



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Cultivate: desarrollo de una aplicación para gestionar
cultivos urbanos

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Montoliu Zarza, Nicolás

Tutor/a: Letelier Torres, Patricio Orlando

CURSO ACADÉMICO: 2023/2024

Resumen

Este TFG está asociado a un proyecto de emprendimiento para llevar a término una idea de negocio en el ámbito del cultivo sostenible y se ha realizado en el marco de Start.inf, el espacio de emprendimiento de la ETSINF. La idea surge de la preocupación de la población por la situación actual del medio ambiente y las maneras de contribuir positivamente a su estado. Se ha desarrollado una aplicación para ayudar a las personas con pocos conocimientos para que puedan realizar un cultivo urbano correctamente, fomentando el autoconsumo. A lo largo de esta memoria se explicarán los detalles sobre la idea de negocio y su desarrollo.

Palabras clave: medio ambiente, emprendimiento, cultivo sostenible, autoconsumo, desarrollo de aplicación móvil.

Abstract

This paper presents our entrepreneurial process to bring a business idea in the field of sustainable cultivation to fruition. The idea stems from the population's concern about the state of the environment and ways to positively contribute to its condition. We have developed an application to help people with little knowledge to correctly carry out urban cultivation, promoting self-consumption. Throughout this report, the details of the business idea and its development will be outlined.

Keywords : environment, entrepreneurship, sustainable cultivation, self-consumption.

Índice de figuras

Figura 1: Logo Cultimate	6
Figura 2: Edades de los encuestados	10
Figura 3: Género de los encuestados	10
Figura 4: Tipo de vivienda de los encuestados	10
Figura 5: Preocupación ambiental de los encuestados	11
Figura 6: Experiencia de plantado de los encuestados	11
Figura 7: Desafíos a los que enfrentan los encuestados	12
Figura 8: Disposición de pago por uso de los encuestados	12
Figura 9: Resultado trimestral acumulado	15
Figura 10: Proyección de ingresos y gastos	16
Figura 11: Lean Canvas de Cultimate [13]	18
Figura 12: Ventana "Explorar" de la aplicación Growlt	20
Figura 13: Ventana "Huerto" de la aplicación Mi Huerto Urbano	22
Figura 14: Ventana "Plant Info" de la aplicación Planter- Garden Planner	23
Figura 15: Tablero Kanban del primer sprint en la aplicación Trello	29
Figura 16: Repositorio GitHub del frontend de la aplicación	30
Figura 17: Ciclo de desarrollo de metodología agile [8]	34
Figura 18: Planificación metodológica	34
Figura 19: Diagrama de CU de la aplicación	35
Figura 20: Esquema de la arquitectura por capas	37
Figura 21: Diagrama de clases de la BBDD	38
Figura 22: Diagrama de Componentes	39
Figura 23: Índice de archivos de API	40
Figura 24: Ventana registro	43
Figura 25: Tabla de Usuarios en BBDD	43
Figura 26: Test login con PostMan	44
Figura 27: Ventana perfil de usuario	44
Figura 28: Tabla usuarios BBDD tras modificar contraseña	45
Figura 29: Tabla Favoritos BBDD tras añadir nuevos favoritos	45
Figura 30: Ventana lista de plantado con nuevos favoritos	46
Figura 31: Ventana "guía de plantado"	46
Figura 32: Tabla "plantadas" BBDD tras iniciar nuevos plantados	47
Figura 33: Plantas añadidas correctamente al huerto simulado	47
Figura 34: Código de la prueba Jest para funcionalidad "Registro"	48
Figura 35: Resultados de la prueba Jest para "Registro"	49
Figura 36: Código de la prueba Jest para "Guía de Plantado"	50
Figura 37: Resultados prueba "Guía de Plantado"	50
Figura 38: Ventana "Login"	52
Figura 39: Ventana "Registro"	53
Figura 40: Ventana "Huerto Simulado"	54
Figura 41: Ventana "Lista Plantas"	55
Figura 42: Ventana "Información de Planta"	55
Figura 43: Ventana "Foro"	56
Figura 44: Ventana "Tienda"	57
Figura 45: Ventana "Lista de Recetas"	58

Figura 46: Ventana "Información de Receta"	58
Figura 47: Ventana "Perfil de usuario"	59
Figura 48: Línea Temporal del desarrollo [12]	60
Figura 49: Contenido del sprint 1.....	61
Figura 50: Respuestas sobre funcionalidades de la aplicación	62
Figura 51: Respuestas sobre registro y seguimiento de cultivos	62
Figura 52: Respuestas sobre recomendaciones de cuidados y mantenimiento	63
Figura 53: Respuestas sobre navegación sencilla e intuitiva	63
Figura 54: Respuestas sobre información recogida en la aplicación	63
Figura 55: Respuestas sobre claridad de la información mostrada en la aplicación	64
Figura 56: Respuestas sobre diseño de la aplicación	64
Figura 57: Respuestas sobre rendimiento de la aplicación	64
Figura 58: Respuestas sobre satisfacción general al usar la aplicación	65
Figura 59: Respuestas sobre recomendación de uso a otras personas	65
Figura 60: Contenidos del sprint 2.....	66
Figura 61: Respuestas sobre velocidad de respuesta de la aplicación.....	67
Figura 62: Respuestas sobre carga inicial de la aplicación	67
Figura 63: Respuestas sobre errores durante el uso de la aplicación	68
Figura 64: Respuestas sobre estabilidad de la aplicación	68
Figura 65: Respuestas sobre navegación fluida entre ventanas de la aplicación.....	68

Índice de tablas

Tabla 1: Análisis DAFO	14
Tabla 2: Tabla comparativa de competidores	24
Tabla 3: Relación de Cultivate con los ODS.....	72

Tabla de contenidos

1. Introducción.....	6
1.1 Motivación.....	6
1.2 Objetivos.....	7
1.3 Estructura de la memoria.....	8
2. Evaluación de la idea de negocio.....	9
2.1 Posibles clientes	9



2.1.1	Resultados	9
2.1.2	Conclusiones.....	12
2.2	Análisis DAFO.....	13
2.4	Proyección de Ingresos y Gastos.....	15
2.4	Lean Canvas	17
2.4.1	¿Qué es un Lean Canvas?	17
3.	Estado del arte	20
3.1	Growlt.....	20
3.2	Mi Huerto Urbano	21
3.3	Planter: Garden Planner	22
3.4	Huerta en casa	23
3.2	Comparativa.....	24
4.	Tecnología Utilizada.....	26
4.1	React Native	26
4.2	MySQL	27
4.3	Expo	27
4.4	Android Studio.....	28
4.5	Trello	29
4.6	GitHub	30
4.7	Visual Studio Code	31
4.7.2	Características principales	31
4.8	DBeaver	32
4.9	PostMan.....	32
5.	Desarrollo de la solución.....	33
5.1	Metodología.....	33
5.1.1	¿Qué es el método Lean Startup?.....	33
5.1.2	¿Qué es la metodología ágil?.....	33
5.1.3	Metodología concreta aplicada	34
5.2	Requisitos.....	35
5.3	Diseño	37
5.3.1	Arquitectura global	37
5.3.2	Modelo de Base de Datos	38
5.3.3	Diagrama de componentes.....	39
5.4	Programación.....	40
5.4.1	Patrón API Rest	40
5.4.2	Huerto Simulado	41

5.4.3 Refactoring	42
5.5 Pruebas	42
5.5.1 Pruebas de integración	43
5.5.2 Pruebas Unitarias	48
5.5.3 Pruebas de Aceptación	51
6. Descripción de la aplicación	52
7. Cronología del TFG	60
7.1 Backlog Inicial	60
7.2 Línea temporal	60
8. Conclusiones y Trabajo Futuro	69
9. Referencias.....	70
Anexo: Relación con los ODS.....	71
Introducción a los ODS.....	71
Descripción de la aplicación.	71
Relación con los ODS.....	72



1. Introducción



Figura 1: Logo Cultimate

En este Trabajo de Fin de Grado (TFG) se presenta el desarrollo de **Cultimate**, una aplicación que proporciona a los usuarios información detallada y guiada sobre el cuidado y crecimiento de cultivos en el ámbito urbano.

También se proporciona herramientas e información adicional sobre las actividades relacionadas con los cultivos, tales como recetas de cocina o un foro con el resto de los usuarios, donde compartir ideas y experiencias.

A continuación, se presenta la motivación detrás de esta idea y la estructura y procedimientos que se han seguido para llevar a término este proyecto.

1.1 Motivación

En la actualidad existe una creciente preocupación sobre el estado del medio ambiente. Según el informe *Pew Research Center's Spring 2018 Global Attitudes Survey*[1], basado en una encuesta realizada a más de 20 000 personas de 26 países, concluía que el cambio climático era la preocupación principal de 13 países y se encontraba entre las mayores preocupaciones en el resto de ellos.

Como consecuencia, se ha visto un aumento de los movimientos sociales y culturales que abogan por estilos de vida y formas de actuar que permitan mejorar la situación en la que nos encontramos.

Con la visibilidad de la iniciativa de los Objetivos de Desarrollo Sostenible ¹ y la Agenda 2030 la población se encuentra muy concienciada sobre la conservación y cuidado de nuestro planeta y sobre los problemas actuales relacionados con el mismo.

Es por ello por lo que nos gustaría aportar nuestro grano de arena en pos de un objetivo común, centrándonos en el ámbito del cultivo urbano y el autoconsumo, actividades que han visto un incremento en su popularidad debido a que sirven tanto como forma de alimentación más saludable como de actividad lúdica.

Esta aplicación se comenzó a desarrollar en el marco de la asignatura de PIN (Proyecto de Ingeniería de Software) y se presentó una primera versión en diciembre de 2023 en la feria de proyectos de la ETSINF.

A lo largo del trabajo se cubrirán los aspectos que forman parte de un proyecto de emprendimiento: evaluación de la idea de negocio, desarrollo basado en el MVP (Minimum Viable Product) ² y la validación con los *early adopters* ³.

1.2 Objetivos

El fin principal está dirigido a la creación de cultivos urbanos, para ello se debía definir unos objetivos principales:

Los objetivos principales que se plantearon al inicio del desarrollo son los siguientes:

- Mejorar las funcionalidades de la aplicación con mejoras propuestas en las fases tempranas del desarrollo.
- Validar el producto en un entorno real, realizando experimentos con *early adopters*.

También existen algunas metas personales asociadas a este TFG:

- Profundizar en la aplicación de técnicas y metodologías estudiadas a lo largo de la carrera.
- Adquirir experiencia en el uso de nuevos lenguajes de programación asociados al desarrollo de este producto.

¹ *Objetivos de Desarrollo Sostenible*:

<https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/>

² MVP: Es la versión más simple de un producto que incluye solo las características esenciales para satisfacer a los primeros clientes y obtener retroalimentación para su mejora.

³ Los *early adopters* son los primeros usuarios que adoptan una nueva tecnología o producto antes de que se haga popular, proporcionando retroalimentación crucial para su desarrollo y mejora.

1.3 Estructura de la memoria

El capítulo 2 cubre el análisis y evaluación de la idea para determinar si el proyecto es viable.

Posteriormente en el capítulo 3 se realiza un análisis de las funcionalidades de los competidores para determinar las funcionalidades que se han de implementar en la aplicación.

En el capítulo 4 se explican las tecnologías y herramientas que se han utilizado a lo largo del proceso de desarrollo.

En el capítulo 5 se detalla todo el proceso de desarrollo y la organización de este, detallando los procedimientos y metodologías que se han llevado a cabo.

En el capítulo 6 se muestra la aplicación y se hace una breve explicación de las funcionalidades a las que se puede acceder desde cada pantalla, así como su correcta utilización.

En el capítulo 7 se hace un seguimiento cronológico del proceso de desarrollo, mostrando en profundidad cada paso realizado en el proceso de desarrollo.

Por último, en el capítulo 8 se presentan las conclusiones tras realizar el proyecto y posibles pasos de continuación con el resultado obtenido.

2. Evaluación de la idea de negocio

En este capítulo se presenta el análisis de la idea de negocio y la viabilidad que esta pueda tener, seguido de unas conclusiones extraídas de estos análisis.

2.1 Posibles clientes

Para asegurar que la idea pudiese tener éxito y para determinar el nicho de mercado, así como el interés que pudiese tener el usuario final respecto a la idea, se decidió realizar una breve encuesta.

La encuesta fue creada en una etapa temprana del desarrollo y fue enviada a conocidos y familiares de los componentes del equipo de desarrollo. En total, tras filtrar las respuestas que no eran relevantes, se obtuvieron a 50 personas como muestra estadística.

Las preguntas que conformaban la encuesta eran las siguientes:

1. Edad
2. Género
3. Tipo de vivienda
4. ¿Tiene preocupaciones relacionadas con el estado del planeta, su conservación o su cuidado?
5. ¿Posee experiencia en el cultivo de plantas?
6. ¿Cuál es el mayor desafío que enfrenta al cultivar plantas?
7. ¿Estaría dispuesto a pagar por una aplicación que le ayude con el cultivo urbano? De estar de acuerdo, ¿Cuánto estaría dispuesto a pagar?

2.1.1 Resultados

A continuación, se presentan las respuestas desglosadas pregunta a pregunta y representadas por su correspondiente gráfico:

1. Edad

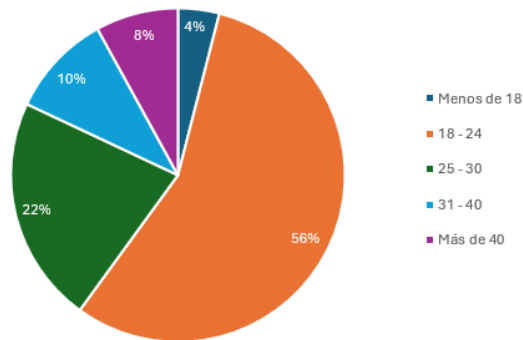


Figura 2: Edades de los encuestados

2. Género

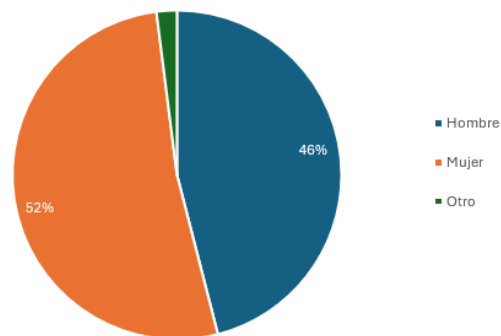


Figura 3: Género de los encuestados

3. Tipo de vivienda

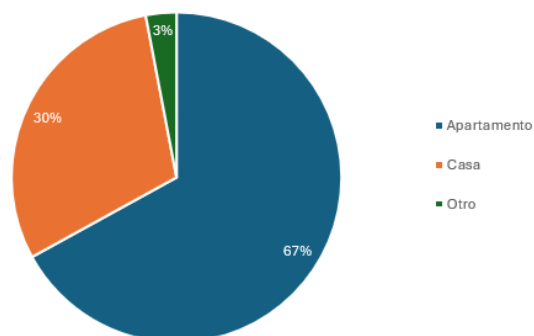


Figura 4: Tipo de vivienda de los encuestados

4. ¿Tiene preocupaciones relacionadas con el estado del planeta, su conservación o su cuidado?

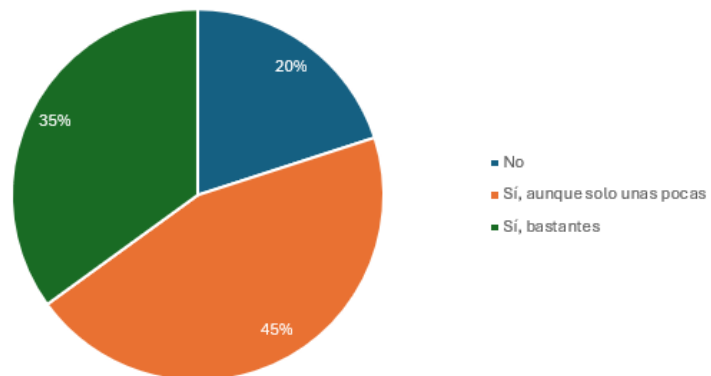


Figura 5: Preocupación ambiental de los encuestados

5. ¿Posee experiencia en el cultivo de plantas?

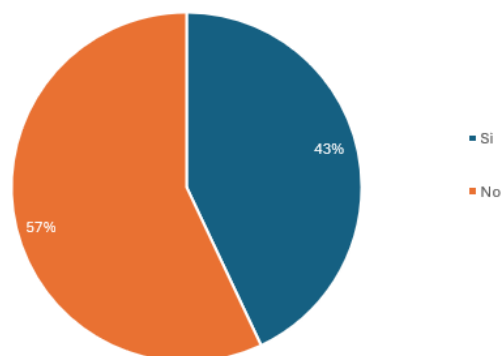


Figura 6: Experiencia de plantado de los encuestados

6. ¿Cuál es el mayor desafío que enfrenta al cultivar plantas?

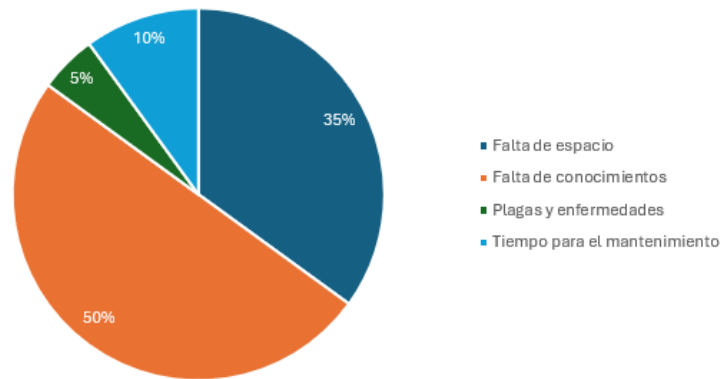


Figura 7: Desafíos a los que enfrentan los encuestados

7. ¿Estaría dispuesto a pagar por una aplicación que le ayude con el cultivo urbano? De estar de acuerdo, ¿Cuánto estaría dispuesto a pagar?

