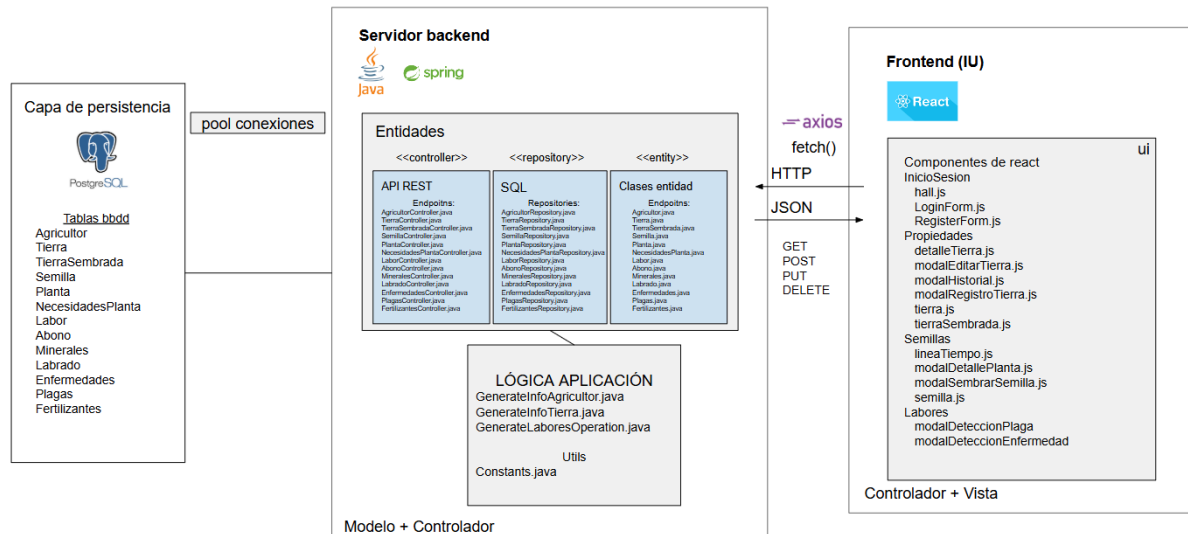


Diagrama lógico detallado

5.5 Arquitectura lógica no detallada

La arquitectura lógica detalla cómo funciona el sistema desde el punto de vista funcional, sin tener en cuenta el hardware. Aquí conseguimos una visión de alto nivel que nos describe los módulos que forman parte y cómo se comunican entre sí, el flujo de información y las peticiones entre un componente y otro.

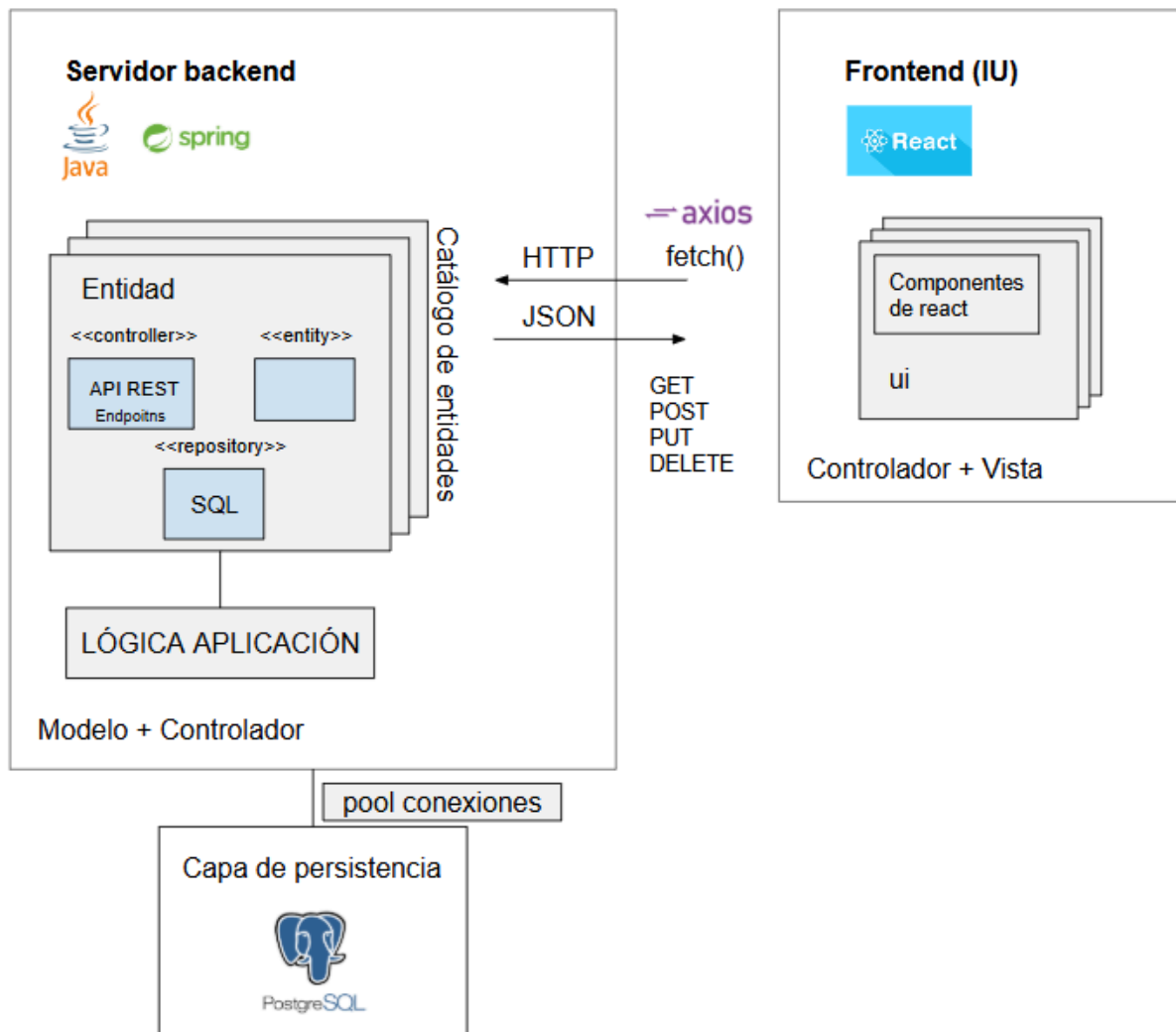


Diagrama lógico no detallado.

5.6 Tipo de modelo

Se describe en este apartado el patrón de diseño seguido (MVC, Modelo Vista Controlador).

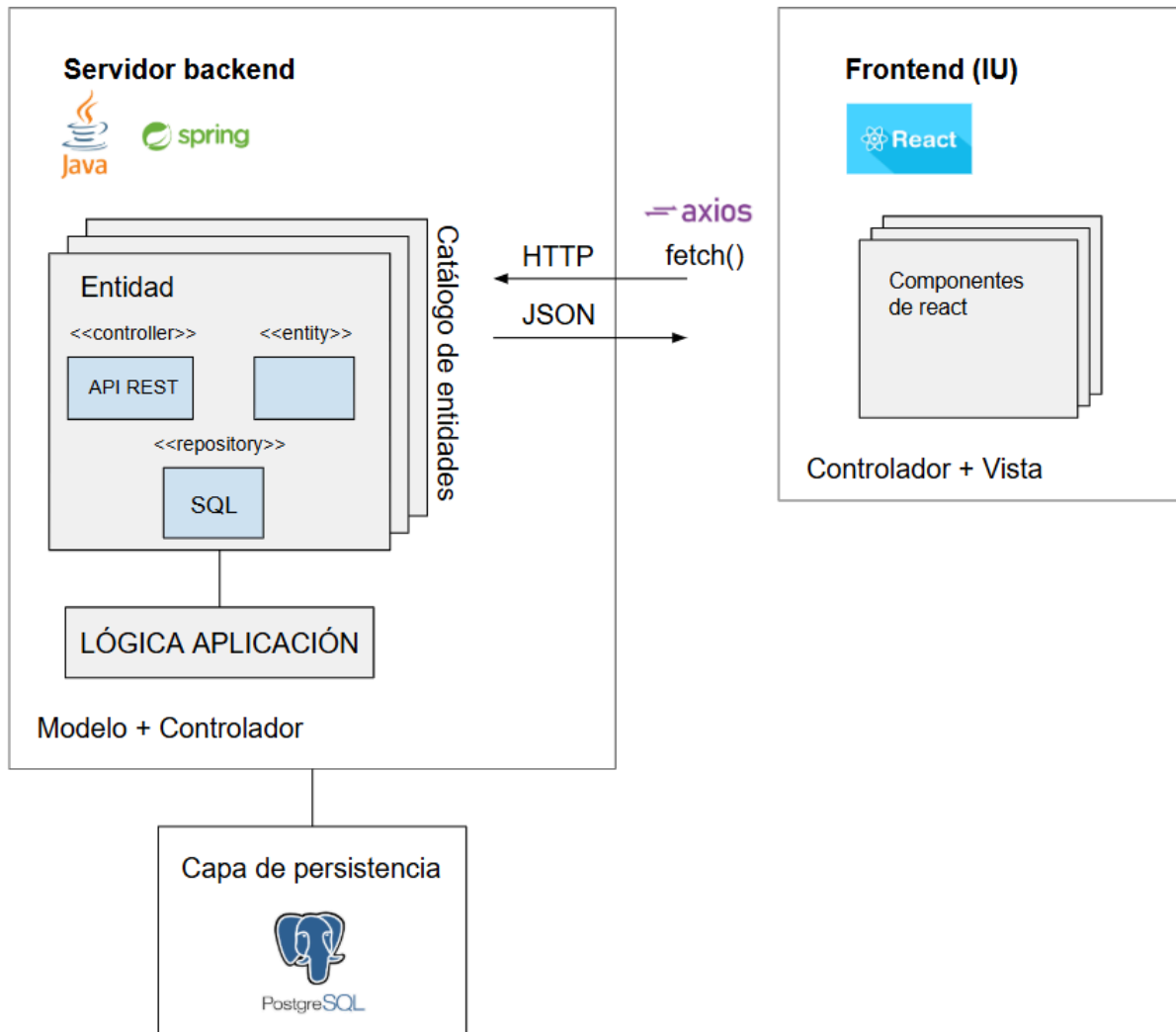


Imagen: Estructura del entorno de la aplicación.