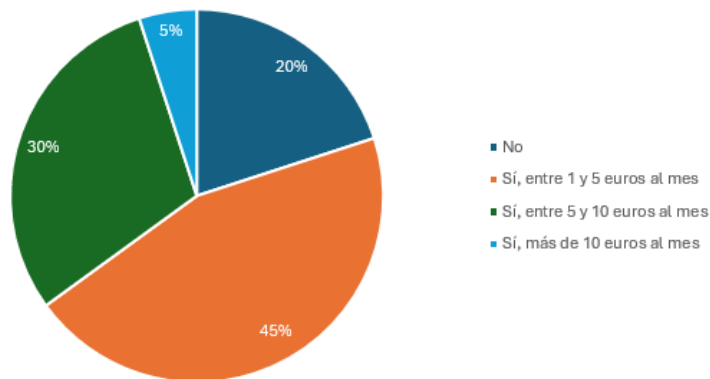


Figura 8: Disposición de pago por uso de los encuestados

2.1.2 Conclusiones

Tras analizar las respuestas obtenidas (figuras 2 a 8), se pudo obtener un conjunto de premisas de la muestra estadística que dan a entender una serie de características y opiniones comunes, que se describirán a continuación, a fin de obtener el perfil de los usuarios.

Se puede describir la muestra estadística con las siguientes afirmaciones:

- La edad mayoritaria se encuentra entre los 18 y los 30 años.
- El género que predomina entre los encuestados es el femenino.
- El tipo de vivienda más común es el apartamento

Respecto a las preguntas referentes al contenido del proyecto se puede extraer:

- La mayoría de los encuestados no posee experiencia en el cultivo de plantas
- Su mayor desafío es la falta de conocimiento sobre los procesos de cultivo
- Una gran parte de los encuestados tiene algún tipo de preocupación respecto al estado del medio ambiente.

Tras este análisis se puede expresar que el conjunto encuestado muestra una tendencia positiva a la idea principal de la aplicación. Por ello que el perfil de usuario que se busca sería el siguiente:

Personas que tienen interés en la jardinería doméstica y están dispuestas a cultivar sus propios alimentos para consumo personal. El perfil acoge a principiantes que desean aprender y profundizar en la materia. Además, se dirige a aquellos que enfrentan limitaciones de espacio, como residentes urbanos, así como a aquellos comprometidos con prácticas sostenibles.

Este perfil surge de nuestra idea principal (proporcionar ayuda a las personas que deseen introducirse en el cultivo urbano y que no posean experiencia) en combinación con las conclusiones extraídas de la encuesta.

2.2 Análisis DAFO

¿Qué es un análisis DAFO?

El análisis DAFO[2] es una herramienta estratégica utilizada para identificar y evaluar los factores internos (Debilidades y Fortalezas) y externos (Amenazas y Oportunidades) que pueden afectar el éxito de un proyecto, organización o iniciativa. Este análisis se suele representar gráficamente mediante una matriz cuadrada.

El objetivo es proporcionar una visión clara de la situación actual y ayudar a formular estrategias para maximizar las fortalezas y oportunidades mientras se minimizan las debilidades y amenazas. Se ha utilizado una herramienta para elaborar el análisis DAFO online ⁴.

⁴ Ministerio de Industria y Turismo, gobierno de España. *Herramienta DAFO* [En línea]. Disponible en: <https://dafo.ipyme.org/Home>



En la tabla 1, se pueden observar los resultados del análisis DAFO:

<p>DEBILIDADES</p> <ul style="list-style-type: none"> • Poca experiencia y conocimiento técnico de las herramientas utilizadas • Recursos limitados en comparación con los competidores ya establecidos • La falta de reconocimiento de marca puede afectar la confianza de los usuarios 	<p>AMENAZAS</p> <ul style="list-style-type: none"> • Entrada a un mercado con competidores ya establecidos • Cambios legislativos respecto a los cultivos urbanos • Necesidad de diferenciación con los competidores
<p>FORTALEZAS</p> <ul style="list-style-type: none"> • Experiencia previa en trabajos de desarrollo • Flexibilidad para mejoras tras recibir feedback • Innovación en características solicitadas por los usuarios 	<p>OPORTUNIDADES</p> <ul style="list-style-type: none"> • Ofrecer funcionalidades que no ofrece la competencia • Sector de mercado con pocos competidores

Tabla 1: Análisis DAFO

Como resultado del análisis se puede concluir que la aplicación posee una ventaja respecto a los competidores del sector, ya que se centralizan las funcionalidades que los usuarios de otras aplicaciones similares demandan.

Por el contrario, eso no elimina el hecho de que existen amenazas que podrían poner en riesgo esa ventaja. Es por ello por lo que el objetivo principal era crear una aplicación atractiva para destacar entre la competencia y atraer al mayor número de potenciales usuarios posible.

2.4 Proyección de Ingresos y Gastos

La proyección de ingresos y gastos constituye un análisis cuantitativo que evalúa la factibilidad económica del proyecto. Esto permite estimar de manera aproximada la evolución monetaria del proyecto durante un período específico.

Se debe tener en cuenta que esta proyección de ingresos y gastos está orientada a 3 años, dividido por trimestres. El modelo de negocio de Cultimate se basa en una suscripción de 3€ mensuales con publicidad o una suscripción de 5€ mensuales sin publicidad.

En la figura 10 se muestran las proyecciones de gastos, ingresos y crecimiento.

La previsión de inversión inicial asciende a 13 220€, alcanzando el punto de equilibrio tras el tercer trimestre del primer año y comenzando a tener beneficios tras el cuarto trimestre del primer año.

En la figura 9 se puede observar una gráfica que recoge los resultados trimestrales acumulados.

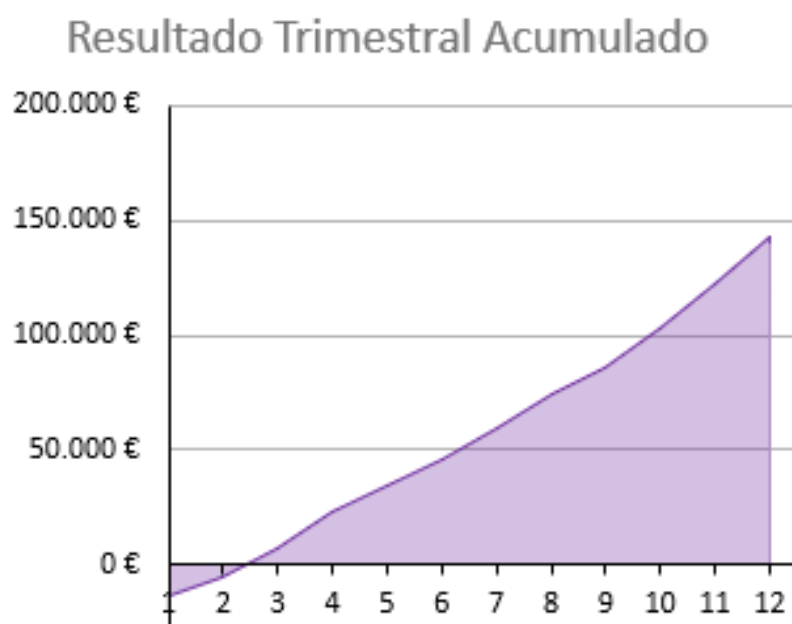


Figura 9: Resultado trimestral acumulado

Tipos de licencias acumuladas	Número de Trimestre											
	1	2	3	4	5	6	7	8	9	10	11	12
Suscripciones nuevas vendidas sin publicidad	200	212	225	238	252	268	284	301	319	338	358	380
Suscripciones renovadas después de tres meses	0	120	150	160	170	150	160	170	200	205	210	210
Suscripciones nuevas vendidas con publicidad	220	232	245	256	260	265	274	281	296	300	306	309
Suscripciones renovadas después de tres meses con publicidad	0	100	130	140	152	164	172	181	185	192	194	198
Total de licencias nuevas vendidas y renovadas	420	664	750	794	834	847	890	933	1000	1035	1068	1097
Ingresos Trimestrales												
ingresos suscripciones	4.980 €	7.968 €	8.996 €	9.537 €	10.045 €	10.126 €	10.670 €	11.219 €	12.111 €	12.571 €	13.023 €	13.408 €
ingresos publicidad	19.800 €	29.880 €	33.750 €	35.640 €	37.080 €	38.610 €	40.140 €	41.580 €	43.290 €	44.280 €	45.000 €	45.630 €
Total Ingresos	24.780 €	37.848 €	42.746 €	45.177 €	47.125 €	48.736 €	50.810 €	52.799 €	55.401 €	56.851 €	58.023 €	59.038 €
Gastos Anuales												
Infraestructura Cloud	400 €	400 €	400 €	400 €	400 €	400 €	400 €	400 €	600 €	600 €	600 €	600 €
Ordenadores e impresoras	4.000 €	0 €	0 €	0 €	2.000 €	0 €	0 €	0 €	2.000 €	0 €	0 €	0 €
Muebles oficina e instalaciones	2.000 €	0 €	0 €	0 €	1.000 €	0 €	0 €	0 €	1.000 €	0 €	0 €	0 €
Marketing	2.000 €	0 €	0 €	0 €	2.000 €	0 €	0 €	0 €	2.500 €	0 €	0 €	0 €
Alquiler oficina	3.000 €	3.000 €	3.000 €	3.000 €	3.000 €	3.000 €	3.000 €	3.000 €	3.500 €	3.500 €	3.500 €	3.500 €
Internet, electricidad, agua, teléfono, etc.	1.500 €	1.500 €	1.500 €	1.500 €	1.500 €	1.500 €	1.500 €	1.500 €	2.000 €	2.000 €	2.000 €	2.000 €
Gestoría	600 €	600 €	600 €	600 €	600 €	600 €	600 €	600 €	800 €	800 €	800 €	800 €
CEO - Director Ejecutivo	5.000 €	5.000 €	5.000 €	5.000 €	5.000 €	5.000 €	5.000 €	5.000 €	5.000 €	5.000 €	5.000 €	5.000 €
CTO - Director Técnico	5.000 €	5.000 €	5.000 €	5.000 €	5.000 €	5.000 €	5.000 €	5.000 €	5.000 €	5.000 €	5.000 €	5.000 €
Desarrolladores Junior	7.500 €	7.500 €	7.500 €	7.500 €	7.500 €	15.000 €	15.000 €	15.000 €	15.000 €	15.000 €	15.000 €	15.000 €
Técnicos de soporte	7.000 €	7.000 €	7.000 €	7.000 €	7.000 €	7.000 €	7.000 €	7.000 €	7.000 €	7.000 €	7.000 €	7.000 €
Total Gastos	38.000 €	30.000 €	30.000 €	30.000 €	35.000 €	37.500 €	37.500 €	37.700 €	44.400 €	38.900 €	38.900 €	38.900 €
Resultado Trimestral	-13.220 €	7.848 €	12.746 €	15.177 €	12.125 €	11.236 €	13.310 €	15.099 €	11.001 €	17.951 €	19.123 €	20.138 €
Resultado Trimestral Acumulado	-13.220 €	-5.372 €	7.374 €	22.551 €	34.676 €	45.912 €	59.222 €	74.320 €	85.321 €	103.272 €	122.395 €	142.533 €

Figura 10: Proyección de ingresos y gastos

2.4 Lean Canvas

2.4.1 ¿Qué es un Lean Canvas?

Un Lean Canvas[6] es una herramienta que permite desarrollar y comunicar de una forma sencilla y visual un modelo de negocio de manera eficiente y rápida. Este representa en una sola página los aspectos clave que son críticos para el desarrollo correcto del proyecto de emprendimiento.

Está compuesto por nueve bloques que permiten describir y diseñar de manera ágil un modelo de negocio. Estos bloques son:

1. Problema: Describe los principales problemas que el producto o servicio busca resolver.
2. Segmentos de clientes: Identifica quiénes son los usuarios o clientes objetivo del producto o servicio.
3. Propuesta de valor única: Articula qué hace que el producto o servicio sea único y atractivo para los clientes.
4. Solución: Presenta la solución que se ofrecerá para resolver los problemas identificados.
5. Canales: Define cómo se llegará y se entregará el producto o servicio a los clientes.
6. Flujos de ingresos: Describe cómo la startup generará ingresos a partir de su propuesta de valor.
7. Estructura de costos: Enumera los principales costos asociados con la operación del negocio.
8. Métricas clave: Identifica las métricas que se utilizarán para medir el éxito y el progreso del negocio.
9. Ventaja injusta: Menciona cualquier ventaja competitiva que sea difícil de replicar por otros.



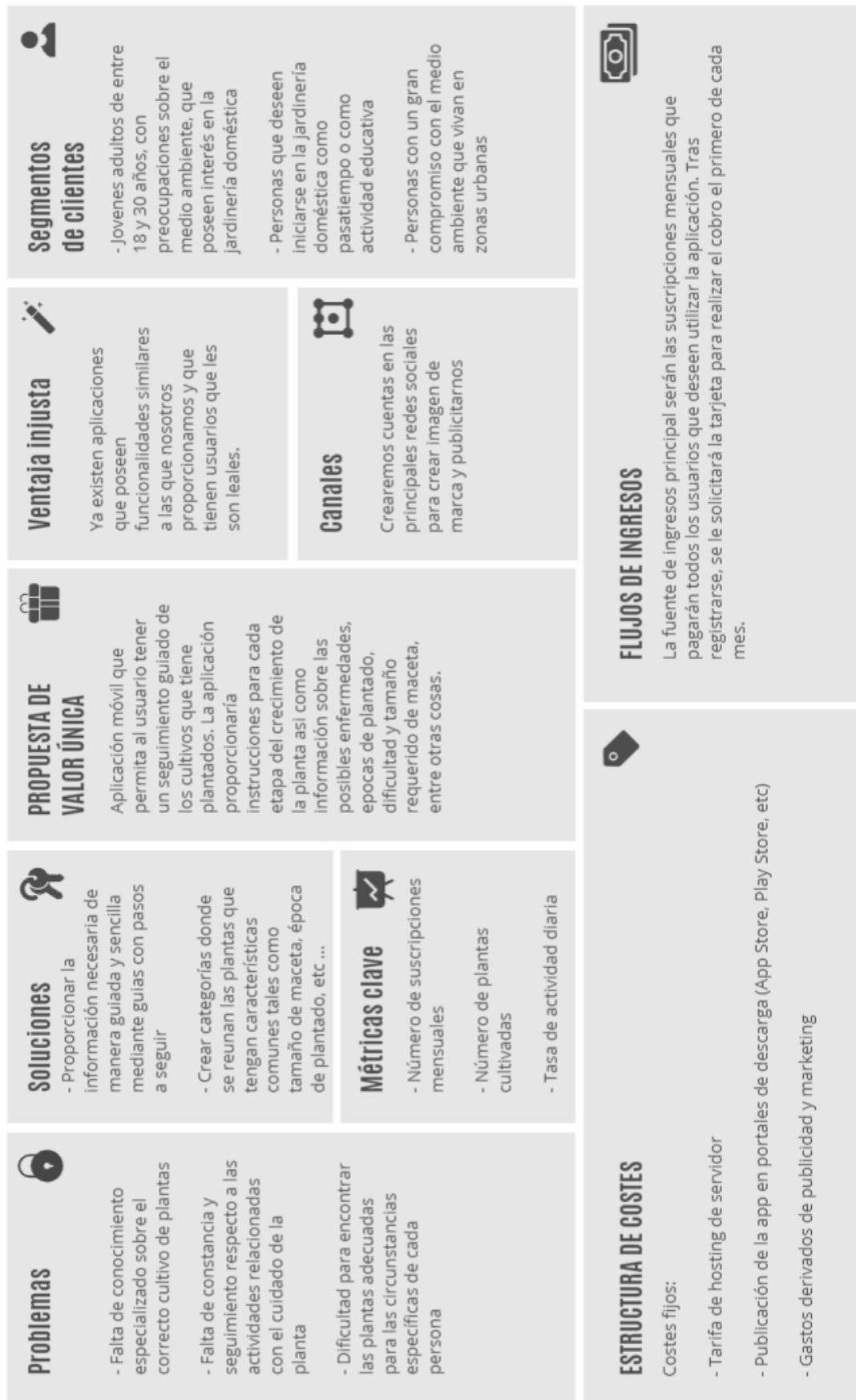


Figura 11: Lean Canvas de Cultimate [13]

Como se explica en la figura 11, la aplicación cuenta con una serie de funcionalidades innovadoras y otras que ya están presentes en otras aplicaciones de índole similar. A pesar de ello, al unificar bajo una misma aplicación todas esas funcionalidades que los usuarios valoran positivamente, proporciona a la aplicación una cierta diferenciación respecto al resto de competidores. Sumado al hecho de que se posee un segmento de clientes amplio, permite orientar de manera correcta el diseño del producto.

De este análisis se puede concluir que la idea de negocio puede funcionar bien y puede llegar a dar beneficios si se implementa de forma correcta. También mencionar que esta propuesta tiene funcionalidades de las que carecen los competidores, lo cual puede ser un gran atractivo para los futuros usuarios.

3. Estado del arte

En este capítulo se verá la situación actual del mercado orientado al cultivo urbano. Tras ello, se realizará una comparativa para ver qué cosas tienen en común y que diferencia a Cultivate del resto de competidores.

3.1 Growlt



Growlt ⁵ es una aplicación móvil creada por la compañía Glority Global Group Ltd. Esta aplicación proporciona numerosas funcionalidades que se comentarán a continuación.



Figura 12: Ventana "Explorar" de la aplicación Growlt

⁵ Growlt: <https://play.google.com/store/apps/details?id=com.growit.ai&hl=es>

Características distintivas

- Permite diagnosticar enfermedades de la planta mediante la cámara del dispositivo

Modelo de negocio

La aplicación sigue un modelo de pago por suscripción mensual. Tras su instalación, se le solicita al usuario la creación de una cuenta para poder identificarse. Hecho esto, se le solicita ingresar los datos para el pago de una cuota de 3 euros que se renovará de forma mensual.

Tras pagar la cuota, todas las funcionalidades que se proporcionan son accesibles por el usuario, sin necesidad de pagos adicionales.

También he de mencionar que no existe ningún tipo de anuncios ni publicidad de ningún tipo.

3.2 Mi Huerto Urbano



Mi Huerto Urbano ⁶ es una aplicación desarrollada por el autor antodrag. Proporciona información y ayuda a la hora de cultivar.

Características distintivas

- Dispone de soporte para múltiples lenguajes: castellano, inglés, valenciano, francés.

Modelo de negocio

La aplicación es de uso gratuito y monetiza mediante publicidad dentro de la misma. También dispone de diferentes packs de pago único a fin de eliminar la publicidad y para desbloquear algunas funcionalidades o información que no está disponible dentro de la versión gratuita.