Como se requiere en los objetivos se llevará a cabo un patrón MVC para describir la estructura de la aplicación.

Como **Modelo** del patrón MVC, se utilizará Java en el *backend* y también cumplirá funciones de **Controlador**, ya que se utilizará para procesar los datos de las llamadas desde el *frontend* y la interacción con la base de datos.

En el lado del cliente, se utilizará ReactJS para describir y generar la **Vista** pero además, también cumplirá funciones de **Controlador**, ya que hará llamadas al lado servidor que devolverá la respuesta que en el lado cliente se procesa.

5.7 Diagrama de secuencia

Se muestra aquí una muestra de diagrama de secuencia donde se podrá ver el comportamiento entre los diferentes componentes del sistema en una determinada acción. Su principal función es demostrar cómo los objetos se comunican entre sí mediante el intercambio de mensajes (respuestas, envío de información, documentos, etc).

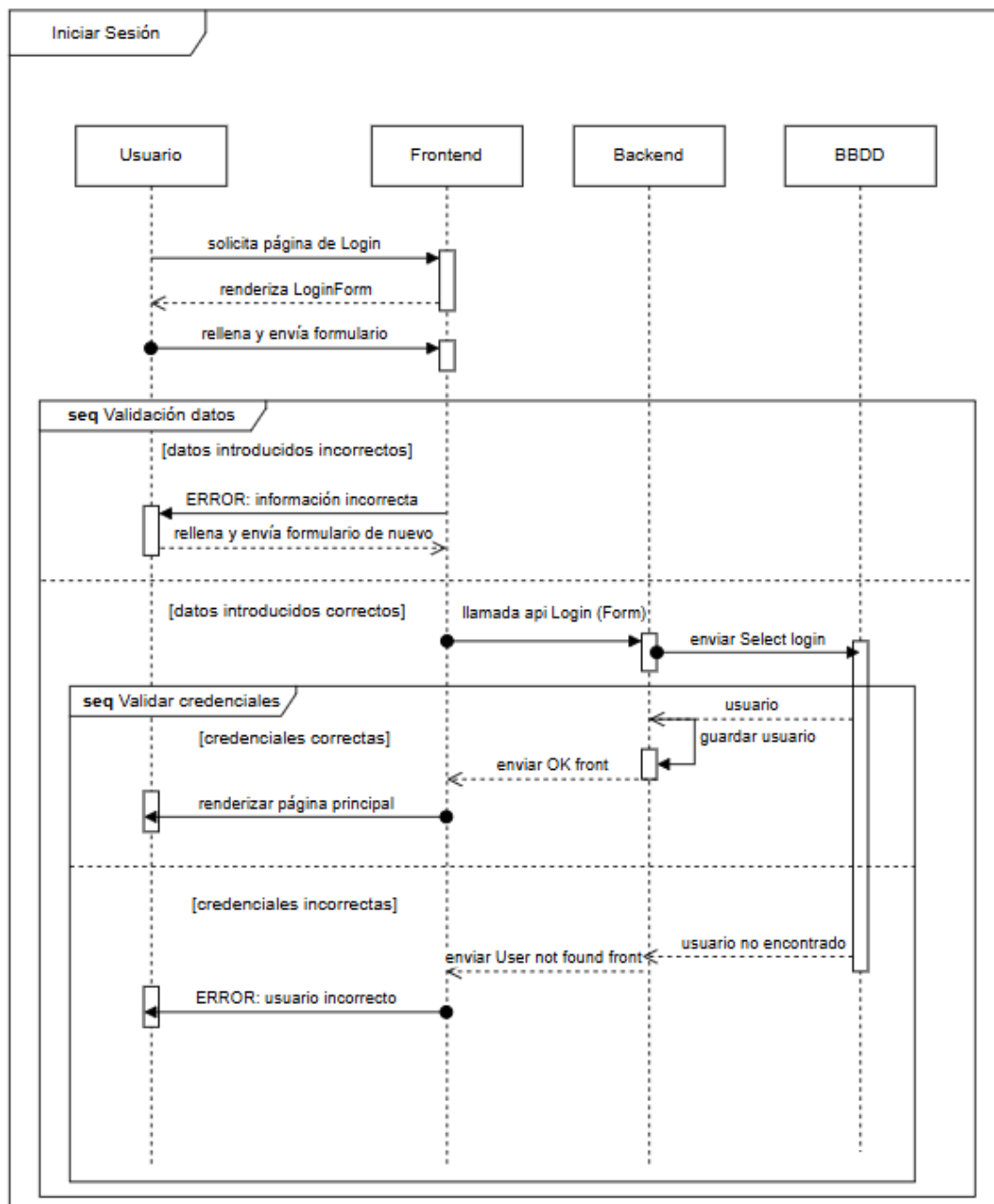


Diagrama de secuencia "Iniciar sesión".

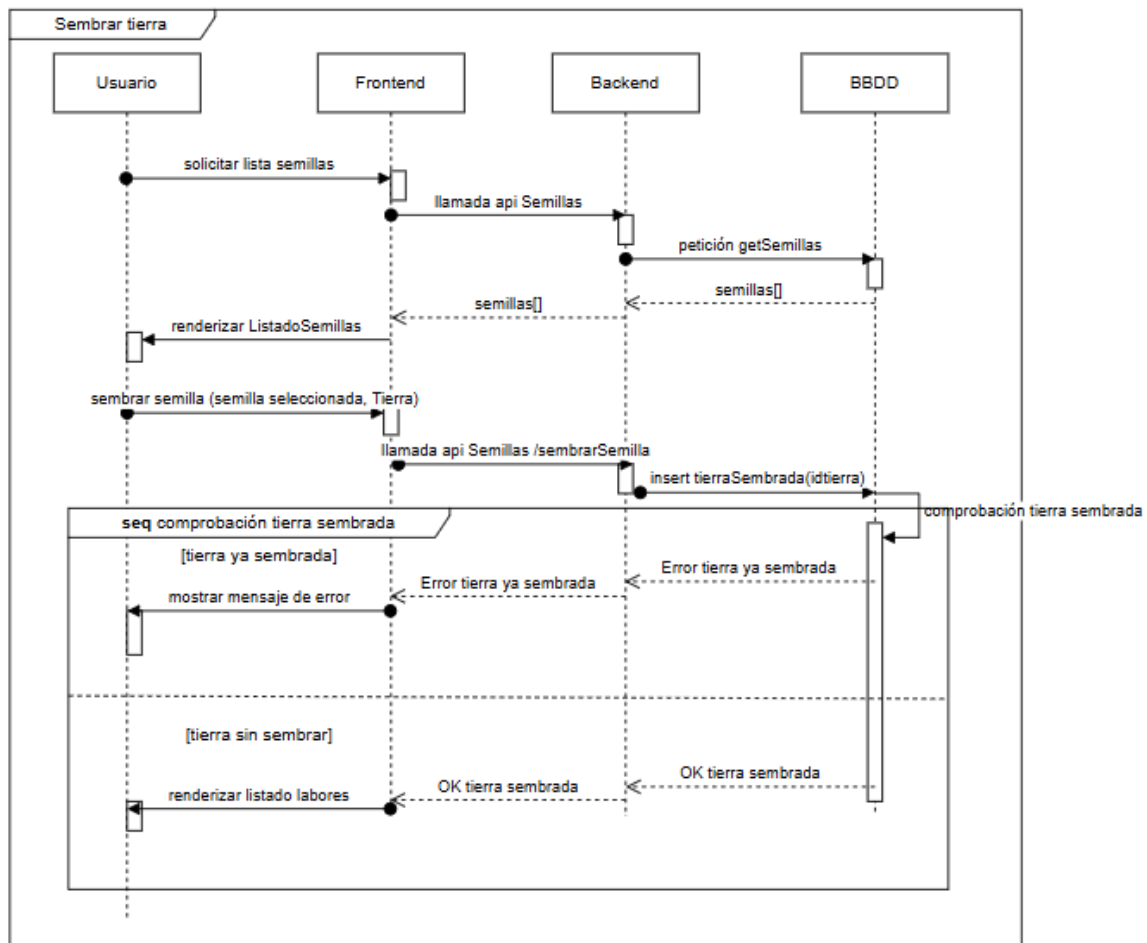


Diagrama de secuencia "Sembrar tierra"

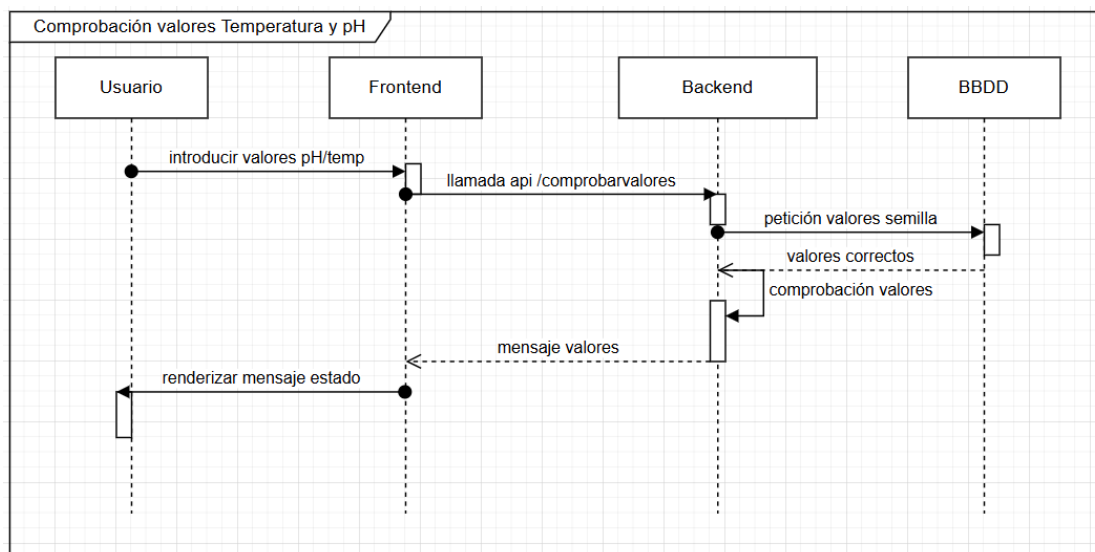


Diagrama de secuencia "Comprobación de Temp y pH"

Capítulo 6: Implementación

Se describe a continuación lo relacionado con la construcción y desarrollo del sistema.

6.1 *Backend* (capa de negocio)

El lenguaje utilizado para desarrollar la capa de negocio es Java. Y el *framework* elegido será Spring Boot.

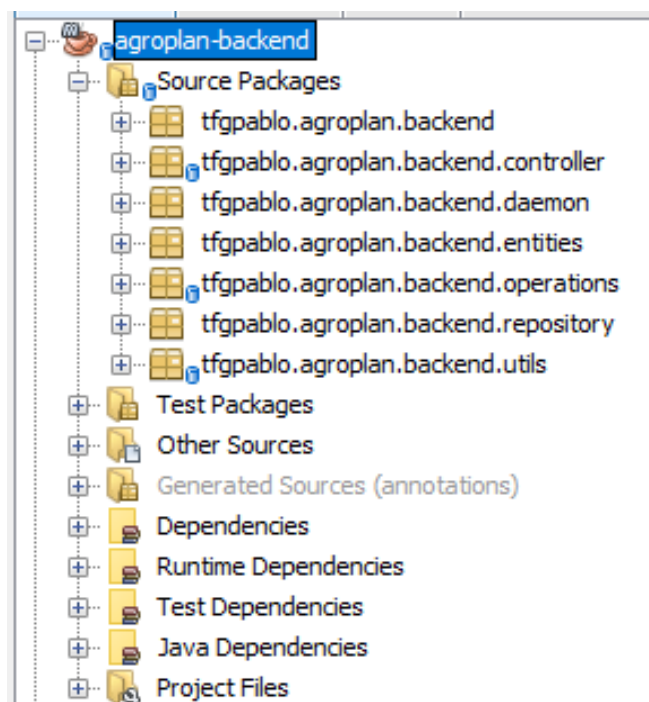


Imagen: Carpeta del proyecto.

El objetivo de Spring es minimizar la configuración manual y el tiempo de desarrollo. En el caso de AgroPlan, se utiliza JPA (Java Persistence API), lo que permite a Spring simplificar mucho el tiempo de desarrollo generando todo el acceso a bases de datos.

Una vez creadas las tablas en la base de datos, se crean las Entity classes y las clases Repository, que son interfaces que heredan de JpaRepository.

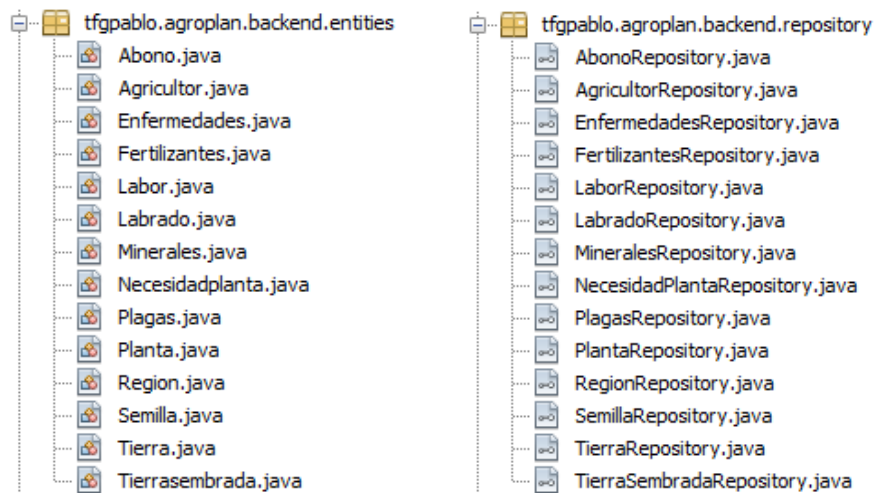


Imagen: paquetería Entities y Repositories.

A continuación se muestra un ejemplo de repository de Spring, en concreto el repositorio asociado a la entidad Planta.

```
package tfgpablo.agroplan.backend.repository;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.stereotype.Repository;
import tfgpablo.agroplan.backend.entities.Planta;

import java.util.List;
import java.util.Optional;

@Repository
public interface PlantaRepository extends JpaRepository<Planta, Integer> {

    // Encuentra una planta por su nombre de cultivo
    List<Planta> findByNombrecultivo(String nombrecultivo);

    // Encuentra una planta por su descripción
    List<Planta> findByDescripcion(String descripcion);

    // Encuentra una planta por el id
    Optional<Planta> findByIdplanta(Integer idplanta);

    List<Planta> findAllByIdplantaAsc();
}
```

Imagen: Repository ejemplo de AgroPlan.

Se instancian los métodos de esa manera y Spring se encarga de interpretar el nombre del método y los parámetros y crea internamente la *query* y la conexión con la base de datos.

Una vez tenemos la capa de persistencia configurada y funcional, se desarrolla la capa de negocio. Spring aquí necesita una clase llamada controller, conectada con el repository de

cada entidad y la clase entidad, donde crea una lista de *endpoints* y de operaciones (*API Rest*),

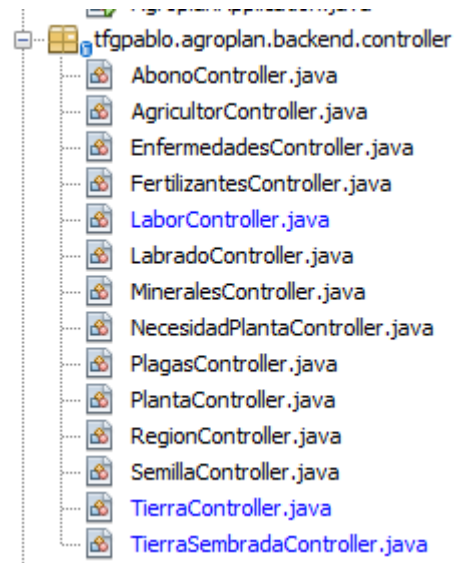


Imagen: Paquetería Controller.

Además de esa lógica de negocio de Agroplan, encontramos una paquetería donde realizamos operaciones (operations), complejas donde se complementan más de una entidad.

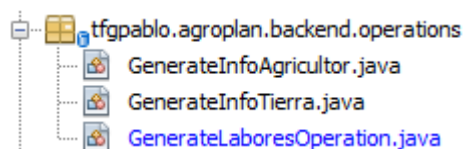


Imagen: Paquetería operations.

Importante recalcar la paquetería de Útiles (Utils), donde encontramos la clase de Constants, donde podemos acceder a información preestablecida para el uso de todo el modelo.

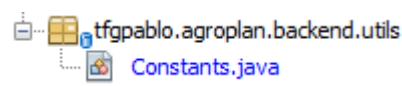


Imagen: Paquetería Utils.

6.2 Frontend (lado cliente)

El lenguaje utilizado para desarrollar el *frontend* es React JS, capaz de realizar diferentes componentes y poderlos reutilizar.

Principalmente toda la lógica del *frontend* se encuentra aquí:

Organizado por carpetas, tenemos los componentes relacionados con el inicio de sesión, las propiedades o tierras, las labores, y las semillas.

Para la conexión entre el *backend* y el *frontend* se utiliza *axios* y *fetch*. Se realizan las llamadas de la siguiente manera, y mediante la url, el *endpoint* correspondiente en el backend recibe la información y la procesa.