⁶ Mi Huerto Urbano:

<https://play.google.com/store/apps/details?id=montar.huerto.urbano&hl=es>

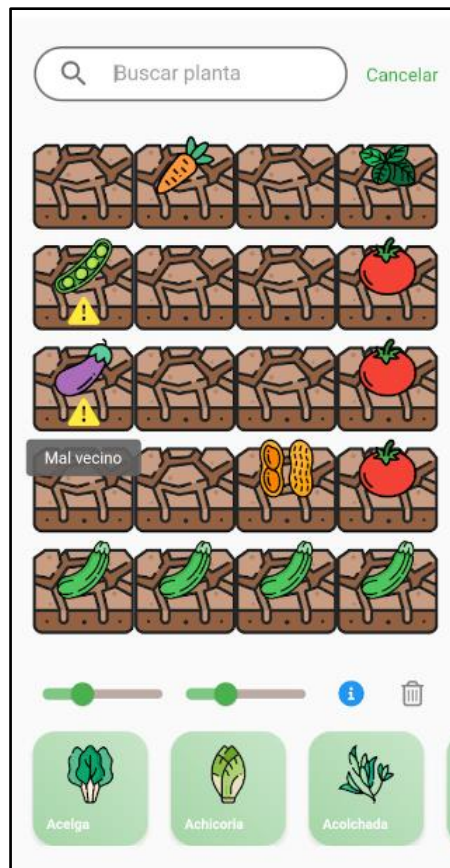


Figura 13: Ventana "Huerto" de la aplicación Mi Huerto Urbano

3.3 Planter: Garden Planner



Planter: Garden Planner ⁷ es una aplicación desarrollada por el autor Planter. Posee una interfaz estilizada e información detallada sobre las plantas cultivadas que son más comunes.

Características distintivas

- No se ha encontrado características que la distinguen de las otras aplicaciones que se mencionan.

Modelo de negocio

La aplicación posee un modelo de negocio de pago único. Tras instalar y crear la cuenta de usuario, la aplicación solicita el pago único de 45 euros para desbloquear todas las funcionalidades que proporciona la aplicación.

No posee ningún tipo de paquete adicional ni publicidad.

⁷ Planter: Garden Planner :

<https://play.google.com/store/apps/details?id=com.perculacreative.peter.gardenplanner&hl=es>

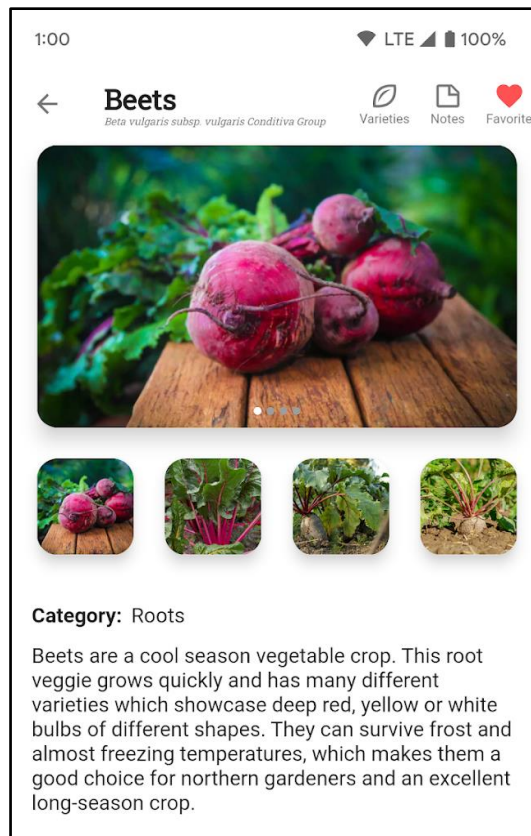


Figura 14: Ventana "Plant Info" de la aplicación Planter- Garden Planner

3.4 Huerta en casa



Huerta en casa ⁸ es una aplicación que está desarrollada por ADL Desarrollos.

Características distintivas

- Seguimiento del estado de la planta en todos los pasos del desarrollo

Modelo de negocio

No ha sido posible obtener los datos que hacen referencia al modelo de negocio que tiene la aplicación.

⁸ Huerta en casa:

<https://play.google.com/store/apps/details?id=com.adldesarrollos.huerta&hl=es>

3.2 Comparativa

En este apartado se muestran las características y funcionalidades principales (usando ✓ para “Sí”, X para “No” o ? si no se ha encontrado información) de las diferentes aplicaciones. Se marcará las funcionalidades que posee cada aplicación a fin de comparar de forma sencilla que tan diferenciada es nuestra aplicación respecto a la competencia del sector.

Nombre	Growlt	Mi Huerto Urbano	Planter: Garden Planner	Huerta en casa	<u>Cultimate</u>
<i>Guía de plantado</i>	✓	✓	✓	✓	✓
<i>Diagnóstico mediante cámara</i>	✓	X	X	X	X
<i>Consejos</i>	✓	X	✓	✓	✓
<i>Sugerencias de plantado</i>	✓	✓	✓	✓	✓
<i>Calendario de plantado</i>	✓	✓	✓	✓	✓
<i>Información sobre crecimiento</i>	✓	✓	✓	✓	✓
<i>Tienda integrada</i>	X	X	X	X	✓
<i>Multilinguaje</i>	Inglés, castellano	5	sólo inglés	X	sólo castellano
<i>Facilidad de uso</i>	✓	✓	X	✓	✓
<i>Huerto simulado</i>	X	✓	✓	✓	✓
<i>Foro</i>	X	X	X	X	✓
<i>Anuncios</i>	X	✓	X	X	✓
<i>Modelo de negocio</i>	suscripción mensual	gratuita	versión de pago	?	Suscripción mensual
<i>Coste</i>	2,5€ / mes	0€	45€ pago único	?	3€/mes o 5€/mes
<i>Seguimiento de planta</i>	✓	X	X	✓	✓
<i>Función de favoritos</i>	✓	X	X	✓	✓
<i>Perfil de usuario</i>	X	X	✓	✓	✓
<i>FAQ</i>	X	X	X	✓	✓

Tabla 2: Tabla comparativa de competidores

De la comparativa vista en la tabla 2 se puede extraer que *Cultivate* cuenta con algunas características que la distinguen del resto. Entre ellas se encuentra la existencia de un **foro** con múltiples canales, diferenciados por el cultivo del que tratan, donde todos los usuarios pueden hacer preguntas y responder. También posee una **tienda integrada** donde se encuentran las herramientas y materiales necesarios para comenzar a cultivar.

A su vez, la aplicación también integra funcionalidades deseadas por los usuarios tales como el **huerto simulado**, donde se recogen todas las plantas que se encuentran siendo cultivadas de una forma visual, o el **seguimiento de la planta** donde se presenta la información y cuidados necesarios para cada etapa del desarrollo de la planta.

4. Tecnología Utilizada

En este capítulo, se comentarán todas las tecnologías y herramientas que se han utilizado a lo largo del desarrollo del proyecto. Estas tecnologías son las que en su momento se pensó que eran las que mejor se adaptaban a las necesidades del equipo, lo cual no implica que no puedan realizarse proyectos de índole similar con tecnologías alternativas.

4.1 React Native

React Native ⁹ es un framework de desarrollo de aplicaciones móviles de código abierto creado por Facebook. Su principal característica es que permite a los desarrolladores construir aplicaciones móviles utilizando JavaScript y React.

Características principales

Las características que hacen React Native un framework tan popular son las siguientes[3]:

- Permite el **desarrollo multiplataforma** sin tener que escribir código específico para alguna plataforma en concreto. Un único código para todas las plataformas.
- Utiliza **componentes nativos** en lugar de vistas web, lo que mejora notablemente el rendimiento y ofrece al usuario final una experiencia más fluida.
- Introduce el **Hot reloading**: El Hot reloading permite ver los cambios hechos al código en tiempo real, sin tener que recompilar la aplicación completa.
- Tiene una gran **modularidad** y **extensibilidad**, lo que permite a los desarrolladores crear módulos para actividades específicas e integrarlas con la aplicación. Estando disponible a su vez la importación de librerías de terceros.
- Posee una **gran comunidad**, lo que proporciona numerosas herramientas de soporte y librerías, haciendo el proceso de desarrollo algo más sencillo.

Se escogió esta tecnología debido a que es de las más populares en los entornos de desarrollo móvil y el equipo valoró positivamente entrar en contacto y adquirir experiencia con ella.

⁹ React Native: <https://reactnative.dev>

4.2 MySQL

MySQL ¹⁰ es un sistema de gestión de bases de datos desarrollado bajo una licencia de código abierto y comercial. Es una alternativa que ofrece una sintaxis bastante sencilla y fácil de comprender, razón por la cual fue elegida.

Utiliza el modelo lógico relacional que se basa en el concepto de relaciones como método para enlazar diferentes datos que puedan estar conectados. Este modelo lógico es bastante fácil de comprender, permitiendo que la introducción de los modelos entidad-relación sea bastante sencilla.

Se escogió MySQL puesto que era la tecnología de BBDD de modelo relacional que el equipo conocía y en la que se había formado a lo largo de las diferentes asignaturas de la carrera.

4.3 Expo

Expo ¹¹ es una plataforma y un conjunto de herramientas que facilitan el desarrollo de aplicaciones móviles utilizando React Native. Está diseñada para simplificar y acelerar el proceso de desarrollo, proporcionando una serie de características y servicios que ayudan a los desarrolladores a construir, desplegar y gestionar aplicaciones móviles.

Características

Las características más destacadas de expo son [4]:

- Acceso a Funcionalidades Nativas Sin Escribir Código Nativo: El SDK de Expo proporciona APIs listas para usar que permiten acceder a funcionalidades del dispositivo (cámara, geolocalización, notificaciones, etc.) sin necesidad de escribir código nativo.
- Desarrollo Multiplataforma: Puedes desarrollar aplicaciones que funcionen tanto en iOS como en Android con una única base de código en JavaScript/TypeScript, siguiendo el principio de desarrollo multiplataforma de React Native.
- Actualizaciones Over-the-Air (OTA): Expo te permite enviar actualizaciones a tus aplicaciones directamente a los dispositivos de los usuarios sin pasar por el proceso de revisión de la App Store o Google Play.

¹⁰ MySQL: <https://www.mysql.com>

¹¹ Expo: <https://expo.dev>



Culminate: desarrollo de una aplicación para gestionar cultivos urbanos

- **Despliegue y Construcción Sencillos:** Con Expo, puedes construir y distribuir aplicaciones fácilmente utilizando servicios en la nube como EAS (Expo Application Services), evitando la necesidad de configurar entornos de construcción locales complejos.
- **Probar Aplicaciones en Dispositivos Físicos:** Expo Go te permite probar aplicaciones en dispositivos reales de manera rápida y sencilla escaneando un código QR, sin necesidad de compilar y desplegar cada vez.
- **Soporte para Plugins de Terceros:** Aunque Expo proporciona una amplia gama de APIs nativas, puedes "expulsar" (eject) tu aplicación para usar librerías nativas específicas que no están incluidas en el SDK de Expo.
- **Entorno de Desarrollo Intuitivo:** Expo CLI y Expo DevTools proporcionan un entorno de desarrollo intuitivo y fácil de usar, con herramientas que simplifican la creación, ejecución y depuración de aplicaciones móviles.

Esta tecnología fue utilizada debido a que es la plataforma más popular para el despliegue y desarrollo de aplicaciones en el entorno de React Native.

4.4 Android Studio

Android Studio ¹² es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) oficial creado por Google para el desarrollo de aplicaciones Android. Proporciona a los desarrolladores todas las herramientas necesarias para diseñar, desarrollar, depurar y desplegar aplicaciones móviles para dispositivos que ejecutan el sistema operativo Android.

Android Studio también dispone de emuladores Android integrados. Estos emuladores son altamente personalizables, pudiendo elegir el modelo de dispositivo Android que deseas emular y también que distribución del sistema operativo deseas instalarle.

Este emulador junto con expo, fueron de mucha ayuda para poder depurar la aplicación a medida que se iba desarrollando. Siendo extremadamente cómoda y sencilla de utilizar a la par que útil y eficiente.

Se utilizó esta aplicación puesto que proporciona un emulador de Android completamente configurable para probar la aplicación en entornos móviles, puesto que no se consiguió encontrar un emulador de iOS funcional fuera de los ordenadores Mac, los cuales no poseían los miembros del equipo de desarrollo.

¹² Android Studio: <https://developer.android.com/studio?hl=es-419>

4.5 Trello

Trello ¹³ es una aplicación de gestión de proyectos basada en web y móvil que utiliza tableros, listas y tarjetas para organizar y priorizar tareas. Es una herramienta muy utilizada para la gestión de proyectos, tareas personales, planificación, etc.

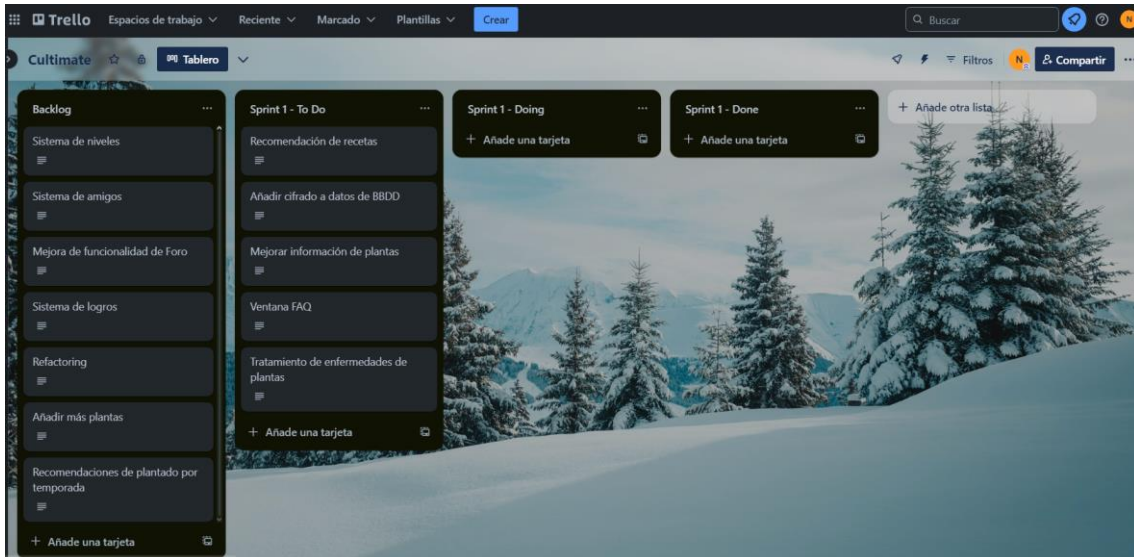


Figura 15: Tablero Kanban del primer sprint en la aplicación Trello

Como se puede observar en la figura 15, la herramienta permite crear diferentes listas y tarjetas. Estas tarjetas pueden desplazarse con el ratón a cualquiera de las otras listas y pueden contener información de todo tipo.

Es muy útil para crear tableros de desarrollo Kanban, los cuales se basan en la visualización del flujo de trabajo a través de un tablero que contiene columnas representando diferentes etapas del proceso y tarjetas que representan las tareas individuales. Kanban promueve la limitación del trabajo en curso, lo que significa establecer un límite máximo de tareas que pueden estar en progreso en cada etapa del proceso. Esto ayuda a evitar la sobrecarga del equipo y fomenta la finalización de tareas antes de comenzar nuevas.

Se escogió esta herramienta por la simpleza visual y la facilidad de uso que esta proporciona.

¹³ Trello: <https://trello.com/es>

4.6 GitHub

GitHub ¹⁴ es una plataforma en línea para alojar y gestionar proyectos de desarrollo de software utilizando el sistema de control de versiones Git. Permite a los desarrolladores almacenar y compartir código, colaborar en proyectos, revisar y gestionar cambios, realizar un seguimiento del historial de versiones y trabajar de manera conjunta en proyectos, ya sea de forma pública o privada.

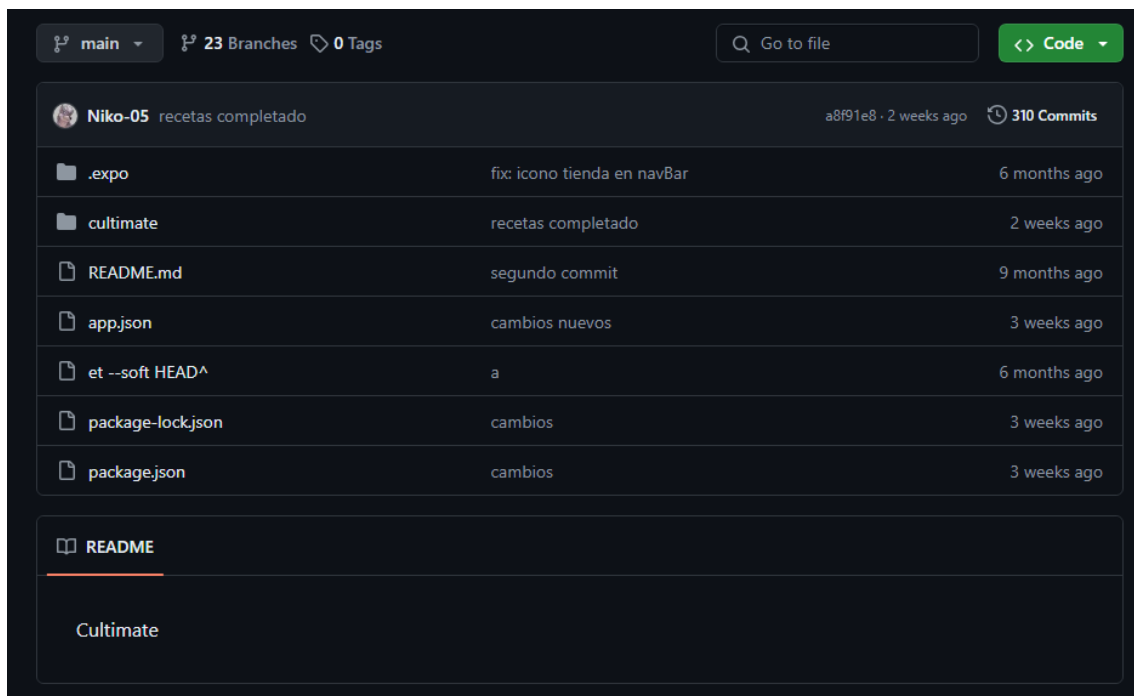


Figura 16: Repositorio GitHub del frontend de la aplicación

Entre todas las opciones que ofrecen gestión de versiones en Git, se eligió GitHub por ser la más conocida y con la que el equipo de desarrollo poseía más experiencia. Además de su facilidad de uso y de ser gratis. En la figura 16 se muestra el repositorio del proyecto.

¹⁴ GitHub: <https://github.com>

4.7 Visual Studio Code

Visual Studio Code (VS Code) ¹⁵ es un editor de código fuente desarrollado por Microsoft. Es una aplicación de código abierto y multiplataforma que ofrece una amplia gama de características diseñadas para mejorar la productividad de los desarrolladores. Además de su potente conjunto de características, Visual Studio Code es conocido por su velocidad y ligereza, lo que lo hace ideal para proyectos de desarrollo de software de todos los tamaños.

4.7.2 Características principales

Las características principales de Visual Studio Code son [5]:

- **Editor de Código Fuente:** Visual Studio Code ofrece un potente editor de código con resaltado de sintaxis, autocompletado inteligente y muchas otras características para facilitar la escritura y edición de código.
- **Soporte Multiplataforma:** Es compatible con Windows, macOS y Linux, lo que permite a los desarrolladores trabajar en diferentes sistemas operativos.
- **Extensiones Personalizables:** Permite la instalación de una amplia variedad de extensiones para personalizar y mejorar la funcionalidad del editor según las necesidades específicas del usuario.
- **Integración con Control de Versiones:** Ofrece integración con sistemas de control de versiones como Git, lo que facilita el seguimiento de cambios en el código y la colaboración en proyectos de desarrollo de software.
- **Depuración Integrada:** Proporciona herramientas de depuración integradas que permiten a los desarrolladores encontrar y corregir errores en su código de manera eficiente.
- **Terminal Integrada:** Incluye una terminal integrada que permite a los desarrolladores ejecutar comandos y scripts directamente desde el editor, lo que mejora el flujo de trabajo.

La elección sobre este entorno de desarrollo fue meramente preferencia personal, ya que esta elección no afecta excesivamente al desarrollo del proyecto.

¹⁵ Visual Studio Code: <https://code.visualstudio.com>



4.8 DBeaver

DBeaver ¹⁶ es una herramienta de administración y desarrollo de bases de datos de código abierto que proporciona una interfaz gráfica para trabajar con una variedad de sistemas de gestión de bases de datos. Permite a los usuarios conectarse, explorar, consultar, editar y administrar bases de datos de manera eficiente.

Esta tecnología posibilita la administración visual de tablas o, en el caso de usuarios experimentados, la ejecución de consultas SQL. También facilita la ejecución de archivos en lenguaje SQL para llevar a cabo operaciones por lotes.

Se escogió esta herramienta puesto que es gratuita y de código abierto.

4.9 PostMan

Postman ¹⁷ es una aplicación que permite a los desarrolladores construir, probar, documentar y compartir API. Se puede utilizar como una aplicación de escritorio o como una extensión del navegador. Postman facilita la creación de solicitudes HTTP y la visualización de respuestas de manera estructurada, lo que ayuda a los desarrolladores a trabajar con API de forma más efectiva.

Esta herramienta ha permitido hacer numerosas pruebas a la API Rest, pudiendo así confirmar que los métodos que posee la aplicación funcionan correctamente y de acuerdo con los parámetros que se le proporcionan.

La elección de esta herramienta vino dada por la popularidad que esta tenía respecto a las otras opciones del mercado y el hecho de que también fuese gratuita y de código abierto.

¹⁶ DBeaver: <https://dbeaver.io>

¹⁷ PostMan: <https://www.postman.com>

5. Desarrollo de la solución

En este capítulo se detalla la metodología que se ha seguido a lo largo de todo el desarrollo del proyecto, así como la estructura del código, las pruebas realizadas y las anécdotas más destacables del desarrollo.

5.1 Metodología

5.1.1 ¿Qué es el método Lean Startup?

El método Lean Startup es una metodología para desarrollar productos que busca acortar los ciclos de desarrollo, medir el progreso y obtener retroalimentación del cliente para validar y ajustar el producto rápidamente [6]. Dos de los componentes clave del método Lean Startup son el Producto Mínimo Viable (MVP) y los experimentos.

El MVP es la versión más simple de un producto que permite recolectar la máxima cantidad de retroalimentación de los clientes con el menor esfuerzo. El objetivo del MVP no es ofrecer un producto completamente desarrollado, sino una versión que pueda ser usada por los primeros usuarios para validar las funcionalidades ideadas para el producto.

Los experimentos son pruebas controladas diseñadas para validar ideas sobre algunos aspectos de un producto. La finalidad de estos experimentos es reducir la incertidumbre y el riesgo al obtener retroalimentación directa de los usuarios.

5.1.2 ¿Qué es la metodología ágil?

La metodología ágil es un enfoque de gestión de proyectos y desarrollo de software que se centra en la flexibilidad, la colaboración y la entrega continua de valor al cliente. Su objetivo es adaptarse rápidamente a los cambios y mejorar continuamente los procesos y productos.

Los principios y componentes de la metodología ágil están basados en el Manifiesto Ágil, creado en 2001 por un grupo de desarrolladores de software [7]. Algunos elementos de los métodos ágiles que se han utilizado son:

- *Sprint*: Periodo de tiempo fijo (de 1 a 4 semanas) durante el cual el equipo trabaja para completar un conjunto de tareas del backlog.
- *Backlog*: Lista priorizada de todas las tareas, características y requisitos del producto.



- *Product Owner*: Es el responsable de asegurar que el equipo de desarrollo produzca el máximo valor posible con su trabajo durante el sprint.



Figura 17: Ciclo de desarrollo de metodología ágil [8]

En la figura 17, se puede observar el proceso que se ha de llevar a cabo durante la aplicación de la metodología ágil. Estos pasos se siguen en orden, repitiéndose los pasos 2 a 6 durante la duración de un sprint.

5.1.3 Metodología concreta aplicada

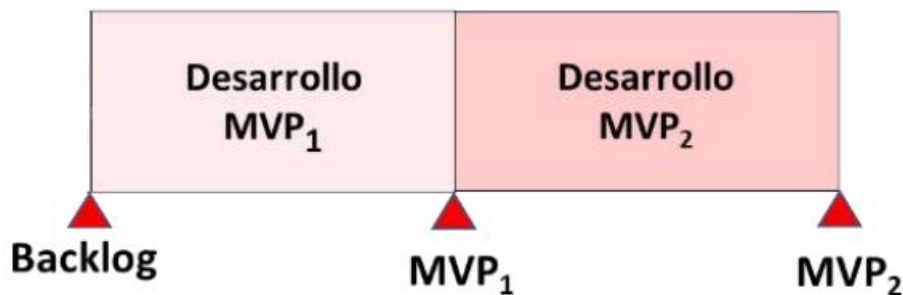


Figura 18: Planificación metodológica

Durante el desarrollo del proyecto se utilizó la metodología ágil siguiendo el esquema de la figura 18. Este esquema consiste en realizar 2 sprints durante los cuales se ha trabajado para producir un MVP (Mínimo Producto Viable) al final de cada sprint. Al finalizar el desarrollo de cada uno de esos MVP, se realizó un experimento con unos pocos *early adopters*, a fin de obtener opiniones y comentarios sobre el sistema de cara a futuros MVP.

La elección del método Lean Startup y las metodologías ágiles vino dada por varios factores. El primero fue porque el equipo ya poseía experiencia trabajando con estas metodologías, ya que a lo largo de su formación tuvieron dos asignaturas que hacían uso de estas para llevar a término proyectos de desarrollo.

La otra fue que estas metodologías son sencillas de comprender y aplicar, sin ningún coste real añadido y que facilitan mucho la detección de problemas en cualquiera de las fases de desarrollo y su posterior corrección, contribuyendo a un flujo de trabajo más natural y eficiente.

5.2 Requisitos

Para ilustrar las principales funcionalidades de Cultimate, en la figura 19 se presenta un diagrama de CU. Aunque este tipo de diagrama no es parte de Lean Startup o de los métodos ágiles, nos resulta útil para tener una representación visual de los requisitos funcionales.

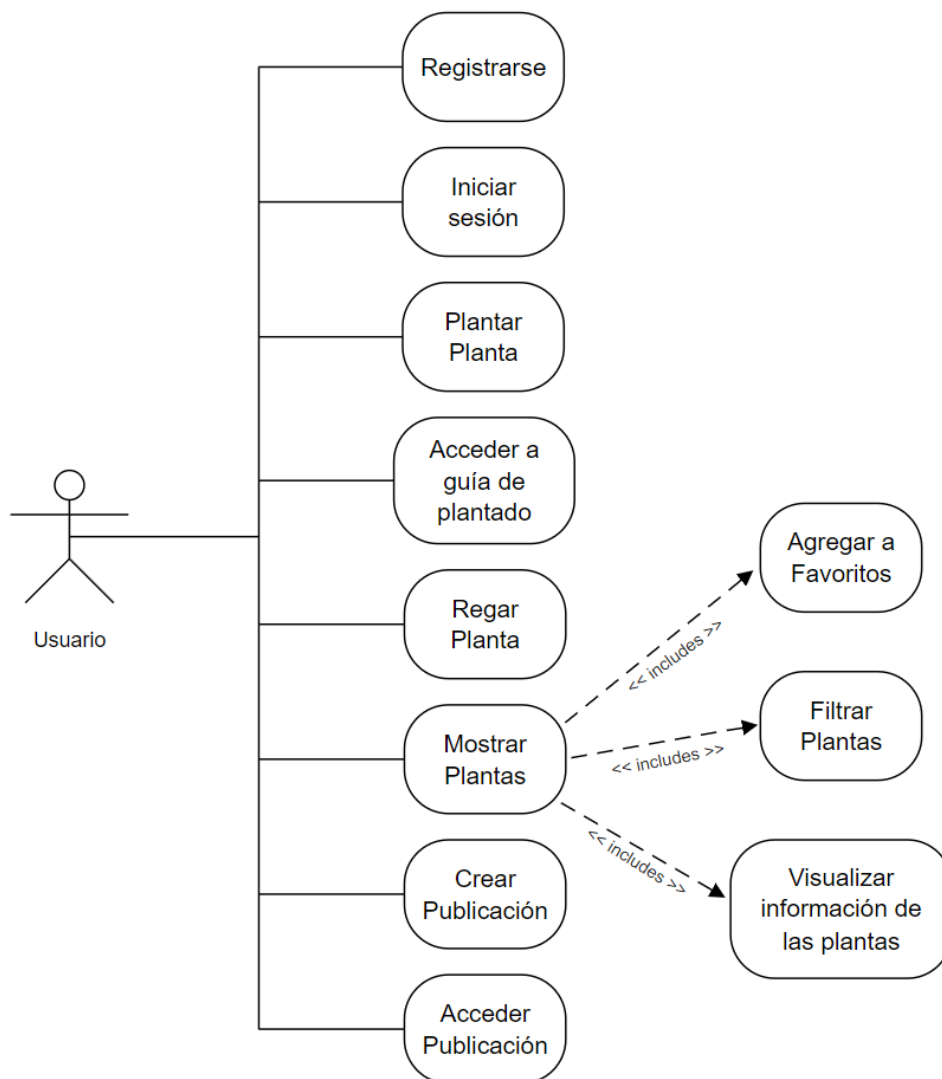


Figura 19: Diagrama de CU de la aplicación

Los requisitos funcionales que presenta la aplicación se listan y explican brevemente a continuación:

- **Guía de plantado:** Información principal de las plantas disponibles para plantar. Desde las condiciones ambientales y el tiempo medio de todo el proceso hasta el tamaño de la maceta y las posibles enfermedades que podrían afectar a la planta.
- **Calendario de plantado:** Ayuda visual que permite al usuario saber que plantas cultivar en que meses del año.
- **Información sobre crecimiento:** Información detallada de cada paso del proceso de plantado. Indicando que problemas se pueden enfrentar y que hacer en cada etapa del cultivo
- **Tienda de productos:** Pequeña tienda donde se encuentran recogidas las principales herramientas y materiales que se puedan necesitar para empezar los cultivos desde cero.
- **Huerto simulado:** Lugar donde se recogerá y mostrará de forma visual todas las plantas a las que se les está realizando seguimiento.
- **Foro:** Sitio donde todos los usuarios podrán publicar mensajes e interactuar con otros usuarios. Cada planta ofrecida por la aplicación tendrá su propio foro para que el usuario tenga más facilidad a la hora de buscar la información de un cultivo en concreto
- **Seguimiento de la planta:** Se presentará un seguimiento de cada etapa de crecimiento de la planta con toda la información relevante, presentada de forma sencilla y entendible para cualquier tipo de usuario, haciendo muy simple el proceso de cultivo.
- **Perfil de usuario:** Apartado donde los usuarios podrán modificar sus datos personales y podrán también personalizar el perfil que se mostrará cuando escriban en el foro

Respecto a los requisitos no funcionales se tienen los siguientes:

- **Facilidad de uso:** La aplicación debe presentar una interfaz de usuario que sea sencilla, con la menor cantidad de botones y con ventanas de confirmación para evitar que el usuario se equivoque, y en el caso de hacerlo, poder volver atrás.
- **Seguridad:** Los métodos de la aplicación han de utilizar los datos sensibles, como las contraseñas, de una forma cifrada para evitar posibles filtraciones de datos indeseadas y cumplir con las regulaciones pertinentes (Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos [9]).

5.3 Diseño

5.3.1 Arquitectura global

La arquitectura que se seleccionó para el desarrollo de la aplicación es la arquitectura por capas (esquema visible en la figura 20). Esta arquitectura divide la estructura de la aplicación en 3 capas con funciones muy bien diferenciadas:

- **Capa de presentación:** Es donde se recoge todo el código y recursos relacionados con la interfaz de usuario y es la encargada de recoger todas las entradas del usuario para enviarlas a la capa de lógica de negocio.
- **Capa de Lógica de negocio:** Contiene la lógica central de la aplicación. Procesa las entradas de la capa de presentación, realiza cálculos y toma decisiones, aplicando las reglas de negocio y coordinando la comunicación entre la capa de presentación y la capa de datos.
- **Capa de Datos:** La capa de datos se encarga de la persistencia y recuperación de datos. Esta capa maneja las interacciones con la base de datos, asegurando que los datos se almacenen y recuperen de manera eficiente y segura.