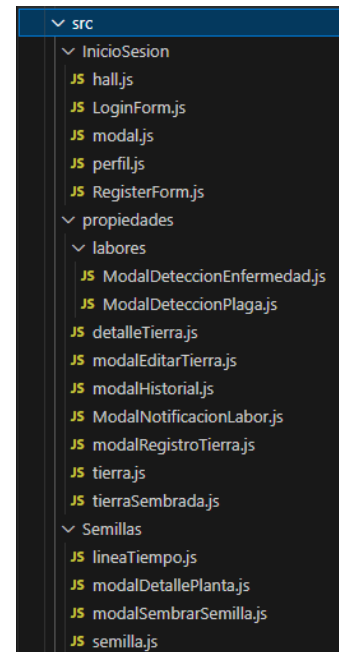


Imagen: Estructura del front

```
useEffect(() => {
  axios
    .get('http://localhost:8080/api/plantas')
    .then((response) => {
      if (Array.isArray(response.data)) {
        setPlanta(response.data);
        setFilteredSemillas(response.data);
      } else {
        console.error('La respuesta no es un array:', response.data);
      }
    })
    .catch((error) => {
      console.error('Error al cargar plantas:', error);
    });
}, []);
```

Imagen: Uso de Axios

Capítulo 7: Pruebas

Se describen en este capítulo las pruebas realizadas para comprobar el correcto funcionamiento de la aplicación. Se han realizado pruebas de caja negra o de caja blanca.

7.1 Pruebas de caja negra

Las pruebas de caja negra se centran en comprobar el funcionamiento externo del *software*, sin necesidad de conocer su estructura interna. Las pruebas de caja negra verifican que todos los requisitos se cumplan.

PCN_01: Registro de un nuevo usuario	
Objetivo	Verificar que un usuario nuevo se puede registrar correctamente.
Precondiciones	El usuario no deberá de existir ya en el sistema
Datos de entrada	Nombre: “Pablo” Apellidos: “Llorente Senín” Correo electrónico: “pablollorete02@gmail.com” Contraseña: “1234Pablo-” Tipo de usuario: Viticultor Región: Castilla y León
Acción esperada	Al pulsar el botón de Registrar, se deberá redirigir al modal de iniciar sesión y se debería de haber registrado el nuevo usuario.
Resultado	Positivo

Tabla: Pruebas caja negra “Registro nuevo usuario”

PCN_02: Inicio de sesión	
Objetivo	Verificar correctamente el inicio de sesión.
Precondiciones	El usuario deberá de estar registrado en el sistema.
Datos de entrada	Nombre de usuario: “pablollorete02@gmail.com” Contraseña: “1234Pablo-”
Acción esperada	Al pulsar el botón de iniciar sesión, se comprobará correctamente los datos introducidos y se redirigirá al usuario a la página principal de la aplicación.
Resultado	Positivo

Tabla: Pruebas caja negra "Inicio sesión"

PCN_03: Agricultor crea tierra	
Objetivo	Verificar que el usuario registra una tierra nueva en su listado.
Precondiciones	El usuario deberá de estar registrado en la aplicación, y localizarse en el componente de "listado de tierras".
Datos de entrada	Nombre de la tierra: "Camino largo N°2" Región: "Castilla y León" Tamaño (ha) : 10
Acción esperada	Al pulsar el botón de registrar se deberá de redirigir al usuario al listado de tierras cerrándose el modal, y mostrando el listado actualizado.
Resultado	Positivo

Tabla: Pruebas caja negra "Agricultor crea tierra"

PCN_04: Agricultor siembra tierra	
Objetivo	Verificar que el usuario siembra una tierra correctamente.
Precondiciones	La tierra que se siembra tiene que existir.
Datos de entrada	Semilla: Semilla seleccionada: "Trigo" Tierra: "Camino largo N°2" Fecha : 15-04-2025
Acción esperada	Al pulsar el botón de sembrar, si no está previamente sembrada, se deberá de sembrar correctamente. Se deberá haber creado la entrada en la base de datos en la tabla de TierraSembrada.
Resultado	Positivo

Tabla: Pruebas caja negra "Agricultor siembra tierra"

PCN_05: Agricultor cosecha tierra	
Objetivo	Verificar que el usuario cosecha una tierra correctamente.
Precondiciones	La tierra que se siembra tiene que existir y tiene que estar sembrada. La acción se realizará desde el componente de labores

	del agricultor.
Datos de entrada	Pulsar el botón de cosechar, y confirmar la acción.
Acción esperada	Al pulsar el botón de cosechar, se mostrará el mensaje de confirmación de la acción, y se confirmará la tierra cosechada.
Resultado	Positivo

Tabla: Pruebas caja negra "Agricultor cosecha tierra"

PCN_06: Comprobar temperatura y ph	
Objetivo	Verificar que la comprobación de temperatura y ph se realice correctamente.
Precondiciones	La tierra tiene que estar sembrada.
Datos de entrada	pH: 4.5 Tierra: 25
Acción esperada	El sistema comprobará los datos introducidos, con los datos ideales para esa semilla. El sistema mostrará un mensaje de corrección o mensaje de datos correctos sobre los introducidos.
Resultado	Positivo

Tabla: Pruebas caja negra "Comprobar temperatura y ph"

PCN_07: Detectar plaga o enfermedad	
Objetivo	Verificar que el usuario pueda detectar una plaga o enfermedad sobre una tierra.
Precondiciones	La tierra debe de existir, y el usuario accede a esa funcionalidad desde el detalle de la tierra.
Datos de entrada	plaga o enfermedad seleccionada
Acción esperada	Al pulsar el botón de detectar, se registrará correctamente la plaga o enfermedad detectada en la tabla de Labores de la base de datos. Además posteriormente se actualizará el listado de antecedentes.
Resultado	Positivo

Tabla: Pruebas caja negra "Detectar plaga o enfermedad"

7.2 Pruebas de caja blanca

Las pruebas de caja blanca, se centran en cambio en el análisis detallado del código fuente. Estas pruebas se encargan de comprobar la funcionalidad interna de cada módulo de la aplicación.

En estas pruebas se comprueba el comportamiento interno del software y así se detectan fallos en el código. En el desarrollo de AgroPlan se han realizado las siguientes pruebas de caja blanca.

- **Manejo de errores en las llamadas a API:** El objetivo es asegurarnos que todas las llamadas desde el *frontend* al *backend* se realicen correctamente en el *endpoint* correspondiente.
- **Entrada y salida de datos con el servidor de la base de datos.** El objetivo es comprobar que Spring realice las correspondientes *queries* y que las mismas reciban los datos requeridos. Además de que las *queries* están correctamente definidas, debe de haber una buena comunicación entre el sistema de bases de datos y el *backend* de la aplicación.
- **Conexión con bases de datos:** Se realizan pruebas para comprobar que la base de datos y el *backend* estén correctamente conectados.
- **Envío de información entre *frontend* y *backend*:** Además de comprobar que se referencian bien los endpoints, también es importante asegurarnos que los datos en las url y en los cuerpos de las peticiones GET, POST, etc. sean correctamente enviados.
- **Navegación entre componentes:** Nos queremos asegurar de que la navegación entre las diferentes funcionalidades dentro de la aplicación sea correcta. Cada botón deberá redirigir al usuario al componente correcto.
- **Redirección en cada acción:** En cada acción que realice el usuario, deberá redirigir al usuario al componente correcto.

Capítulo 8: Manuales

Se detallan a continuación las instrucciones necesarias para instalar y ejecutar el proyecto de AgroPlan. También se detalla la documentación y los módulos necesarios para desplegar el proyecto y empezar a trabajar.

8.1 Manual de instalación

Este manual describe las instrucciones necesarias para instalar y desplegar el proyecto de Agroplan.

En primer lugar se recomienda un sistema operativo Windows 10/11, macOS o linux, para la instalación.

Como herramientas necesarias se listan las siguientes:

- Java JDK 17 o superior
- Maven
- Node.js y npm
- PostgreSQL
- Visual Studio Code
- GitHub

Instalar un JDK, se recomienda versión 17 o superior.

<https://www.oracle.com/java/technologies/javase/jdk17-archive-downloads.html>

Una vez esté instalado, verifica la instalación:

```
java -version  
mvn -version
```

Si se tiene instalado Maven, descárgalo del sitio web:

<https://maven.apache.org/download.cgi>, y siga los pasos de la instalación.

Instalar también Node.js , descargándolo desde:

<https://nodejs.org/en>

Para verificar la instalación de Node.js pueden ejecutarse los siguientes comandos:

```
node -v  
npm -v
```

Se sigue con la instalación del sistema de bases de datos para proporcionar a la aplicación la capa de persistencia, descargando PostgreSQL del sitio web: [\[PostgreSQL\]](#)

Una vez se tenga postgresQL descargado, se procederá a crear una base de datos que se llamará: **agroplan-database**.

Comprobar que las credenciales de acceso a la BD PostgreSQL son las correctas:

Usuario: postgres

Password: root

Una vez instaladas las herramientas que soportan el entorno de ejecución, se procederá a construir el proyecto.

Antes de comenzar se recomienda la instalación de VsCode para el desarrollo del *frontend* y Netbeans, para el desarrollo del backend de AgroPlan.