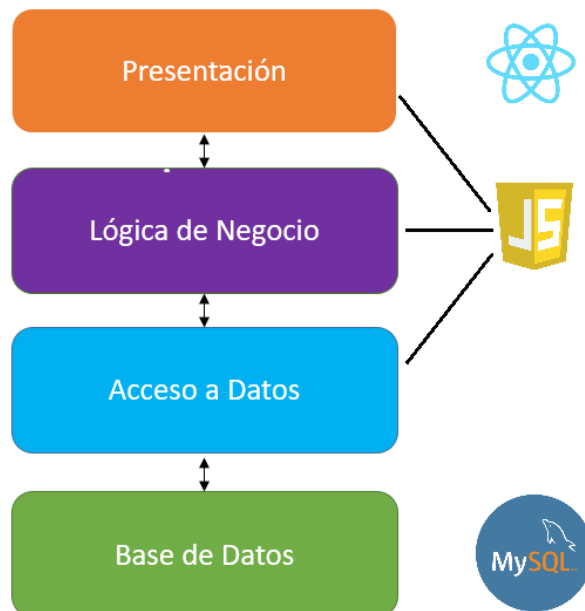


Figura 20: Esquema de la arquitectura por capas

5.3.2 Modelo de Base de Datos

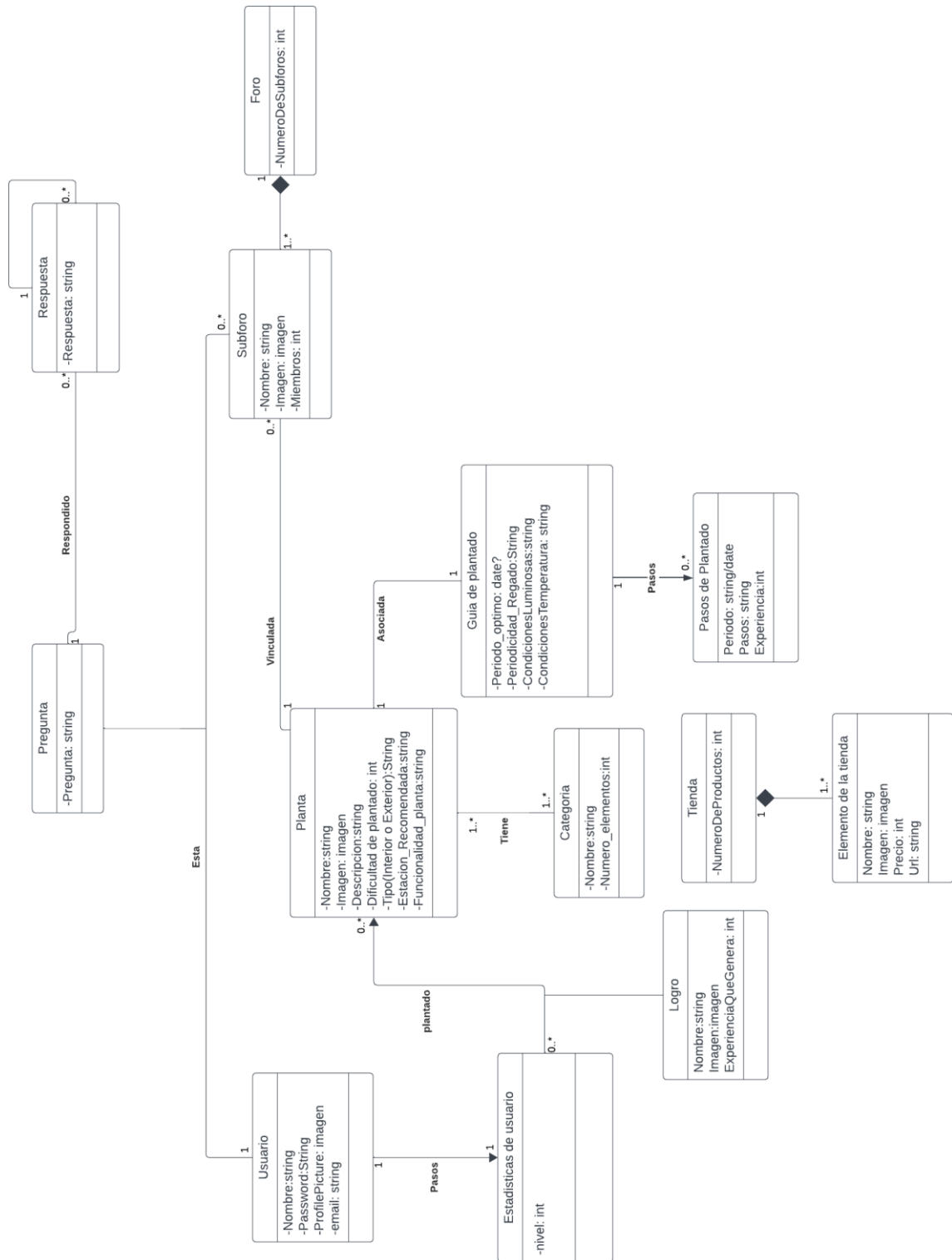


Figura 21: Diagrama de clases de la BBDD

5.3.3 Diagrama de componentes

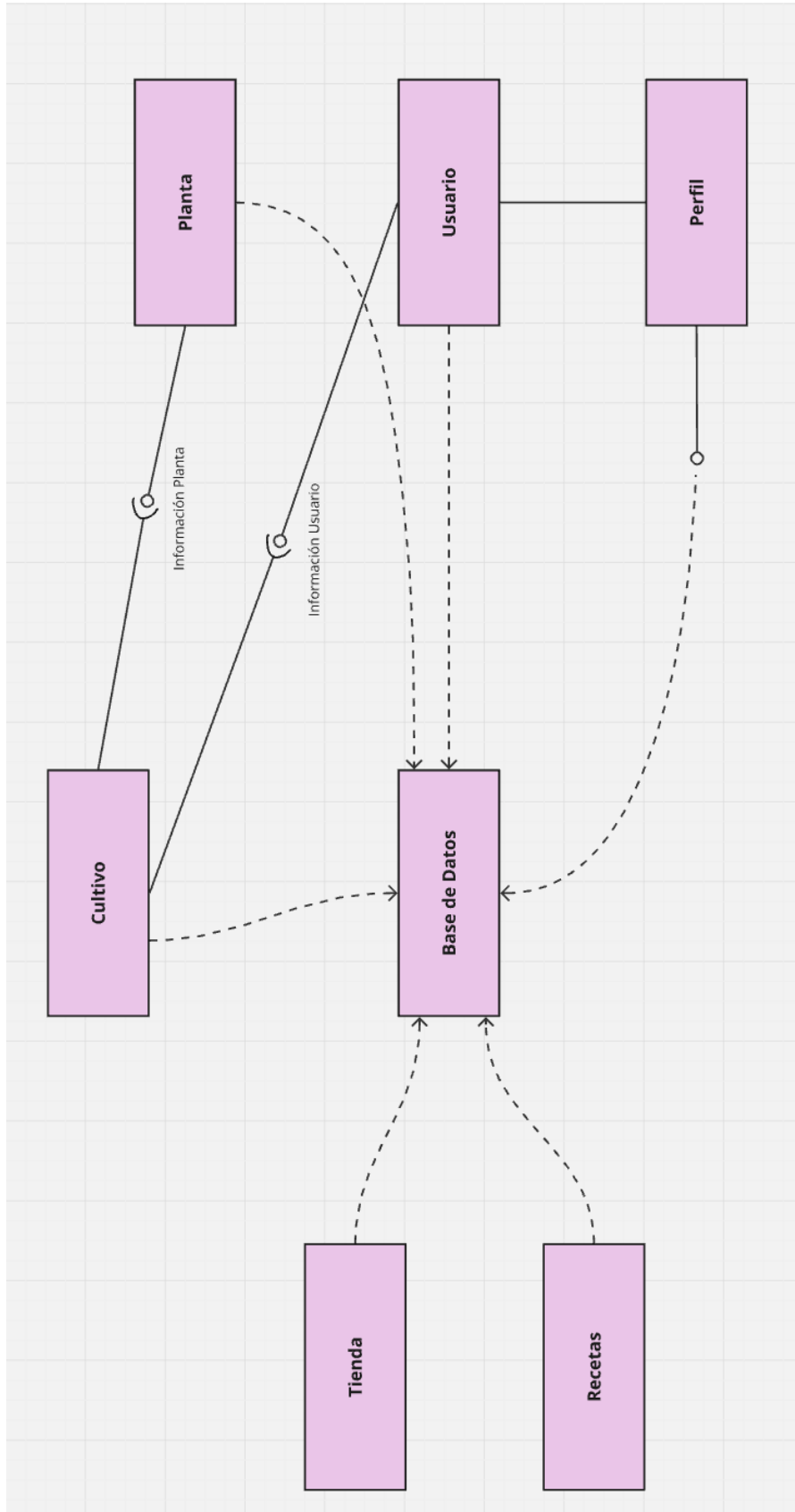


Figura 22: Diagrama de Componentes

5.4 Programación

5.4.1 Patrón API Rest

Una API REST ¹⁸ es una interfaz que permite la comunicación entre sistemas a través de HTTP, basado en una arquitectura sin estado, cliente-servidor, y caché. Identifica los recursos mediante URI y los manipula utilizando métodos HTTP estándar como GET, POST, PUT, DELETE y PATCH. Las respuestas se presentan en formato JSON y utilizan códigos de estado HTTP para informar sobre el resultado de las solicitudes.

En la figura 23 se puede observar el índice del que está compuesta la API.

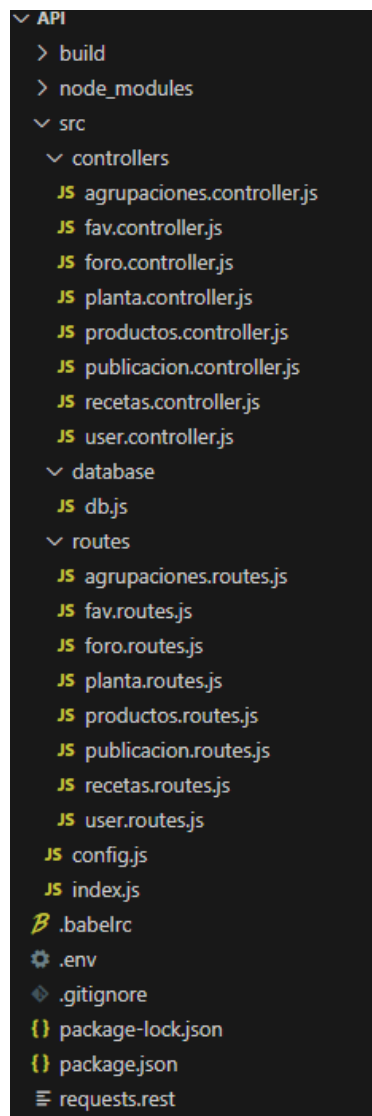


Figura 23: Índice de archivos de API

¹⁸ API REST principios de diseño. Disponible en: <https://www.ibm.com/es-es/topics/rest-apis>

Se pueden observar 2 grupos de archivos que merecen la pena explicar:

- **Controllers:** En estos archivos se recogen todos los métodos de acceso a datos relacionados con cada una de las clases.
- **Routes:** Estos archivos contienen las rutas de cada método. Sirven para enlazar una ruta url con los métodos de las clases controller.

Por último, mencionar el archivo `index.js`, que unifica todos los archivos routes para que la API pueda redireccionar las solicitudes al método correcto.

5.4.2 Huerto Simulado

Para realizar la ventana del huerto simulado, se enfrentaron varios retos a lo largo de su desarrollo. El objetivo era que el huerto simulado fuese una ventana donde se pudiese ver un *grid* 3 x 3, donde cada celda fuese un botón donde se mostrara la planta que había seleccionado el usuario previamente.

El primer problema que se presentó fue crear el grid, ya que al intentarlo en un principio no se conseguía que los botones mantuvieran los límites de cada celda y acababan siendo muy grandes y tapando a los demás. Se acabó solucionando con el uso del componente *touchable*¹⁹ que ofrece React de forma nativa. Estos componentes actúan de la misma manera que un botón, pero son mucho más personalizables permitiendo modificar el tamaño a las dimensiones que nosotros queríamos.

El siguiente reto fue rellenar este grid con las plantas que el usuario había plantado y con huecos vacíos para añadir más en el caso de que el usuario quisiera. Originalmente se quería que cada celda del grid fuera independiente y mantuviera su estado, permitiendo así que se pudiesen intercalar huecos con planta con huecos vacíos, pero esta idea fue desechada rápidamente, tras ver que era algo muy costoso sin que aportara excesivo valor. Tras implementar esta funcionalidad, se obtuvo una primera versión del huerto simulado, al que se le realizaron las pruebas que se mencionarán a continuación pudiéndose ver esta versión en la figura 33.

Tras comprobar que la funcionalidad era correcta y que se comportaba de acuerdo con los requisitos establecidos, nos pusimos en contacto con unas compañeras de la carrera de Bellas Artes, con las que teníamos trabajo conjunto, para que hicieran los iconos y la interfaz de usuario del huerto. Una vez recibido todo eso se implementó y obtuvimos la versión final, visible en la figura 40 en el capítulo 6.

¹⁹ Componente *Touchable* React: <https://reactnative.dev/docs/handling-touches>



5.4.3 Refactoring

A lo largo de todo el desarrollo, aunque con más intensidad a lo largo del sprint 2, se ha ido aplicando refactoring al código con el fin de mejorarlo. El refactoring[10] es una técnica que se utiliza para mejorar el diseño de un código ya existente mediante la aplicación de cambios y transformaciones de pequeño tamaño, que de forma acumulativa introducen una mejora significativa en el código.

Se han aplicado diferentes técnicas referentes a las variables y los parámetros de los métodos. Simplificando la lectura de estos, mejorando su rendimiento y añadiendo nombres de variables descriptivas, ayudando así a una mejor comprensión del funcionamiento de estos.

También se retiraron numerosas líneas de código que eran redundantes, puesto que declaraban y usaban métodos similares en diferentes partes del código, en lugar de usar un método común, y se eliminaron numerosos imports y llamadas a funciones de librerías de las cuales no se hacía uso.

Por último, se simplificaron las llamadas a métodos para evitar que las llamadas a estos fuesen muy extensas y con demasiados parámetros.

5.5 Pruebas

En este apartado se hablará de todas las pruebas que se han realizado a lo largo del proceso de desarrollo para comprobar que el funcionamiento del sistema y sus funcionalidades son correctos.

Las pruebas que se han pasado son de tres tipos, explicados a continuación [11]:

- *Pruebas de Integración:* El producto ensamblado se prueba para comprobar si cumple con los requisitos funcionales y los no funcionales (rendimiento, seguridad, ...).
- *Pruebas Unitarias:* Prueban el funcionamiento de un componente concreto de forma individual.
- *Pruebas de Aceptación:* Prueban que realiza el cliente sobre el producto final para determinar si acepta el estado en el que está o si desea realizar modificaciones o introducir requisitos adicionales.

5.5.1 Pruebas de integración

Para las pruebas de integración, el interés residía en comprobar la correcta comunicación de la API con el frontend y la correcta aplicación de los métodos que se habían implementado. Se hizo uso del emulador de Android Studio para poner en marcha la aplicación y también se usó la herramienta Postman para captar el flujo de mensajes entre el front y la API. También se hizo uso de la herramienta DBeaver para comprobar que la API, una vez llamados a los métodos correctos, realizaba la persistencia de manera consistente y correcta.

Registro de usuario:

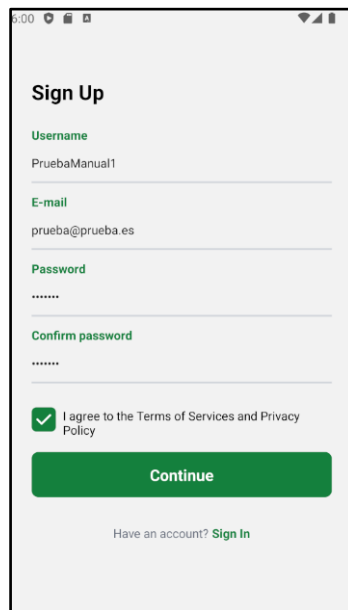


Figura 24: Ventana registro

123 id	abc username	abc password	abc email	123 profilePictureId
1	JuanPerez	clave_segura	juan.perez@example.com	1
2	MariaGomez	otra_clave_segura	maria.gomez@example.com	1
3	CarlosRodriguez	contraseña_fuerte	carlos.rodriguez@example.com	1
4	a	a	a	13
8	Elxkx	Xifkfk	Fofifi	1
9	Fkejcgj	Xorkrrkfgkfkffif	Ckrkelcloffo	1
10	Karro	1234	Karro@gmail.com	3
11	Bchskj	a	Bjknc,	1
12	Ckfkckck	Obo	Xorkckck@obo.com	1
13	Ar	V	R@ovo.com	1
14	Arg	O	Obo@obo.com	1
15	PruebaManual1	prueba1	prueba@prueba.es	1

Figura 25: Tabla de Usuarios en BBDD

Para esta prueba nuestra intención es comprobar que el registro de usuario funcione correctamente. Para ello registramos un usuario con nombre “PruebaManual1”, email “prueba@prueba.es” y contraseña “prueba1” rellenando el formulario de registro.

Al introducir todos los datos correctamente y pulsar el botón de continuar, comprobamos que en la base de datos se ha creado correctamente la entrada en la tabla de Usuarios. También vemos que la API devuelve el usuario que acabamos de registrar.



Inicio de sesión:

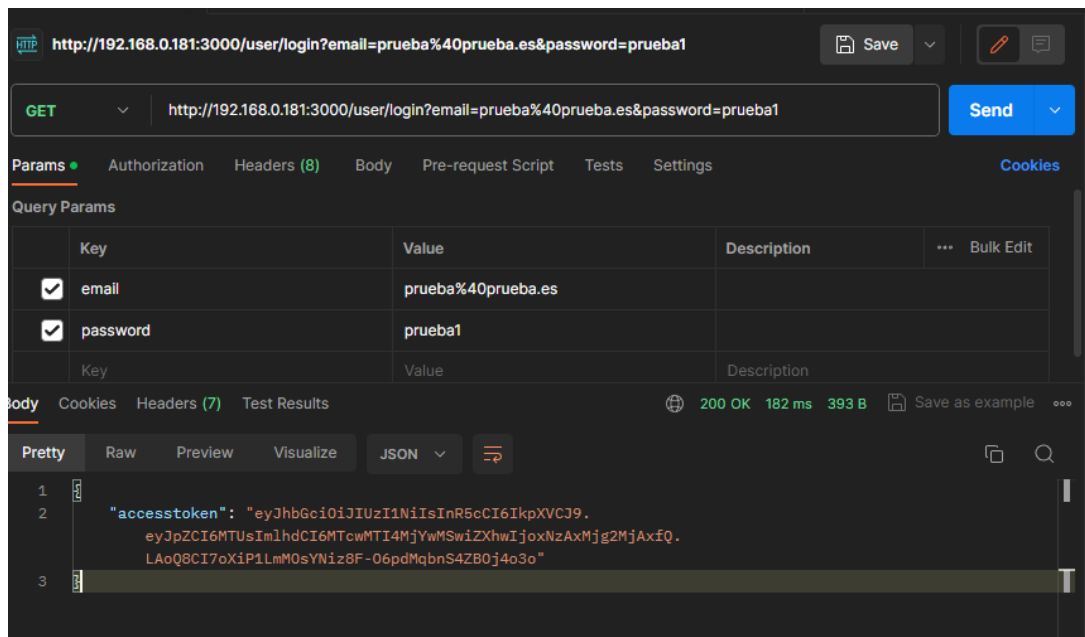


Figura 26: Test login con PostMan

Para esta prueba se quiere testear el comportamiento del inicio de sesión. Esta funcionalidad recibe un usuario y una contraseña para comprobar que existe en la base de datos. En caso afirmativo, devuelve un token de acceso con el que el usuario se autentica.

Como se puede ver en la figura 26, se le pasó a la API las credenciales de un usuario creado para hacer estas pruebas (usuario: “prueba@prueba.es”, contraseña “prueba1”) mediante Postman y una solicitud GET y devuelve el access token correspondiente al mismo, dando por válido el inicio de sesión.

Modificación de usuario:

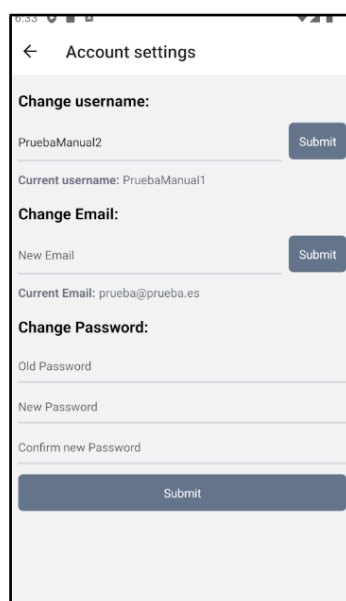


Figura 27: Ventana perfil de usuario

id	username	password	email
1	JuanPerez	clave_segura	juan.perez@example.com
2	MariaGomez	otra_clave_segura	maria.gomez@example.com
3	CarlosRodriguez	contraseña_fuerte	carlos.rodriguez@example.com
4	a	a	a
8	EloXoX	Xifkfk	Fofifi
9	Fkejcgj	Xorkrkkfkgkfkflif	Ckrkelccloffo
10	Karro	1234	Karro@gmail.com
11	Bchskj	a	Bjknc,
12	Ckfkckck	Obo	Xorkckck@obo.com
13	Ar	V	R@ovo.com
14	Ara	O	Obo@obo.com
15	PruebaManual2	prueba1	prueba@prueba.es

Figura 28: Tabla usuarios BBDD tras modificar contraseña

Para esta prueba se quiere comprobar que se realiza correctamente cualquier modificación de los datos del usuario registrado, cuando se utiliza la ventana creada para ese fin.

Para ello se modificó el nombre de usuario actual a “PruebaManual2” comprobando, tal como se ve en la figura 27, que al confirmar el cambio pulsando “Submit” se actualiza en la base de datos.

Añadir a favoritos:

	UsuarioID	PlantalD
1	15	1
2	15	2
3	15	3
4	4	4
5	4	5
6	4	6
7	4	16

Figura 29: Tabla Favoritos BBDD tras añadir nuevos favoritos

En esta prueba se quiere comprobar que la funcionalidad de añadir como favoritas plantas de la lista de plantas se comporta correctamente.

Para ello, se seleccionan como favoritos los tomates, las fresas y las moras (con id en base de datos de 1, 2 y 3 respectivamente). Tras realizar esto, nos dirigimos a DBeaver para comprobar la tabla “Favoritos”. En la base de datos se puede observar que se añaden los favoritos en la tabla correcta, tal como se ve en la figura 29.



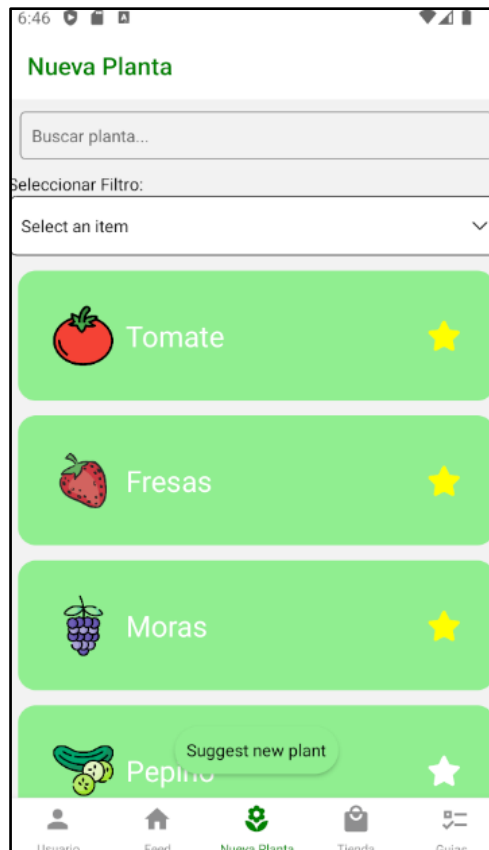


Figura 30: Ventana lista de plantado con nuevos favoritos

Comenzar plantado:

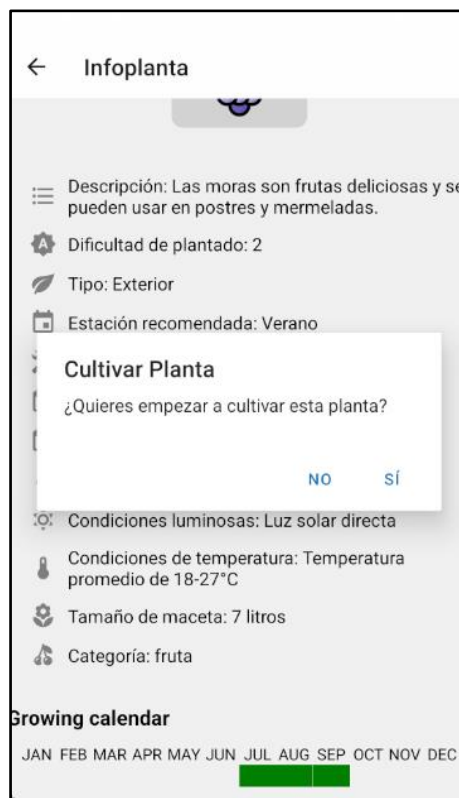


Figura 31: Ventana "guía de plantado"

Para esta prueba se comprobará que la funcionalidad de nuevo cultivo de planta funciona correctamente, añadiendo la planta al Huerto Simulado y registrándola en la Base de Datos.

Se comienza a plantar un nuevo cultivo desde la ventana de información de la planta. Al presionar sobre el botón de “sí” en la ventana emergente, se registra una nueva entrada en la base de datos con el id del usuario que la planta, si la planta ha sido regada, fecha de plantado y que planta es.

id	paso	regada	fechaPlantada
1	1	0	2023-11-21
2	2	0	2023-11-30
3	4	0	2023-03-15
6	4	1	2023-11-02
11	1	1	2023-11-25
13	1	1	2023-11-29
14	1	1	2023-11-29
15	1	1	2023-11-29
16	1	1	2023-11-29
17	1	1	2023-11-29
18	7	1	2023-11-29
19	1	0	2023-11-29
20	1	0	2023-11-29
21	1	0	2023-11-29
22	2	1	2023-11-30

Figura 32: Tabla “plantadas” BBDD tras iniciar nuevos plantados

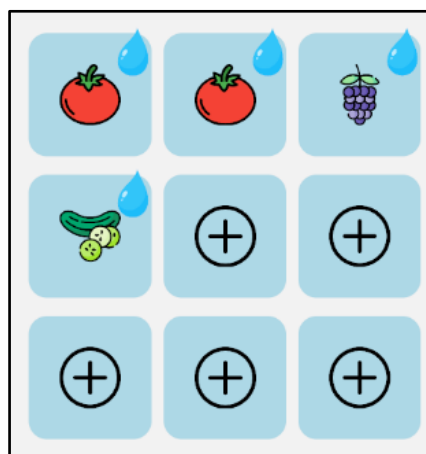


Figura 33: Plantas añadidas correctamente al huerto simulado

5.5.2 Pruebas Unitarias

Para las pruebas unitarias, nuestra intención era comprobar que los módulos principales, como son el módulo de Usuario y el de Plantas y guía de plantado, tuvieran un funcionamiento óptimo, ya que sobre ellos se iba a construir el resto de las funcionalidades de la aplicación. Para ello, se escribieron unas pruebas automáticas en Jest para poder pasar las pruebas de forma más sencilla cada vez que se realizase alguna modificación al código.

Pruebas automatizadas (Usuario):

```
import { render, fireEvent } from '@testing-library/react-native';
import RegisterScreen from '../src/screens/login/RegisterScreen';
import { Alert } from 'react-native';

jest.spyOn(Alert, 'alert');

jest.mock('../assets/check-solid.svg', () => 'CheckIcon');

jest.mock('../src/api/user', () => ({
  checkDuplicateUsername: jest.fn(),
  checkDuplicateEmail: jest.fn(),
  registerUser: jest.fn()
}));
jest.mock('react-native-safe-area-context', () => ({
  useSafeAreaInsets: jest.fn().mockReturnValue({ top: 0, bottom: 0 })
}));
jest.mock('expo-secure-store', () => ({
}));

describe('RegisterScreen', () => {
  it('debe mostrar un error si los campos están vacíos', async () => {
    const { getByText } = render(<RegisterScreen />);
    const continueButton = getByText('Continue');
    fireEvent.press(continueButton);

    expect(Alert.alert).toHaveBeenCalledWith("Error", "You must fill all the fields");
  }
});
```

Figura 34: Código de la prueba Jest para funcionalidad "Registro"

```

> culminate@1.0.0 test
> jest

PASS tests/registerScreen.test.js
  RegisterScreen
    ✓ debe mostrar un error si los campos están vacíos (488 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        2.444 s, estimated 3 s
Ran all test suites.

```

Figura 35: Resultados de la prueba Jest para "Registro"

La clase de prueba proporcionada para RegisterScreen en una aplicación móvil verifica que se muestre un error cuando se intenta continuar sin completar los campos requeridos. Utiliza mocks para simular componentes externos y alertas nativas de React Native. Al presionar el botón "Continue", se comprueba si se llama a una alerta de error adecuada, asegurando así la validación correcta de los campos en la pantalla de registro.

Pruebas automatizadas (Guía de plantado y Favoritos):

```

import React from 'react';
import { render, fireEvent, waitFor } from '@testing-library/react-native';
import GuiaPlantado from '../src/screens/GuiaPlantado';
import { NavigationContainer } from '@react-navigation/native';
import { createStackNavigator } from '@react-navigation/stack';

const Stack = createStackNavigator();

jest.mock('@react-navigation/stack', () => ({
  createStackNavigator: jest.fn().mockReturnValue({ Navigator: jest.fn(),
Screen: jest.fn() }),
}));

describe('GuiaPlantado Screen Test', () => {
  it('should update to the next step on button press', async () => {
    const mockRoute = {
      params: {
        plantaData: {
          id: 11,
          planta_id: 1,
          paso: 1,
          nombre: "Tomate",
          fechaPlantada: "2023-11-24T23:00:00.000Z",
          periodicidad_regado: 4,
          condiciones_temperatura: "20°C a 30°C"
        }
      }
    };

    const { getByText, findByText } = render(
      <NavigationContainer>
        <Stack.Navigator>

```

```
<Stack.Screen
  name="GuiaPlantado"
  component={GuiaPlantado}
  initialParams={mockRoute.params}
/>
</Stack.Navigator>
</NavigationContainer>
);

await waitFor(() => {
  const result = findByText("Planta: Tomate");
  debug();
  return result;
}, { timeout: 20000 });

const nextStepButton = getByText("Siguiete paso");
fireEvent.press(nextStepButton);
}, 30000);
});
```

Figura 36: Código de la prueba Jest para "Guía de Plantado"

```
PASS tests/pasosHuerto.test.js
  GuiaPlantado Screen Test
    ✓ should update to the next step on button press (1039 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  0 total
Time:        5.616 s
Ran all test suites.
```

Figura 37: Resultados prueba "Guía de Plantado"

Esta clase de prueba automatizada valida la funcionalidad de la pantalla GuiaPlantado en una aplicación móvil. Se simula la navegación y los datos de entrada para una planta específica (Tomate) con detalles como el paso actual de plantación y condiciones de cultivo. La prueba espera que la información de la planta se cargue correctamente y luego simula un evento de botón para avanzar al siguiente paso de plantación, asegurando que la pantalla responda como se espera en el flujo de usuario de plantación.

Tanto las pruebas unitarias como las de integración se aplicaron en regresión, es decir, tras la introducción de nuevas funcionalidades, ventanas o datos; con la finalidad de detectar cualquier error en el nuevo módulo o fallos en las funcionalidades ya preexistentes.

5.5.3 Pruebas de Aceptación

Las pruebas de aceptación permiten verificar, por parte del cliente, el correcto funcionamiento del software desarrollado según unas pruebas que, en teoría, el mismo cliente ha especificado.

Como en este caso, el “cliente” que especificaba los requisitos era el equipo de desarrollo, las pruebas de aceptación se iban pasando a medida que se producían las diferentes ventanas y funcionalidades. Siguiendo la metodología ágil antes mencionada, tras terminar de programar una unidad de trabajo, se le realizaban las pruebas que fueran convenientes para asegurarse que el módulo producido era correcto respecto a los requisitos planteados al inicio y de no pasar esas pruebas, el módulo se volvía a programar para corregirlo.

Se realizaron en total 8 pruebas unitarias, 6 pruebas de integración y numerosas pruebas de aceptación, aplicándolas tras cumplir alguna de las condiciones anteriormente mencionadas, teniendo una alta efectividad para ayudar a detectar fallos graves de funcionamiento en la API y para detectar fallos de validación en los formularios y botones de las ventanas.



6. Descripción de la aplicación

En este capítulo se van a describir y explicar las diferentes ventanas que presenta la aplicación, así como el funcionamiento que estas mismas presentan.

Al abrir la aplicación se observa la ventana de inicio de sesión, la cual ofrece la posibilidad de o bien iniciar sesión con unas credenciales registradas (ver figura 38) o registrarse como un usuario nuevo (ver figura 39). Remarcar que es necesario tener una cuenta registrada para poder hacer uso de todas las funcionalidades que ofrece la aplicación.



Figura 38: Ventana "Login"

The image shows a mobile application registration screen. At the top, there is a green header with a white back arrow on the left and a white pumpkin icon on the right. Below the header, the word 'FORMULARIO' is written in white. The main content area is white and contains a toggle switch labeled 'DATOS BÁSICOS' which is currently turned on. Below the toggle are five rounded rectangular input fields, each with a label: 'NOMBRE COMPLETO', 'NOMBRE DE USUARIO', 'CORREO ELECTRÓNICO', 'CONTRASEÑA', and 'CONFIRMA CONTRASEÑA'. At the bottom center, there is a rounded rectangular button with a right-pointing arrow.

Figura 39: Ventana "Registro"

Una vez el usuario ha iniciado sesión, la aplicación ofrece 6 ventanas a las que se puede navegar pulsando sobre los iconos que se encuentran en la barra de la parte inferior de la pantalla.

La primera ventana es el Huerto Simulado. En ella se muestran las plantas que el usuario está cultivando en ese momento. Si el usuario no tiene ningún cultivo en curso el huerto presenta las macetas vacías y si por el contrario el usuario tiene algún cultivo en ese momento, presenta el aspecto de la figura 40.



Figura 40: Ventana "Huerto Simulado"

Al pulsar una maceta vacía (las que tienen el icono de +) o pulsar el segundo icono de la barra de navegación, se navegará a la ventana de "Lista de plantas". En esta ventana se muestran todas las plantas de las que la aplicación ofrece seguimiento e información, como se observa en la figura 41.

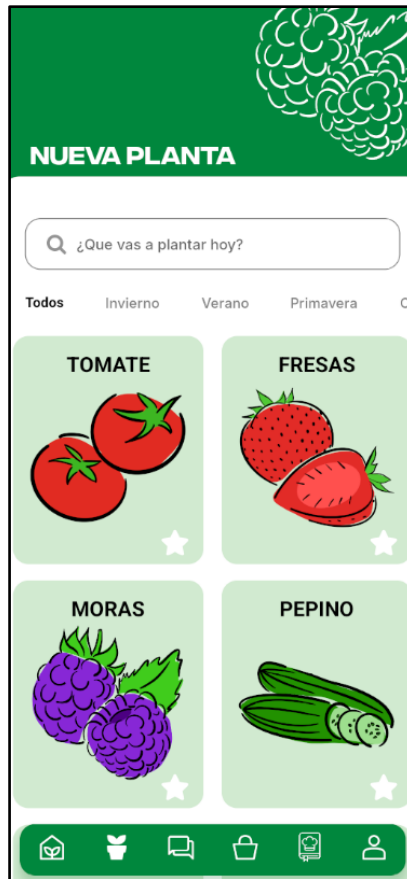


Figura 41: Ventana "Lista Plantas"



Figura 42: Ventana "Información de Planta"

Tras seleccionar cualquier planta de la lista, se abrirá una ventana donde se muestra información muy detallada sobre las condiciones del cultivo. Entre esta información se encuentra la dificultad del cultivo, las condiciones ambientales (luz, temperatura, humedad), los meses del año en los que se ha de plantar, el tamaño de maceta y las enfermedades y plagas más comunes. Esta ventana se puede ver en la figura 42.

La tercera ventana de la barra de navegación corresponde a la funcionalidad del foro. Como se puede observar en la figura 43, cada cultivo posee su propio foro dentro del cual todos los usuarios pueden publicar mensajes o responder a mensajes escritos por otros usuarios.



Figura 43: Ventana "Foro"

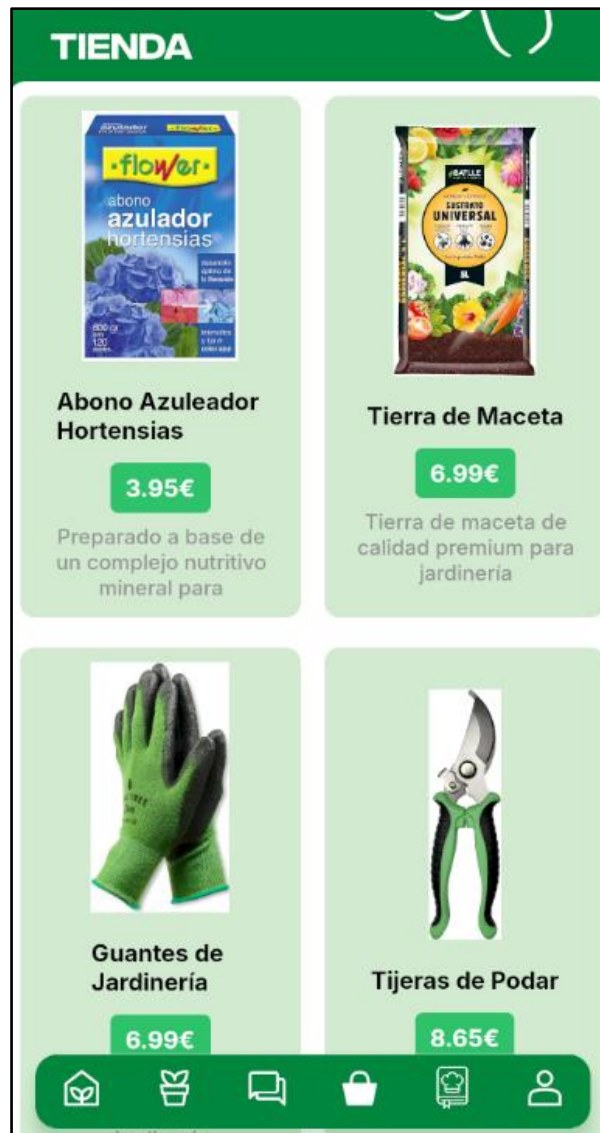


Figura 44: Ventana "Tienda"

El icono de la cesta es la tienda. En ella se encuentran los principales materiales y herramientas necesarias para crear un cultivo desde cero. Al pulsar sobre cualquier elemento de la lista, se redirige al usuario a un portal externo donde podrá adquirir estos productos.

El siguiente icono al de la tienda corresponde a la lista de recetas. Estas recetas se muestran en una lista y contienen como ingrediente principal alguno de los cultivos que ofrece la aplicación. De esta manera se pretende ofrecer ideas para utilizar los productos producidos en el cultivo.



Figura 45: Ventana "Lista de Recetas"



Figura 46: Ventana "Información de Receta"

Por último, la ventana de la derecha del todo de la barra corresponde a la ventana del perfil de usuario. En esta ventana se ofrecen diferentes opciones para personalizar el perfil que se mostrará en el foro y diferentes ajustes para personalizar la experiencia de uso de la aplicación.