Enlace de descarga en la página oficial de Netbeans: [\[Netbeans\]](#)

Enlace de descarga de Visual Studio Code: [\[VSC\]](#)

Una vez se tenga el entorno de desarrollo preparado continuaremos con el proyecto. El proyecto está alojado de un repositorio de GitHub, cuyos enlaces de acceso son los siguientes:

Enlace para **agroplan-backend**:

<https://github.com/pablolls/agroplan-backend.git>

Enlace para **agroplan-ui**:

<https://github.com/pablolls/agroplan-ui.git>

Cómo se utiliza **Spring**, será necesario configurar el archivo del proyecto llamado:
application.properties,

y, configurándolo con los parámetros de conexión a la base de datos:


```
spring.datasource.url=jdbc:postgresql://localhost:5432/agroplan_db
spring.datasource.username=postgres
spring.datasource.password=admin123
```

Así estaríamos configurando la base de datos para poder utilizarla desde el proyecto.

Una vez compilado el proyecto, se ejecutará y desplegará atendiendo al puerto 8080.

Ahora para configurar el *frontend*, se abrirá en VSCode el proyecto **agroplan-ui** clonado previamente.

```
cd agroplan-ui
npm install

npm start
```

Con esta secuencia, se desplegará la capa de interfaz de usuario en el puerto 3000.

Finalmente, se accede automáticamente a <http://localhost:3000> y se verifica que la aplicación se comunica con el *backend*.

La aplicación del *backend* es básicamente una API Rest, con varios *endpoints*, por lo que accediendo a las rutas correctas desde el 8080, podremos comprobar que funcionan las peticiones, para esto último se recomienda también la aplicación de Postman. [\[postman\]](#)

8.2 Manual de usuario

Manual diseñado para guiar al usuario agricultor en el uso de la aplicación de AgroPlan, una solución de gestión agrícola. A continuación se describe el funcionamiento de la aplicación y todas sus funcionalidades.

Lo primero al entrar en la aplicación verá la página de inicio donde podrá iniciar sesión o registrarse en caso de que no lo haya hecho todavía.

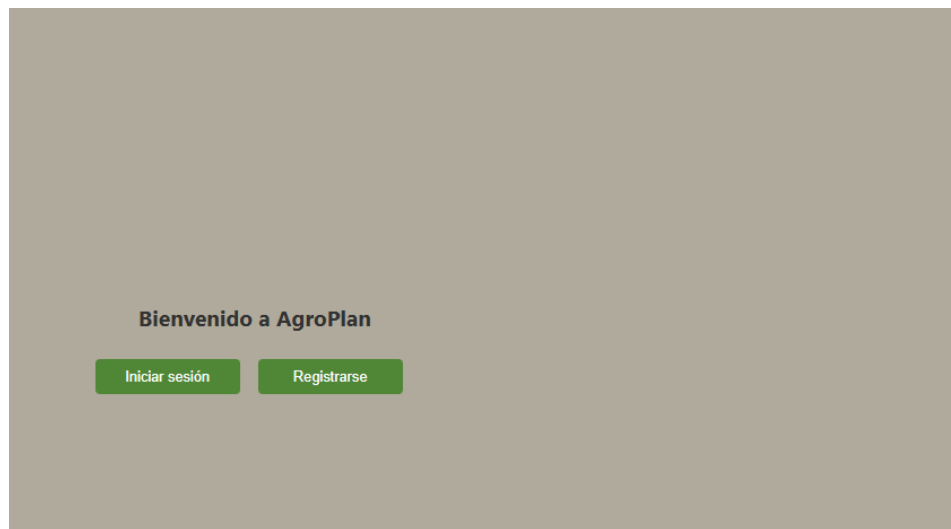


Imagen: Manual de usuario "Login".

Como bien se describe en la página principal, el usuario que llega podrá iniciar sesión rellenando los campos de inicio de sesión o bien, registrarse y rellenar el siguiente formulario.

Imagen: Formularios Inicio sesión y registro.

En caso de no hacer un correcto inicio de sesión, el usuario recibirá un mensaje de error.

Una vez iniciada la sesión con el nombre de usuario y la contraseña correcta, el usuario entrará en la aplicación y accede al listado de labores, funcionalidad donde verá una lista de todas las siembras que ha realizado en sus tierras.

Nombre	Localización	Hectáreas	Estado actual	Fecha Siembra	Fecha Cosecha
Camino Zarzuela N1	Castilla y León	1.4	Cosechado	2025-03-24	2025-03-24
Camino Zarzuela N1	Castilla y León	1.4	Cosechado	2025-03-22	2025-03-24
Camino Zarzuela N1	Castilla y León	1.4	Cosechado	2025-03-19	2025-03-19
Camino Zarzuela N1	Castilla y León	1.4	Cosechado	2025-03-19	2025-03-19
Camino Zarzuela N1	Castilla y León	1.4	Cosechado	2025-03-19	2025-03-19
Camino Zarzuela N1	Castilla y León	1.4	Cosechado	2025-03-19	2025-03-19
Casa Nuevas	Castilla y León	10	Sembrado	2025-03-14	
Temeroso N1	Castilla y León	21	Sembrado	2025-03-12	
Camino Zarzuela N1	Castilla y León	1.4	Cosechado	2025-03-12	2025-03-19
Casa Nuevas	Castilla y León	10	Cosechado	2025-02-22	2025-03-10
Camino Zarzuela N1	Castilla y León	1.4	Cosechado	2025-02-18	2025-03-02
Camino Zarzuela N1	Castilla y León	1.4	Cosechado	2025-02-04	2025-02-17

Imagen: Listado de labores del agricultor.

Desde esta barra de navegación podrá el usuario filtrar su búsqueda en este listado:



Imagen: Barra navegación listado labores.

El usuario podrá filtrar ese listado si lo que quiere es ver sólo las tierras que están sembradas o próximas a cosechar o en cambio si solo necesitar acceder a las cosechas.

Tus labores

Todas	Sembradas	Próximas	Cosechadas			
Nombre	Localización	Hectáreas	Estado actual	Fecha Siembra	Fecha Cosecha	
Casa Nuevas	Castilla y León	10	◆ Sembrado	2025-03-14		Cosechar
Temeroso N1	Castilla y León	21	◆ Sembrado	2025-03-12		Cosechar

Imagen: Listado “Sembradas” labores.

Tus labores

Todas	Sembradas	Próximas	Cosechadas			
Nombre	Localización	Hectáreas	Estado actual	Fecha Siembra	Fecha Cosecha	
Camino Zarzuela N1	Castilla y León	1.4	◆ Próximo a Sembrar	2025-04-25		

Imagen: Listado “Próximas a sembrar” labores.

Tus labores

Todas	Sembradas	Próximas	Cosechadas			
Nombre	Localización	Hectáreas	Estado actual	Fecha Siembra	Fecha Cosecha	
Camino Zarzuela N1	Castilla y León	1.4	◆ Cosechado	2025-02-04	2025-02-17	
Camino Zarzuela N1	Castilla y León	1.4	◆ Cosechado	2025-02-18	2025-03-02	
Casa Nuevas	Castilla y León	10	◆ Cosechado	2025-02-22	2025-03-10	
Camino Zarzuela N1	Castilla y León	1.4	◆ Cosechado	2025-03-12	2025-03-19	
Camino Zarzuela N1	Castilla y León	1.4	◆ Cosechado	2025-03-19	2025-03-19	
Camino Zarzuela N1	Castilla y León	1.4	◆ Cosechado	2025-03-19	2025-03-19	
Camino Zarzuela N1	Castilla y León	1.4	◆ Cosechado	2025-03-19	2025-03-19	
Camino Zarzuela N1	Castilla y León	1.4	◆ Cosechado	2025-03-19	2025-03-19	
Camino Zarzuela N1	Castilla y León	1.4	◆ Cosechado	2025-03-22	2025-03-24	
Camino Zarzuela N1	Castilla y León	1.4	◆ Cosechado	2025-03-24	2025-03-24	

Imagen: Listado “Cosechadas” labores

Como bien se puede ver, las tierras que el usuario tiene sembradas, la aplicación le mostrará la opción al usuario de cosechar.

Casa Nuevas	Castilla y León	10	◆ Sembrado	2025-03-14	Cosechar
Temeroso N1	Castilla y León	21	◆ Sembrado	2025-03-12	Cosechar

Imagen: Fila labor, botón cosechar.

Como nos hemos podido dar cuenta, para poder sembrar una tierra, primero el agricultor debe crearla. Haciendo click en el botón del “tractor” de la barra de navegación lateral, nos llevará al listado de tierras registradas por el agricultor, donde podremos añadir nuevas tierras, modificar las ya existentes e incluso eliminarlas.



Imagen: Barra navegación lateral.

Sus propiedades



Nombre	Localización	Hectáreas	Estado			
Casa Nuevas	Castilla y León	10	◆ Sembrado	Detalles	✎ Editar	🗑 Eliminar
Camino Zarzuela N1	Castilla y León	1.4	◆ Próximo a Sembrar !	Detalles	✎ Editar	🗑 Eliminar
Temeroso N1	Castilla y León	21	◆ Sembrado	Detalles	✎ Editar	🗑 Eliminar

Imagen: Listado propiedades agricultor.

Si le damos al botón superior derecho “ + “, se nos abrirá el modal de registro de nuevas tierras. Donde nos pedirá los datos de la tierra a registrar.

Registrar nueva propiedad

Nombre de la propiedad:

Localización (Región):

Tamaño (ha):

Registrar

Imagen: Modal registro nueva propiedad.

Editar propiedad

Nombre de la propiedad:

Localización (Región):

Tamaño (ha):

Guardar cambios

Imagen: Modal edición propiedad.

En este listado, de vez en cuando aparecerá un icono de una exclamación en movimiento en alguna tierra, se refiere a esto:

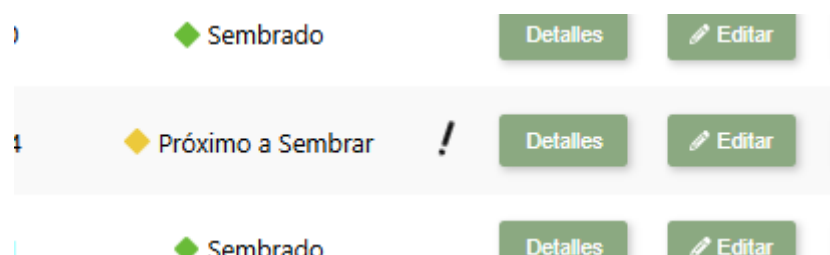


Imagen: Indicador "tareas por hacer"

Si hacemos click ahí, nos abrirá un modal con las tareas recomendadas de hacer sobre la tierra donde se encuentre esa exclamación.

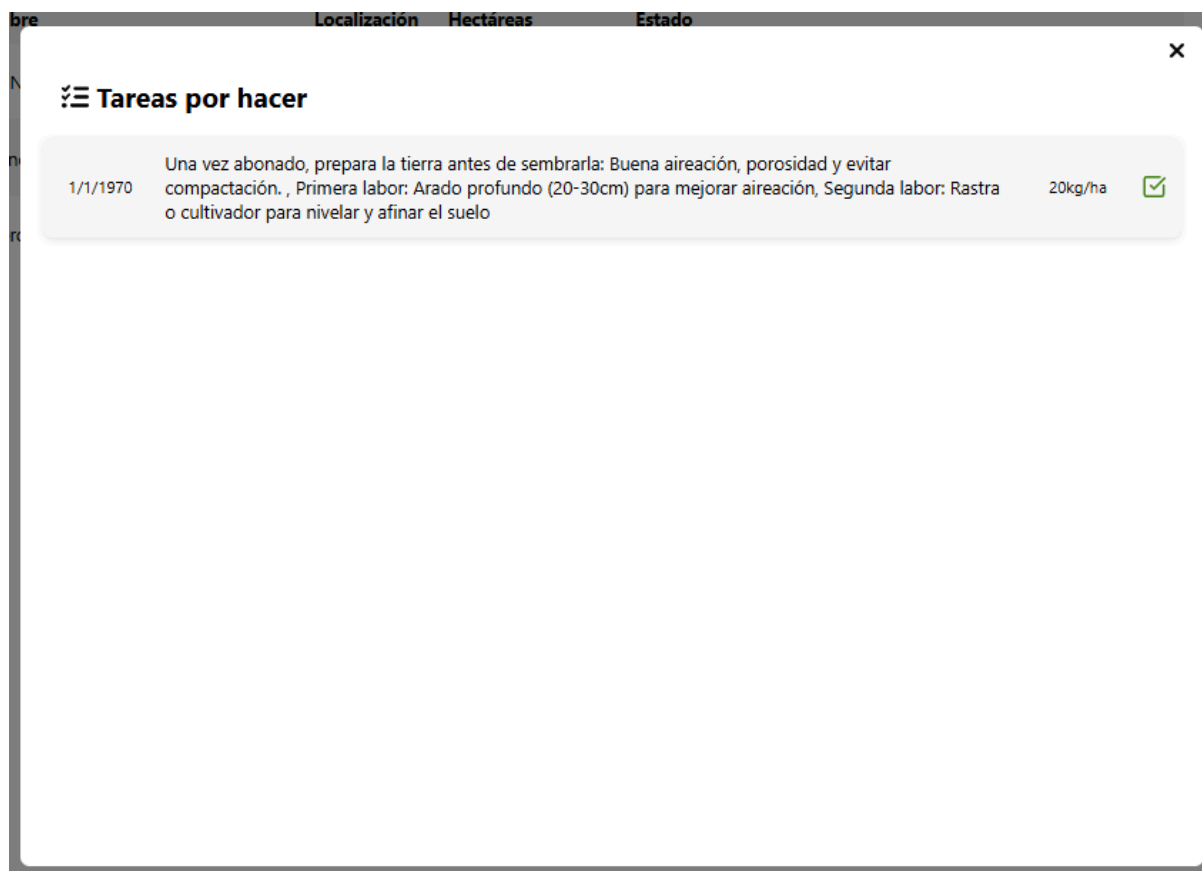


Imagen: Modal “tareas por hacer”

Como se ve, la tierra está como “próxima para sembrar”, y eso indica que se ha planificado la siembra, por lo tanto Agroplan, te recomienda tareas y labores que hacer antes de sembrar, para que cuando llegue el día de la siembra, la semilla tenga las mejores condiciones posibles.

Si se pulsa en el *tick* de la derecha de la tarea, se marcará como hecha y dejará de aparecer en el tablón de tareas. 

Cuando queramos eliminar una tierra, el sistema nos advertirá si la tierra está sembrada o no.

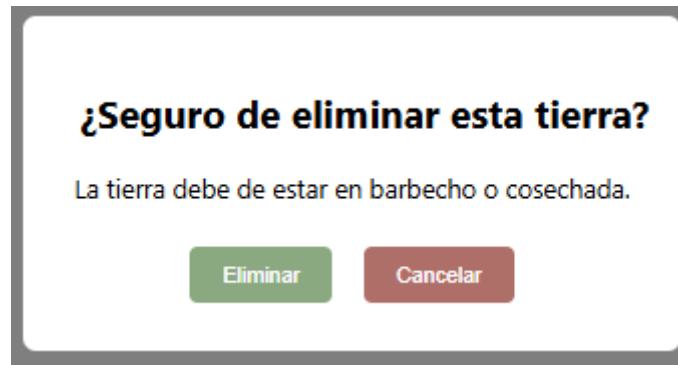


Imagen: Modal confirmación "Eliminar tierra".

Ya que no puedes eliminar una tierra si está sembrada actualmente:

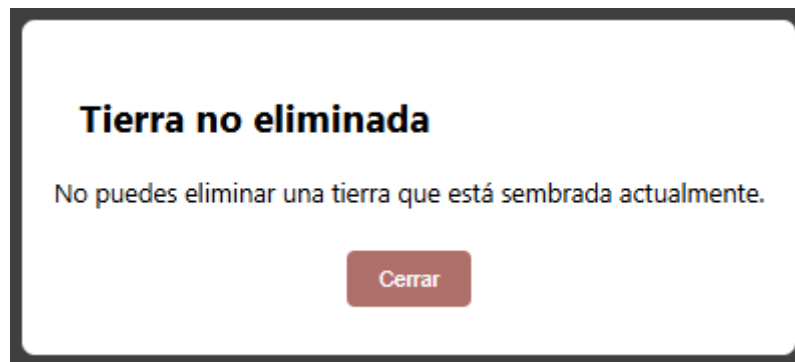


Imagen: Modal "Tierra no eliminada"

También hemos podido observar que en el listado podemos acceder a un detalle específico de cada tierra. Se nos abrirá un modal con las siguientes partes.

En caso de que el usuario detecte una plaga o una enfermedad en su plantación, podrá acceder al detalle donde se le mostrarán las plagas o enfermedades comunes al tipo de semilla que está sembrada en la tierra e identificará los síntomas.

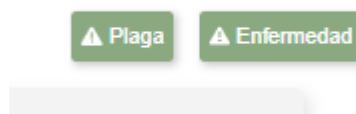


Imagen: Botones detectar plaga y enfermedad.

Una vez pulse en plaga o en enfermedad, se le abrir dicho modal:

Detección de Plagas				
1 Rotación de cultivos: Evita que las plagas y enfermedades se instalen en el suelo.				
Plaga	Síntomas	Riesgo	Fitosanitario	
Pulgón del trigo	Hojas enrolladas y amarillentas. Reducción del crecimiento y transmisión de virus.	Media	Jabón potásico	Detectar
Gusano trozador	Plántulas cortadas al nivel del suelo y reducción de la densidad de plantas.	Alto	Cebos con insecticida	Detectar
Trips del trigo	Hojas plateadas y secas. Granos deformados y de bajo peso.	Alto	Insecticidas (spinosad, acetamiprid)	Detectar

Imagen: Modal detección de plagas

Detección de Enfermedades				
1 Rotación de cultivos: Evita que las plagas y enfermedades se instalen en el suelo.				
Enfermedad	Síntomas	Riesgo	Fitosanitario	
Fusariosis de la espiga	Espigas con granos deformados y blancos, puede generar micotoxinas peligrosas para el consumo.	Baja	Fungicidas	Detectar
Roya del trigo	Manchas pústulas naranjas o amarillas en hojas y tallos. Reduce la fotosíntesis y el rendimiento del grano.	Alto	Fungicidas sistémicos (triazoles, estrobilurinas)	Detectar
Fusariosis de la espiga	Espigas con granos deformados y blancos, puede generar micotoxinas peligrosas para el consumo.	Baja	Fungicidas	Detectar
Septorios del trigo	Manchas marrones en hojas con puntos negros en el centro. Disminuye la fotosíntesis y el rendimiento del cultivo.	Alto		Detectar
Tizón bacteriano del trigo	Estrias marrones en las hojas con aspecto húmedo, marchitamiento progresivo de la planta.	Alto		Detectar
Virus del mosaico del trigo	Manchas amarillas en forma de mosaico en las hojas. Reducción del crecimiento y menor producción del grano.	Alto		Detectar

Imagen: Modal detección de enfermedades.