Figura 47: Ventana "Perfil de usuario"

7. Cronología del TFG

7.1 Backlog Inicial

El backlog inicial consistía en las siguientes unidades de trabajo ordenadas por nivel de importancia de mayor a menor:

- Recomendaciones de recetas
- Tratamiento de enfermedades de plantas
- Añadir plantas adicionales
- Mejorar información de plantas actuales
- Refactoring y reorganización del código del proyecto
- Mejora de métodos de API y cifrado de información sensible en base de datos
- Ventana de FAQ
- Mejoras en funcionalidad de ventana de foro
- Recomendaciones personalizadas
- Sistema de amigos

La estrategia que se ha seguido para la priorización de las unidades de trabajo fue la de terminar de implementar las funcionalidades ideadas al inicio del desarrollo de la idea de negocio. Tras realizar esta implementación, se da paso a la mejora de la estructura del código mediante refactoring para cimentar una base estable para poder dar paso en un futuro a funcionalidades adicionales.

7.2 Línea temporal

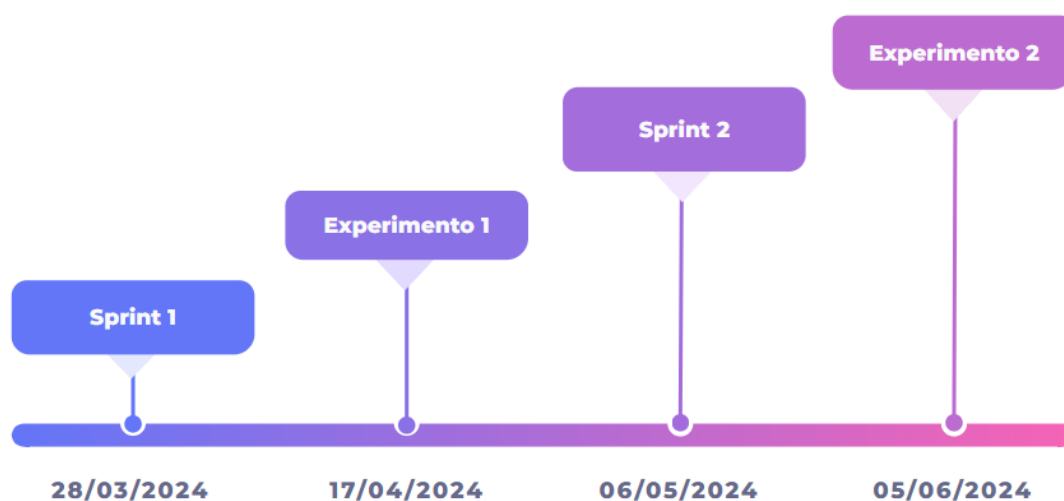


Figura 48: Línea Temporal del desarrollo [12]

Las unidades de trabajo que se seleccionaron para realizar en el primer sprint son las que se pueden leer en la figura 49. Este sprint tuvo una duración de 3 semanas, comenzando el 28 de marzo de 2024 y concluyendo el día 15 de abril de 2024. A la fecha de finalización del sprint se completaron todas las unidades de trabajo que se plantearon al inicio de este.

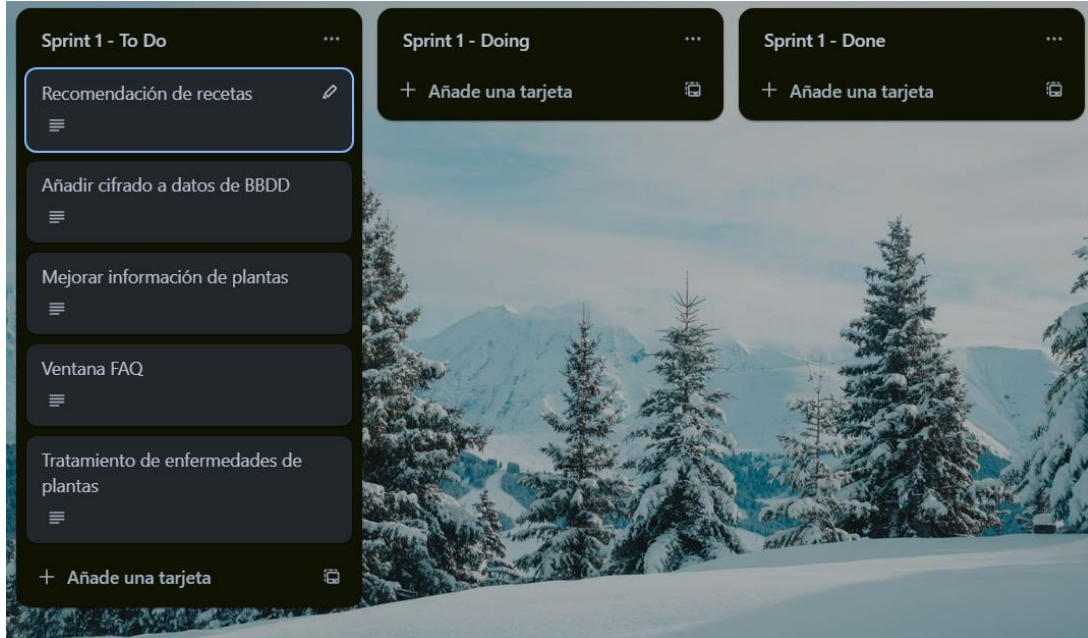


Figura 49: Contenido del sprint 1

Dos días después de la finalización del sprint, se llevó a cabo el experimento con los early adopters. Se les dio a probar la aplicación a un conjunto de 7 personas y se les pidió que, tras usarla durante un tiempo, contestaran a una serie de preguntas que hacen referencia a los diferentes aspectos de la aplicación.

Las preguntas se valoraban con una nota numérica del 1 al 5, siendo 1 muy en desacuerdo y 5 muy de acuerdo. Las preguntas que se hicieron fueron las siguientes:

- La aplicación tiene todas las funcionalidades que necesito para gestionar mis cultivos.
- Puedo registrar y hacer seguimiento de mis cultivos fácilmente.
- Las recomendaciones sobre cuidados y mantenimiento de los cultivos son precisas y útiles.
- La navegación dentro de la aplicación es intuitiva y fácil de entender.
- Encuentro fácilmente la información que necesito dentro de la aplicación.
- Las instrucciones y tutoriales dentro de la aplicación son claros y útiles.

Cultivate: desarrollo de una aplicación para gestionar cultivos urbanos

- El diseño de la aplicación es atractivo y agradable a la vista.
- La aplicación carga la información rápidamente al abrirla.
- Estoy satisfecho con mi experiencia general usando la aplicación.
- Recomendaría esta aplicación a otras personas interesadas en cultivos.

Los resultados de la encuesta se pueden observar a continuación:

La aplicación tiene todas las funcionalidades que necesito para gestionar mis cultivos.

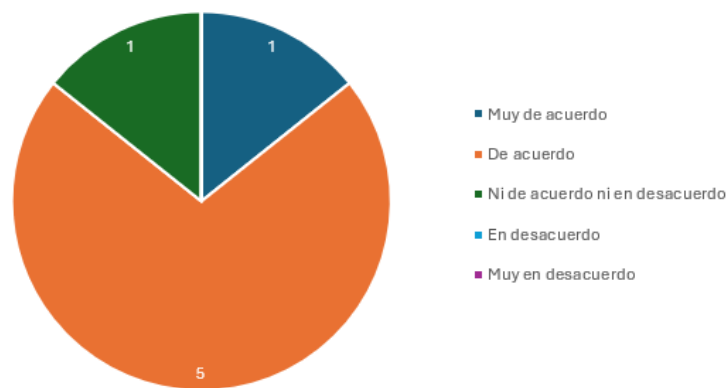


Figura 50: Respuestas sobre funcionalidades de la aplicación

Puedo registrar y hacer seguimiento de mis cultivos fácilmente.

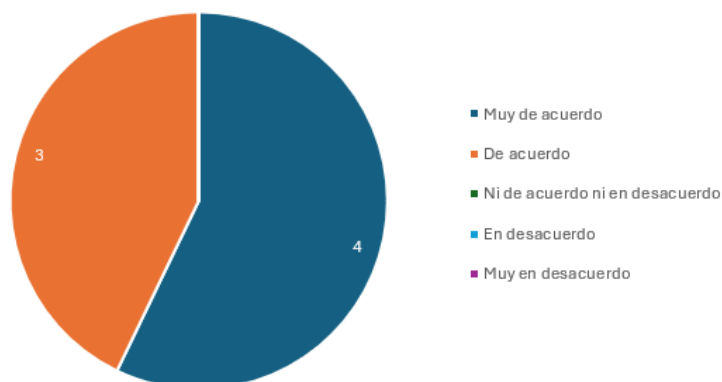


Figura 51: Respuestas sobre registro y seguimiento de cultivos

Las recomendaciones sobre cuidados y mantenimiento de los cultivos son precisas y útiles.

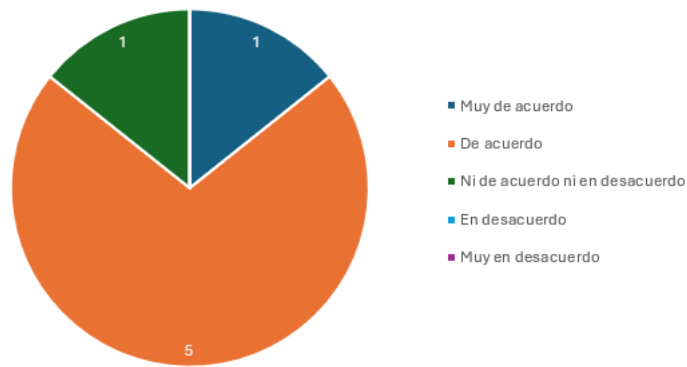


Figura 52: Respuestas sobre recomendaciones de cuidados y mantenimiento

La navegación dentro de la aplicación es intuitiva y fácil de entender.

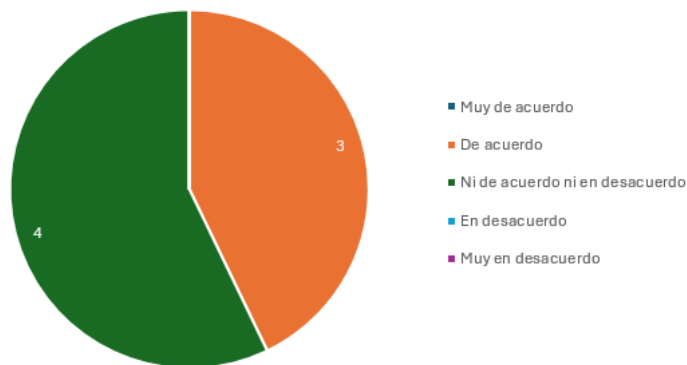


Figura 53: Respuestas sobre navegación sencilla e intuitiva

Encuentro fácilmente la información que necesito dentro de la aplicación.

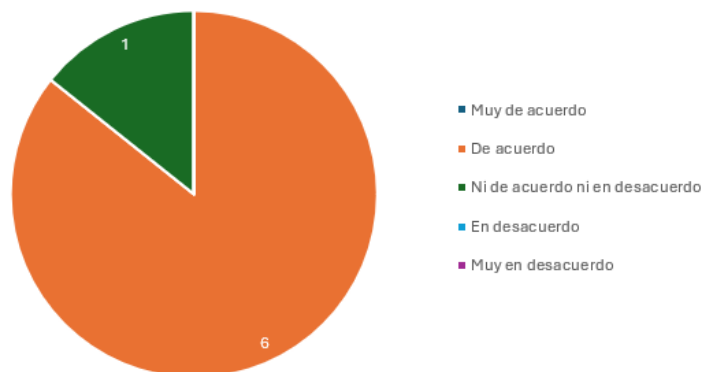


Figura 54: Respuestas sobre información recogida en la aplicación

Las instrucciones y tutoriales dentro de la aplicación son claros y útiles.

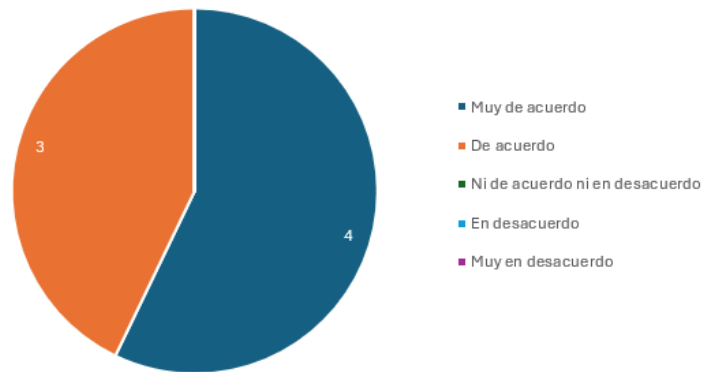


Figura 55: Respuestas sobre claridad de la información mostrada en la aplicación

El diseño de la aplicación es atractivo y agradable a la vista.

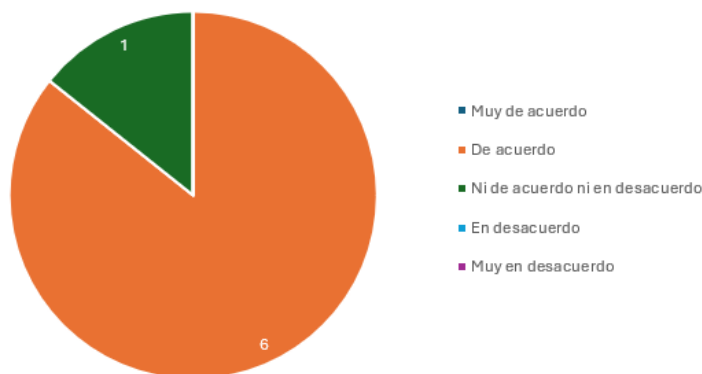


Figura 56: Respuestas sobre diseño de la aplicación

La aplicación carga la información rápidamente al abrirla.

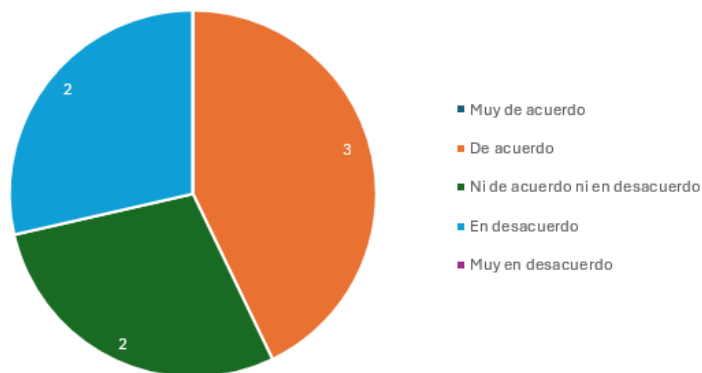


Figura 57: Respuestas sobre rendimiento de la aplicación

Estoy satisfecho con mi experiencia general usando la aplicación.

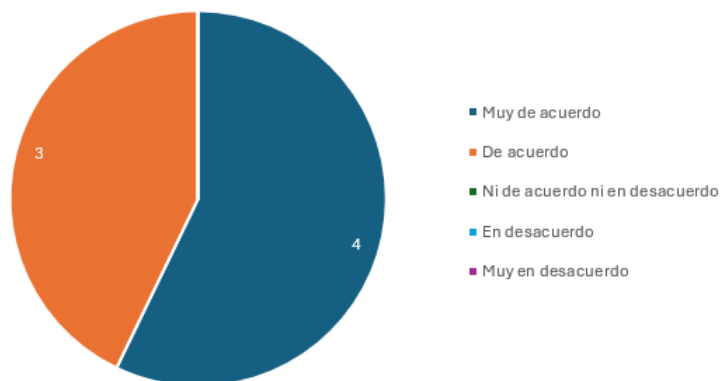


Figura 58: Respuestas sobre satisfacción general al usar la aplicación

Recomendaría esta aplicación a otras personas interesadas en cultivos.

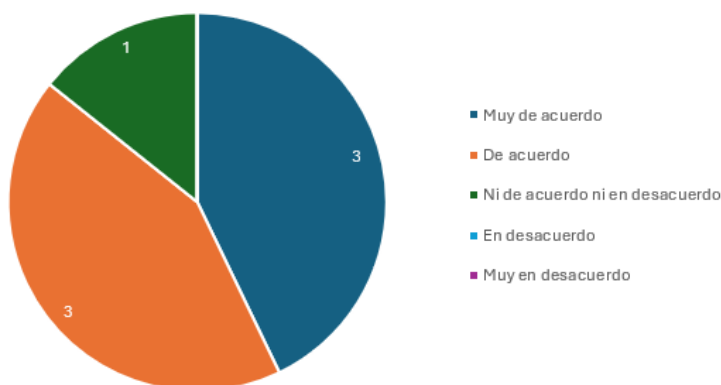


Figura 59: Respuestas sobre recomendación de uso a otras personas

Tras analizar los resultados obtenidos del experimento, se puede concluir lo siguiente:

- La interfaz cumple correctamente con el requisito de usabilidad
- La información se muestra de forma clara y ordenada
- La aplicación presenta fallos de carga y lentitud al transicionar entre algunas ventanas.
- La satisfacción general es buena, aunque se han de corregir problemas de rendimiento.

Una vez realizado este experimento y obtenido el feedback de los usuarios se planteó el contenido del segundo sprint. La lista de unidades de trabajo que se trabajaron en el sprint 2 se puede ver en la figura 60.

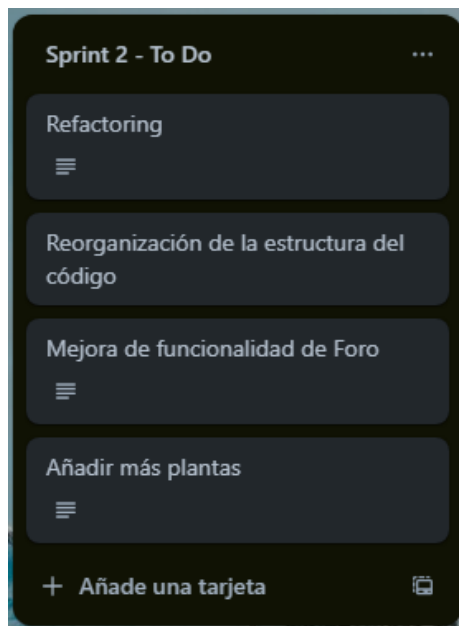


Figura 60: Contenidos del sprint 2

En este sprint, la fuerza de trabajo se centró mucho más en mejorar la fiabilidad y el rendimiento de la aplicación. Se realizó una reestructuración del código, mediante *refactoring*, para eliminar los métodos repetidos, recursos en desuso y corrección de métodos necesarios para mejorar su eficiencia.