Aquí se listará el nombre de la enfermedad o plaga, los síntomas para identificar, el nivel de riesgo y los fitosanitarios recomendados para paliar la enfermedad o plaga. Una vez el usuario reconozca la enfermedad o plaga, podrá pulsar el botón de “Detectar”, para añadir este incidente en la lista de antecedentes de la tierra, el cual también se puede acceder a él en el detalle antes abierto:

Antecedentes

Tipo	Descripción	Fertilizante recomendado	Fecha detección
Enfermedad	Roya del trigo	Fungicidas sistémicos (triazoles, estrobilurinas)	17-03-2025
Plaga	Pulgón de la cebada	Insecticidas. Imidacloprid o aceites vegetales.	21-03-2025
Plaga	Pulgón de la cebada	Insecticidas. Imidacloprid o aceites vegetales.	23-03-2025

Imagen: Listado antecedentes tierra.

Además del listado de antecedentes de la tierra, se muestra también el listado historial de siembras y cosechas de la tierra:

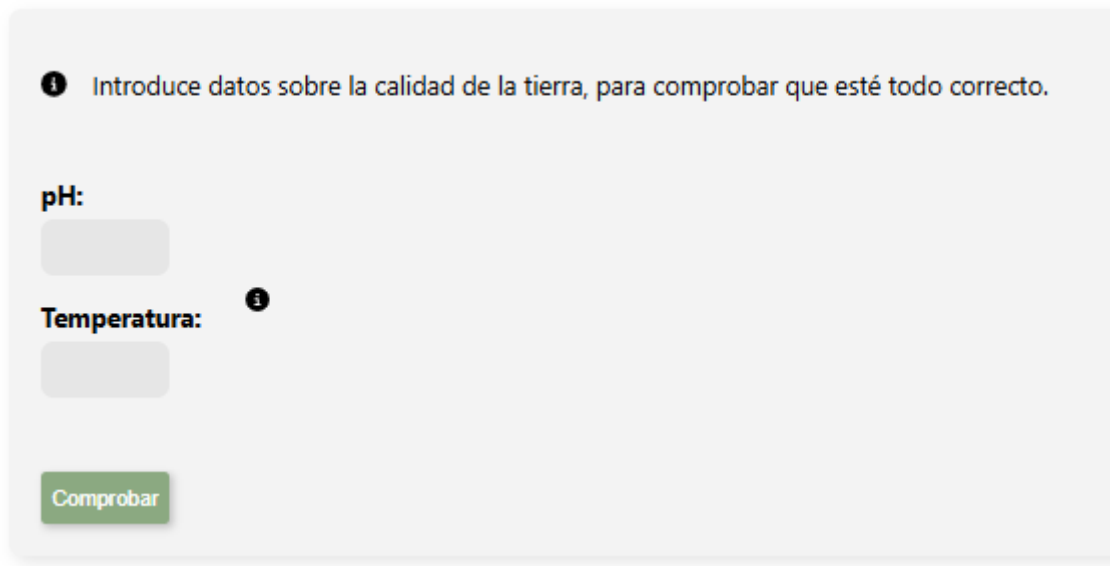
Historial

	Cultivo	Región	Hectáreas	Sembrado	Cosechado
Actual	trigo	Castilla y León	1.4	2025-04-25	
	cebada	Castilla y León	1.4	2025-03-24	2025-03-24
	cebada	Castilla y León	1.4	2025-03-22	2025-03-24
	trigo	Castilla y León	1.4	2025-03-19	2025-03-19
	cebada	Castilla y León	1.4	2025-03-19	2025-03-19
	quinoa	Castilla y León	1.4	2025-03-19	2025-03-19
	arroz	Castilla y León	1.4	2025-03-19	2025-03-19
	cebada	Castilla y León	1.4	2025-03-12	2025-03-19
	quinoa	Castilla y León	1.4	2025-02-18	2025-03-02
	avena	Castilla y León	1.4	2025-02-04	2025-02-17

Imagen: Listado historial tierra.

Marcada como el estado actual de la tierra.

Por último, se le permite al usuario poder comprobar diferentes parámetros del estado del suelo de la tierra. El pH y la temperatura, aspectos imprescindibles para un correcto crecimiento de la semilla.



Introduce datos sobre la calidad de la tierra, para comprobar que esté todo correcto.

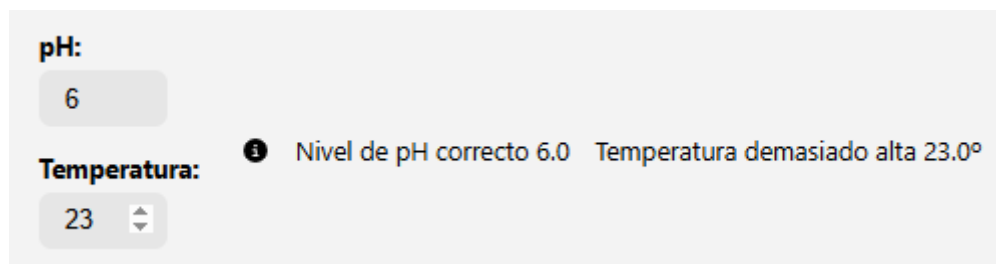
pH:

Temperatura:

Comprobar

Imagen: Formulario comprobar estado tierra.

El usuario agricultor, podrá introducir los valores recabados de sus mediciones y el sistema le responderá con un mensaje de estado de su tierra.



pH:

Temperatura:

Nivel de pH correcto 6.0 Temperatura demasiado alta 23.0°

Imagen: Resultado comprobar estado tierra.

Y por último, se describe la última pantalla de funcionalidades de Agroplan, y trata del listado de semillas que se pueden sembrar en Agroplan:

Buscar semillas

Planta	Descripción		
trigo	Trigo (<i>Triticum</i> spp.): planta cereal perteneciente a la familia Poaceae, cultivada por su grano comestible, u...	+	Sembrar
cebada	Cebada (<i>Hordeum vulgare</i>): planta cereal perteneciente a la familia Poaceae, utilizada principalmente par...	+	Sembrar
avena	La avena (<i>Avena sativa</i>) es un cereal cultivado principalmente por su alto valor nutricional. Se utiliza tanto...	+	Sembrar
centeno	Centeno (<i>Secale cereale</i>): cereal de la familia Poaceae, cultivado en climas fríos y utilizado en la producció...	+	Sembrar
arroz	Arroz (<i>Oryza sativa</i>): planta cereal de la familia Poaceae, cultivada en zonas húmedas y cálidas, principal f...	+	Sembrar
maíz	Maíz (<i>Zea mays</i>): cereal de la familia Poaceae, originario de América, utilizado como alimento humano, fo...	+	Sembrar
quinoa	Quinoa (<i>Chenopodium quinoa</i>): planta pseudocereal de la familia Amaranthaceae, rica en proteínas, amin...	+	Sembrar
judía blanca	Judía blanca (<i>Phaseolus vulgaris</i>): leguminosa de la familia Fabaceae, consumida por su alto contenido en...	+	Sembrar
habas	Habas (<i>Vicia faba</i>): leguminosa de la familia Fabaceae, cultivada por sus semillas comestibles, que son una...	+	Sembrar
lenteja	Lenteja (<i>Lens culinaris</i>): leguminosa de la familia Fabaceae, consumida por sus semillas ricas en proteínas, ...	+	Sembrar
garbanzo	Garbanzo (<i>Cicer arietinum</i>): leguminosa de la familia Fabaceae, cultivada por sus semillas comestibles, ric...	+	Sembrar
guisante seco	Guisante seco (<i>Pisum sativum</i>): leguminosa de la familia Fabaceae, cultivada por sus semillas secas comes...	+	Sembrar

Página 1

Imagen: Listado semillas.

Incorpora un buscador en la parte superior donde se puede hacer una búsqueda asíncrona del nombre de la planta, sin necesidad de dar al botón de buscar.

El listado lo conforman 4 columnas, el nombre de la planta, una pequeña descripción, un botón de detalle y el botón de sembrar.

Si se abre el detalle, accederemos 4 bloques principales de información acerca del cuidado de la semilla seleccionada.

Recomendaciones sobre el cultivo de cebada

Imagen: Título detalle semilla.

Abonos, minerales, labrados de la tierra recomendados y recomendaciones sobre la calidad del suelo.

Entre estos datos se listarán todos los minerales recomendados para el perfecto nacimiento y crecimiento de la semilla, además de recomendaciones sobre el tipo de suelo ideal para la semilla seleccionada.

Abonos:

Compost, Estiércol, Humus de lombriz

Minerales:

Nitrógeno (N),	Fósforo (P),	Potasio (K),	Azufre (S),	Zinc (Zn),	Hierro (Fe),
----------------	--------------	--------------	-------------	------------	--------------

Labrado de la tierra

Buena aireación, porosidad y evitar compactación.
Primera labor: Arado profundo (20-30cm) para mejorar aireación
Segunda labor: Rastra o cultivador para nivelar y afinar el suelo

Calidad del suelo

Terreno franco-arenoso o franco-limoso, bien drenado. Evitar suelos arcillosos y pesados, ya que pueden compactarse y afectar al crecimiento radicular.

Temperatura ☺ Debe de estar entre 10° y 22°

pH 🛠 Debe de estar entre 6 y 7.5

Para suelos ácidos: aplicar cal agrícola; Para suelos alcalinos: aplicar azufre elemental o materia orgánica

Materia orgánica de entre 2% y 3%.

Imagen: Detalle información acerca de la semilla.

Si pulsamos en el calendario de la derecha del título, nos abrirá un calendario de siembra y de cosecha recomendado para la región de la tierra.



Imagen: Periodos de siembra y cosecha semilla.

Y un gran listado de posibles enfermedades y plagas de la semilla seleccionada:

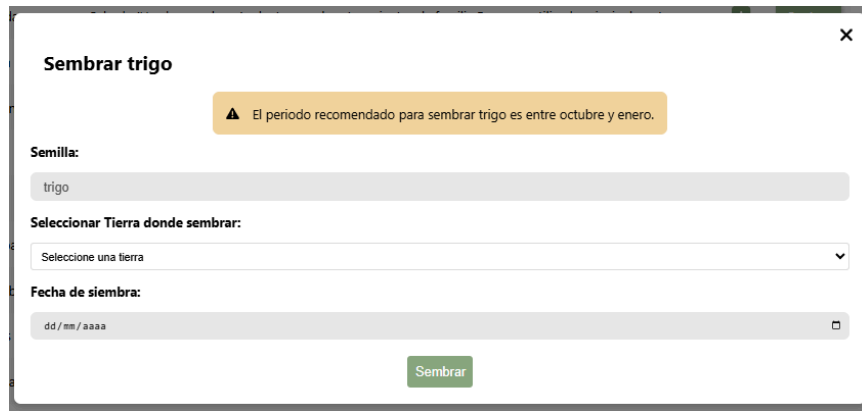
 Posibles enfermedades			
Enfermedad	Síntomas	Prioridad	Fertilizante
Roya de la cebada	Manchas pústulas naranjas o amarillas en hojas y tallos. Reduce la fotosíntesis y el rendimiento del grano.	▲ Baja prioridad	Fungicidas sistémicos (triazoles, estrobilurinas)
Oídio	Polvo blanco en hojas	▲ Baja prioridad	Fungicida a base de azoxistrobina y propiconazol.
Fursariosis de la espiga	Espigas blanquecinas y granos deformados	▲ Baja prioridad	Fungicidas
Roya de la cebada	Manchas pústulas naranjas o amarillas en hojas y tallos. Reduce la fotosíntesis y el rendimiento del grano.	▲ Baja prioridad	Fungicidas sistémicos (triazoles, estrobilurinas)
Oídio	Polvo blanco en hojas	▲ Baja prioridad	Fungicida a base de azoxistrobina y propiconazol.
Fursariosis de la espiga	Espigas blanquecinas y granos deformados	▲ Baja prioridad	Fungicidas

 Posibles plagas			
Plaga	Síntomas	Prioridad	Fertilizante
Pulgón de la cebada	Hojas enrolladas y amarillas.	▲ Baja prioridad	Insecticidas. Imidacloprid o aceites vegetales.
Trips del cereal	Hojas secas y granos dañados.	▲ Baja prioridad	Insecticidas (spinosad, acetamiprid)
Pulgón de la cebada	Hojas enrolladas y amarillas.	▲ Baja prioridad	Insecticidas. Imidacloprid o aceites vegetales.
Trips del cereal	Hojas secas y granos dañados.	▲ Baja prioridad	Insecticidas (spinosad, acetamiprid)
Gusano trozador	Plántulas cortadas al nivel del suelo y reducción de la densidad de plantas.	▲ Baja prioridad	Cebos con insecticida

Imagen: Información sobre posibles plagas y enfermedades.

Y por último en el botón de sembrar, se abrirá un modal donde se piden los datos para registrar o planificar una siembra.

Aparecerá un mensaje como el de la imagen en caso de que se intente sembrar una semilla fuera del plazo recomendado.



Modal titled "Sembrar trigo" (Sow wheat) with a close button (X) in the top right corner. It contains a warning message, a seed input field, a land selection dropdown, a date input field, and a "Sembrar" button.

Sembrar trigo

⚠ El periodo recomendado para sembrar trigo es entre octubre y enero.

Semilla:

trigo

Seleccionar Tierra donde sembrar:

Seleccione una tierra

Fecha de siembra:

dd/mm/aaaa

Sembrar

Imagen: Modal sembrar tierra.

Capítulo 9: Conclusión y ampliaciones

Como se hablaba en la introducción, y como se ha podido comprobar a lo largo del proyecto, cada vez más, se puede ver cómo la tecnología tiene un impacto directo y por supuesto, positivo, en las actividades laborales diarias, hasta en los sectores más tradicionales, como puede ser la agricultura.

AgroPlan es un proyecto desarrollado por una sola persona. Esta experiencia me ha permitido entender y dimensionar mejor la carga de trabajo que implica crear una solución funcional, además de enfrentarme a muchos otros aspectos del desarrollo. Al asumir los roles de gestor de proyecto y analista, he adquirido una perspectiva valiosa que será fundamental para mi futuro profesional, especialmente al trabajar en equipo.

Una vez acabada la carrera o cerca de acabar, en las prácticas o en los primeros trabajos, el empleo será de desarrollador junior y no se tendrá una función cerca de gestionar proyecto más que la de “picar código” y realizar tareas en un principio sencillas, pero como Ingenieros que somos, llegaremos a un punto donde gestionaremos nuestros propios proyectos, y toda la formación previa como gestores, nos será muy necesaria.

AgroPlan mantiene un alcance pequeño, comparado con la solución más completa que se hubiera podido desarrollar. Reconozco que una tarea importante de un gestor de proyectos, es tener proyección del proyecto y permitirle que en un futuro, se pueda ampliar e introducir nuevas funcionalidades. En resumen, diseñar un producto para facilitar escalabilidad y adaptabilidad a futuras realidades.

9.1 Futuras ampliaciones AgroPlan:

Uno de los objetivos principales de AgroPlan como se demuestra en varios puntos de este documento, era la de construir una base de datos completa para proveer al sistema y al usuario de todas las posibles funcionalidades y consejos.

Por lo tanto, ese seguirá siendo uno de los objetivos principales para futuras ampliaciones. Así, se espera poder añadir más **información acerca de abonos y de fertilizantes**, así como de marcas y componentes, y relacionarlos con la semilla sembrada.

Otra de las funcionalidades previstas a corto plazo será la de añadir necesidad de agua por *ha*, y **control de riesgos**. Para esto se espera poder contar con un *web service* que nos proporcione información climática (litros por m² precipitados) o entradas del usuario de horas de riego, así, poder aconsejar al agricultor sobre riegos.

En base de datos hay diferentes atributos sobre alguna entidad las cuales todavía no se han utilizado en ningún desarrollo, como es por ejemplo, el tipo de agricultor. En AgroPlan se espera poder mostrar **diferentes vistas** en función de si el agricultor trabaja con “**secano**” (trigo, cebada, centeno...), “**regadío**” (lechugas, tomates, zanahorias,etc.) o por ejemplo, “**viticultor**” y poderle proveer de consejos también en el **proceso de vinificación**.

Y por último, como en otras aplicaciones del mercado, me gustaría, poder añadir una funcionalidad sobre el “**estudio de mercado**”, para que el agricultor tenga un control de cuándo y qué semillas serán más rentables para su siembra. Proporcionando estimaciones con IA de precios en el pasado y en un posible futuro, además de compararlo con los precios de la venta de semillas no tanto del producto final.

Anexo: Referencias

Ministerio de Agricultura, Pesca y Alimentación. (s.f.). *Política Agraria Común (PAC)*.

<https://www.mapa.gob.es/es/pac/>

Información del calendario

Ministerio de Agricultura, Pesca y Alimentación. (s.f.). *Calendarios de siembra, recolección y comercialización*.

<https://www.mapa.gob.es/es/estadistica/temas/estadisticas-agrarias/agricultura/calendarios-siembras-recoleccion/>

Ministerio de Agricultura, Pesca y Alimentación. (2021). *Calendario de siembra, recolección y comercialización (Informe n.º TCM30-514260)*. Ministerio de Agricultura, Pesca y Alimentación.

https://www.mapa.gob.es/es/estadistica/temas/estadisticas-agrarias/01-calendariosiembra-nuevo-sencilla-1_tcm30-514260.pdf

Datos calidades tierra

ICL Growing Solutions. (2025, abril). *Soluciones nutricionales para cultivos*.

<https://icl-growingsolutions.com/es-es/agriculture/crops/>

Estado del arte

Agroptima. (s.f.). *Software y aplicación de gestión agrícola fácil de usar*.

<https://www.agroptima.com/>

AGRIVI. (s.f.). *Software líder de gestión agrícola para agricultura digital*.

<https://www.agrivi.com/es/>

Kaampo. (s.f.). *Software de agricultura inteligente*

<https://www.kaampo.com/>

Hispattec. (2024). *Efemis: Software agrícola SIEX*.

<https://www.hispatec.com/productos/efemis/>

Farmable. (s.f.). *App de Gestión Agrícola – Farmable Core*.

<https://farmable.tech/es/farmable-core/>