La duración de este sprint fue algo más larga que el anterior, 3 semanas y media. Desde el 6 de mayo de 2024 hasta el día 2 de junio de 2024. El día 5 de junio se realizó otro experimento con el mismo grupo de early adopters para ver si se obtenían mejores resultados.

Las preguntas que se realizaron fueron mayormente las mismas que en el primer experimento, aunque se añadieron nuevas preguntas que hacían referencia al rendimiento de la aplicación, a fin de observar de forma empírica si la percepción de los usuarios era diferente a la del primer experimento.

Puesto que no se modificó la interfaz ni la información que se muestra en la guía de plantado, las preguntas referentes a esos temas no se mostrarán. Por el contrario, las preguntas que hagan referencia a las métricas de rendimiento se mostrarán a continuación:

- La aplicación responde rápidamente a mis acciones.
- La carga inicial de la aplicación es rápida.
- No he experimentado fallos o errores mientras uso la aplicación.

- La aplicación es estable y rara vez se cierra inesperadamente.
- La transición entre diferentes secciones de la aplicación es fluida.

Como se puede ver en los resultados de la encuesta, la percepción de los *early adopters* respecto al rendimiento general de la aplicación ha mejorado mucho. Y por tanto la satisfacción general también ha subido. Las graficas a continuación ilustran las respuestas a estas nuevas preguntas:

La aplicación responde rápidamente a mis acciones.

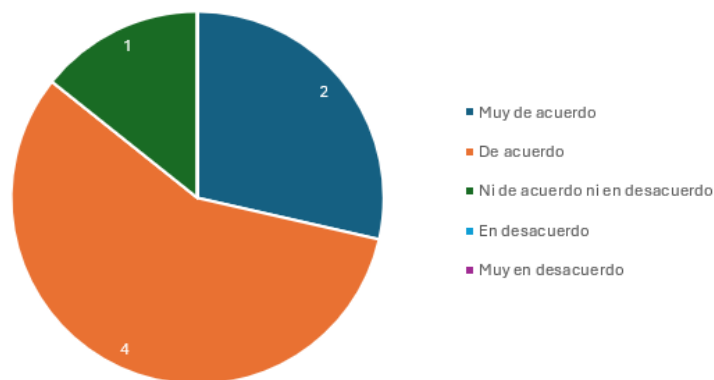


Figura 61: Respuestas sobre velocidad de respuesta de la aplicación

La carga inicial de la aplicación es rápida.

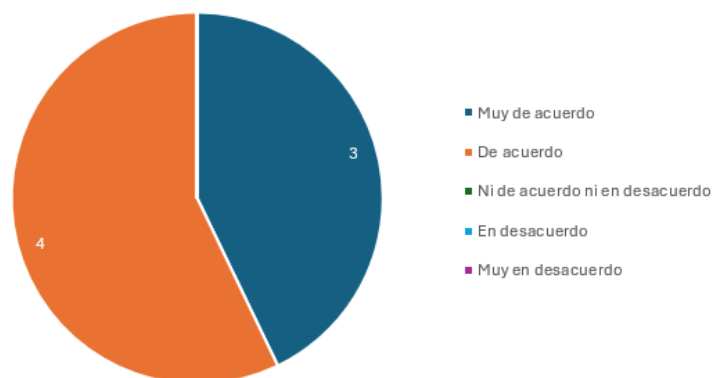


Figura 62: Respuestas sobre carga inicial de la aplicación

No he experimentado fallos o errores mientras uso la aplicación.

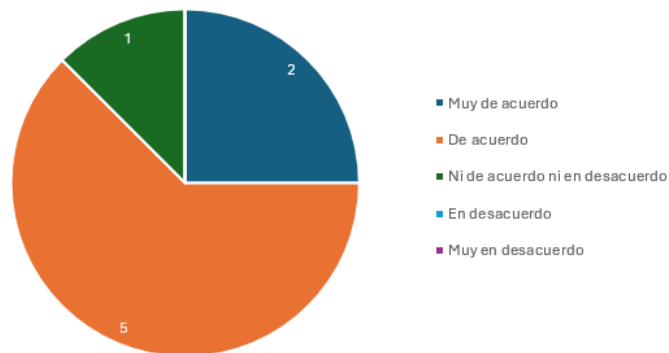


Figura 63: Respuestas sobre errores durante el uso de la aplicación

La aplicación es estable y rara vez se cierra inesperadamente.

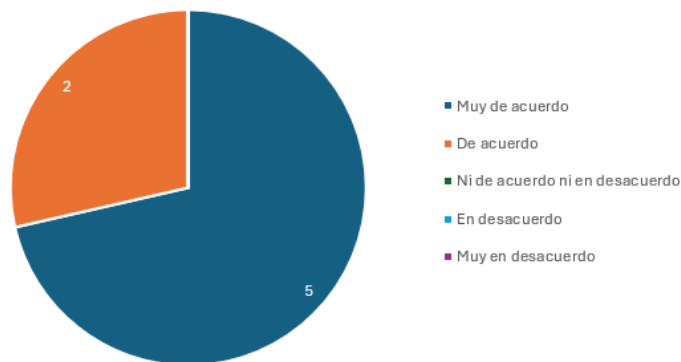


Figura 64: Respuestas sobre estabilidad de la aplicación

La transición entre diferentes secciones de la aplicación es fluida.

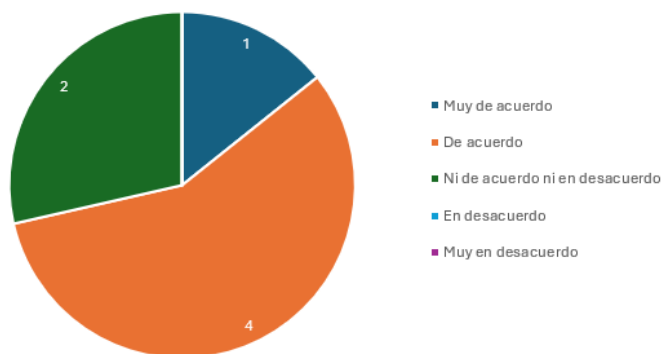


Figura 65: Respuestas sobre navegación fluida entre ventanas de la aplicación

8. Conclusiones y Trabajo Futuro

Respecto a la aplicación, podemos decir que ha cumplido con nuestras expectativas y con la de los usuarios. Desde las primeras pruebas y experimentos, siempre recibimos opiniones muy positivas lo cual nos impulsó a seguir trabajando duro en el desarrollo de esta.

La realización de este trabajo de fin de grado ha sido una experiencia enriquecedora y que nos ha permitido cumplir con la mayor parte de los objetivos que teníamos. Hemos sido capaces de crear el producto que teníamos en mente, ofreciendo gran parte de las funcionalidades que planteamos al inicio del desarrollo del proyecto.

El desarrollo de esta aplicación no habría sido posible sin los conocimientos adquiridos en estos últimos años. Las asignaturas que más nos han ayudado a la hora de estructurar el proyecto y llevarlo a cabo han sido, las de Ingeniería de Software (ISW) y Proyecto de Ingeniería de Software (PIN) donde adquirimos todos los conocimientos sobre las metodologías ágiles, poniéndolas en práctica mediante el desarrollo de proyectos de corta duración. Otra que también ha sido muy útil ha sido la de Diseño de Software, donde se nos introdujo los conceptos sobre los patrones de diseño y la forma de aplicarlos correctamente.

A nivel personal, me llena de satisfacción haber sido capaz de llevar a cabo este proyecto. Ha sido muy útil para ampliar mis conocimientos sobre las tecnologías y el desarrollo en dispositivos móviles, sirviendo también para reforzar los conocimientos que ya poseía anteriormente.

El futuro de Cultimate es muy prometedor. Tras terminar el proyecto, la idea es ampliar la aplicación para que esta tenga más alcance, aumentando el número de usuarios y por tanto el número de ingresos.

No sería mala idea tampoco ofrecer la aplicación en otros idiomas, mejorando la accesibilidad y la base de usuarios.

La idea es que Cultimate sea una plataforma intuitiva y sencilla de utilizar, que ayude a la gente a mejorar su estilo de vida y a cuidar del planeta. Por tanto, sería muy positivo ampliar el catálogo de cultivos que ofrecemos, dando soporte a plantas autóctonas de cada región.

Es por ello por lo que esto es solo el comienzo en el ciclo de vida de esta aplicación y nos gustaría que se tuviese en cuenta como opción a la hora de intentar crear un mundo mejor.

9. Referencias

- [1] *Informe Pew Research Center's Spring 2018 Global Attitudes Survey* [En línea]. Consultado el 19 de Julio de 2024. https://www.pewglobal.org/wp-content/uploads/sites/2/2019/02/Pew-Research-Center_Global-Threats-2018-Report_2019-02-10.pdf
- [2] SWOT Analysis, Tanya Sammut-Bonnici. [En línea]. Consultado el 26 de julio de 2024. <https://kobbytamakloe.wordpress.com/wp-content/uploads/2020/07/swot-analysis-kjm.pdf>
- [3] *Top features of React Native* [En línea]. Consultado el 22 de julio de 2024. Disponible en: <https://medium.com/@anuj.tomar11/top-features-of-react-native-app-development-355eff652c00>
- [4] *Expo vs React Native Cli* [En línea]. Consultado el 22 de julio de 2024. Disponible en: <https://medium.com/@softworthsolutionspvtltd/expo-vs-react-native-cli-7e47c7630039>
- [5] *Why Visual Studio Code?* [En línea]. Consultado el 22 de julio de 2024. Disponible en: <https://code.visualstudio.com/docs/editor/whyvscode>
- [6] Ries, Eric, 1978, *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*. New York, Crown Business, 2011.
- [7] *Agile Manifesto* [En línea]. Consultado el 18 de julio de 2024. Disponible en: <https://agilemanifesto.org>
- [8] *Infografía Metodología Agile. Figura en línea*, consultada el 22 de julio de 2024. Disponible en: https://www.researchgate.net/figure/Agile-Methodology-in-System-Development-source-Okeke2021-retrieved-from_fig1_354310848
- [9] *Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales* [En línea]. Consultado el 24 de julio de 2024. Disponible en: <https://www.boe.es/buscar/act.php?id=BOE-A-2018-16673>
- [10] *Refactoring Improving the Design of Existing Code*, Martin Fowler with Kent Beck. 2018.
- [11] COLLARD, J.F, BURNSTEIN, I, *Practical Software Testing: A Process-Oriented Approach*, Springer. 2003.
- [12] *Plantilla diagrama de línea temporal*. [En línea]. Consultado el 22 de julio de 2024. Disponible en: <https://moqups.com/es/templates/charts-graphs/infographics/timeline-diagram/>
- [13] *Plantillas de Lean Canvas* [En línea]. Consultado el 20 de julio de 2024. Disponible en: <https://edit.org/es/blog/plantillas-lean-canvas>

Anexo: Relación con los ODS

Introducción a los ODS.

Los Objetivos de Desarrollo Sostenible (ODS) son un conjunto de 17 metas globales establecidas por las Naciones Unidas para abordar desafíos mundiales y promover un futuro más sostenible para todos para el año 2030. Estos objetivos van desde la erradicación de la pobreza y el hambre hasta la igualdad de género, la acción climática y la promoción de la paz y la justicia.

La relevancia de los ODS radica en su enfoque integral, que abarca aspectos sociales, económicos y ambientales. Permiten a gobiernos, organizaciones, empresas y la sociedad civil trabajar juntos en la búsqueda de soluciones que beneficien a las personas y al planeta en su conjunto.

La relación entre los ODS y el desarrollo de software es significativa. Las tecnologías de la información y la comunicación desempeñan un papel crucial en el logro de estos objetivos. El desarrollo de aplicaciones, plataformas y sistemas puede contribuir directamente a metas como la educación de calidad, la atención médica accesible, la gestión sostenible de recursos, la inclusión financiera y la innovación en sectores clave.

El software puede ser una herramienta poderosa para recopilar datos, analizar información, mejorar la eficiencia en la implementación de proyectos y facilitar la colaboración entre diversas partes interesadas en la consecución de los ODS.

Además, el enfoque en la sostenibilidad también se ha extendido al desarrollo de software mismo, promoviendo prácticas de codificación sostenible, diseño centrado en la eficiencia energética y la reducción de residuos digitales.

Descripción de la aplicación.

CultiMate es una aplicación móvil diseñada para ofrecer a los usuarios una guía detallada sobre el cultivo de plantas en entornos domésticos, balcones y jardines. Proporciona información acerca de diversos tipos de plantas, desde frutas hasta vegetales y hierbas, permitiendo que usuarios de un rango de edad variado, a través de una interfaz sencilla, aprendan los fundamentos básicos para cultivar en casa. El objetivo principal es fomentar el cuidado del planeta y la salud de las personas. Este objetivo es el enfoque principal para guiar el valor añadido de nuestra aplicación.

Los usuarios interactúan con la aplicación cuando desean comenzar el plantado de un cultivo. En la app podrán acceder a una lista de plantas, donde se les presentará información útil sobre cada planta, así como un calendario con fechas recomendadas para su siembra. Cuando un usuario selecciona una planta, esta se añade a un huerto simulado, proporcionando una forma más amigable de mostrar las plantas que están siendo cultivadas en ese momento. Al seleccionar una planta en el huerto simulado, el usuario podrá visualizar una guía de siembra que le proporcionará una serie de pasos para el cultivo y cuidado, actualizándose según el estadio de siembra en el que se encuentre el usuario. La aplicación ayuda al usuario recordando cuándo debe regar sus cultivos y que deben hacer en cada momento del plantado.



Por otro lado, la aplicación cuenta con una tienda donde los usuarios pueden comprar una variedad de productos relacionados con el cultivo. Y también dispondrá de un foro donde los usuarios pueden interactuar con otros para resolver dudas.

Relación con los ODS.

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
ODS 1. Fin de la pobreza.				X
ODS 2. Hambre cero.				X
ODS 3. Salud y bienestar.	X			
ODS 4. Educación de calidad.		X		
ODS 5. Igualdad de género.				X
ODS 6. Agua limpia y saneamiento.			X	
ODS 7. Energía asequible y no contaminante.				X
ODS 8. Trabajo decente y crecimiento económico.				X
ODS 9. Industria, innovación e infraestructuras.				X
ODS 10. Reducción de las desigualdades.				X
ODS 11. Ciudades y comunidades sostenibles.	X			
ODS 12. Producción y consumo responsables.	X			
ODS 13. Acción por el clima.		X		
ODS 14. Vida submarina.				X
ODS 15. Vida de ecosistemas terrestres.		X		
ODS 16. Paz, justicia e instituciones sólidas.				X
ODS 17. Alianzas para lograr objetivos.			X	

Tabla 3: Relación de Cultimate con los ODS

ODS 1: Fin de la pobreza.

CultiMate no se relaciona directamente con este objetivo, ya que no aborda la pobreza a nivel macroeconómico ni proporciona medios directos para superar la pobreza.

ODS 2: Hambre cero.

La aplicación no tiene un impacto directo en la erradicación del hambre a gran escala, dado que su alcance se limita a la jardinería doméstica y no necesariamente al suministro de alimentos a nivel comunitario o regional.

ODS 3: Salud y bienestar.

CultiMate sí contribuye a este ODS al educar a los usuarios sobre cómo cultivar sus propios alimentos, lo que puede llevar a una dieta más saludable y a la mejora del bienestar físico y mental.

ODS 4: Educación de calidad.

Este ODS está relacionado indirectamente, ya que la aplicación puede ser una herramienta educativa para aprender sobre horticultura, pero no aborda la educación de calidad en un sentido amplio o formal.

ODS 5: Igualdad de género.

No hay una relación directa, ya que CultiMate no se enfoca específicamente en abordar la igualdad de género o en empoderar a las mujeres y las niñas.

ODS 6: Agua limpia y saneamiento.

Aunque CultiMate podría educar sobre prácticas de riego eficientes, no tiene un impacto directo en garantizar la disponibilidad de agua limpia y saneamiento para todos.

ODS 7: Energía asequible y no contaminante.

No hay una relación directa, ya que la aplicación no está vinculada a la promoción de energía sostenible.

ODS 8: Trabajo decente y crecimiento económico.

CultiMate no tiene una conexión directa con este objetivo, ya que no se centra en la creación de empleo, condiciones laborales, ni crecimiento económico.

ODS 9: Industria, innovación e infraestructura.

Aunque la aplicación es un producto de la innovación, no está directamente relacionada con la construcción de infraestructura resiliente o la promoción de industrialización sostenible.

ODS 10: Reducción de las desigualdades.

No existe una conexión directa; CultiMate no aborda la reducción de desigualdades más allá de la posibilidad de proporcionar conocimientos sobre jardinería a un público amplio.

ODS 11: Ciudades y comunidades sostenibles.

Aquí sí hay una relación directa. CultiMate fomenta el cultivo de alimentos en el hogar, lo que puede contribuir a la sostenibilidad de las ciudades al promover espacios verdes urbanos y la autosuficiencia alimentaria.

ODS 12: Producción y consumo responsables.

Este ODS está directamente relacionado, ya que la aplicación educa sobre el cultivo sostenible y puede influir en los hábitos de consumo de los usuarios hacia prácticas más responsables.

ODS 13: Acción por el clima.

Hay una conexión indirecta, puesto que las prácticas de jardinería promovidas por CultiMate pueden tener un pequeño impacto positivo en el clima, pero la aplicación no se enfoca en la acción climática a gran escala.

ODS 14: Vida submarina.

No hay una relación directa, ya que CultiMate no aborda cuestiones relacionadas con la conservación de la vida submarina ni la explotación sostenible de los océanos.

ODS 15: Vida de ecosistemas terrestres.

CultiMate podría tener un impacto indirecto al promover la biodiversidad a través del cultivo de diferentes tipos de plantas, pero no está específicamente orientada a la protección de ecosistemas terrestres en un sentido amplio.

ODS 16: Paz, justicia e instituciones sólidas.

No hay una conexión directa, ya que CultiMate no aborda temas de paz, justicia o construcción de instituciones.

ODS 17: Alianzas para lograr objetivos.

La aplicación podría servir indirectamente a este ODS si promueve alianzas con otras entidades para fomentar el desarrollo sostenible, pero no es su enfoque principal.