



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

APLICACIÓN DE GESTIÓN SEMIAUTOMÁTICA DE CULTIVOS URBANOS

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN COMPUTACIÓN

RODRIGO DAVID CLAUDIO SORIA RAMÍREZ

PROFESOR GUÍA:
SERGIO OCHOA DELORENZI

MIEMBROS DE LA COMISIÓN:
BARBARA POBLETE LABRA
EDUARDO GODOY VEGA

SANTIAGO DE CHILE
2021

Resumen

Los cultivos urbanos son una actividad que va en aumento en las grandes urbes, estos son realizados principalmente dentro de las viviendas de los cultivadores, debido principalmente por la escasa disponibilidad de áreas verdes tanto públicas como privadas. Es por ello que realizar esta actividad puede tornarse problemática pues, se suma a los quehaceres diarios del cultivador, que además debe mantener un monitoreo constante en el lugar de su cultivo. Para resolver este problema se diseñó un sistema *IoT*, el cual se compone por una aplicación móvil y un sistema de control. En esta memoria se llevó a cabo la aplicación móvil. Por otra parte, en la memoria del alumno Benjamín Zamora se trató sobre el sistema de control.

La aplicación móvil desarrollada permite el control y monitoreo de un cultivo mediante un sistema de control. Ésta cuenta con módulos para la gestión de ambientes para el cultivo, que están compuestos por una cantidad determinada de macetas, variedades de plantas, sensores y actuadores. Los sensores permiten obtener una amplia variedad de datos tanto del ambiente, como de las macetas del cultivo, que pueden ser visualizados en gráficos desde la aplicación, y alertan al usuario frente a eventos que pudiesen afectarlo. Los actuadores son los encargados de mantener las condiciones ambientales óptimas para el cultivo, así como también el riego de las macetas. También se puede visualizar el historial de funcionamiento éstos, y modificar su modo de operación. La aplicación además cuenta con indicadores que permiten al usuario conocer rápidamente el estado de sus ambientes y sus respectivas macetas, sensores y actuadores.

La aplicación también posee una sección de plantas que permite al usuario conocer nuevas variedades de plantas a través de un catálogo y así poder especificar que especies fueron ubicadas en cada maceta. También se tiene una sección de comunidades para fomentar la interacción entre cultivadores que a futuro se espera promueva la interacción entre los usuarios para que estos puedan compartir conocimientos o plantas inclusive.

Esta aplicación fue evaluada para medir su usabilidad y utilidad. Para esta se reclutaron ocho participantes, que individualmente recibieron una breve introducción de la aplicación. Luego realizaron un sketch de operaciones y una encuesta. Para elaborar esta encuesta se usó la herramienta System Usability Scale(SUS), como medio para medir la usabilidad, esta se compone por una encuesta de diez preguntas que son respondidas según la escala de Likert; a esta se le añadieron tres preguntas más un comentario abierto para medir la utilidad de la aplicación. Los resultados de la evaluación fueron muy positivos y la aplicación logró un nivel de usabilidad “excelente” y una utilidad “buena”. No obstante surgieron varios servicios deseables sugeridos por los participantes que podrían ser incluidos en la aplicación.

Dedico esta memoria a mis padres, quienes tenían el anhelo de ver a sus hijos convertirse en profesionales.

Agradecimientos

Agradezco a todas las personas que me apoyaron en esta carrera, sobre todo en esta última etapa. También quisiera incluir a aquellos que ya no están aquí de manera física, pues la perseverancia y determinación que poseo y que me permitieron llegar hasta el final, de alguna u otra manera es un legado de ellos. Gracias, totales...

Tabla de Contenido

1. Introducción	1
1.1. Motivación	1
1.2. Contexto y Problema a Abordar	1
1.3. Objetivos de la Memoria	2
1.4. Estructura de la Solución	3
1.4.1. Tareas realizadas y división del trabajo	5
1.5. Estructura del Documento	6
2. Estado del Arte	7
2.1. Aplicaciones Móviles de Apoyo al Cultivador	7
2.2. Frameworks Multi-Plataforma	8
2.3. Plataformas <i>IoT</i>	9
2.4. Dispositivos de Apoyo al Cultivo Urbano	10
2.5. Discusión	11
3. Concepción de la solución	13
3.1. Perfiles de Usuario del Sistema	13
3.2. Principales Requisitos	13
3.3. Proceso a Apoyar	14
3.4. Arquitectura del Sistema	17
3.5. Modelo de Datos del Sistema	21
4. Implementación del Sistema de Gestión de Información de Cultivos	30
4.1. Inicio	31
4.2. Ambientes	32
4.3. Macetas	39
4.4. Plantas	42
4.5. Comunidades	43
5. Evaluación de la Solución	45
5.1. Participantes	45
5.2. Descripción del Proceso de Evaluación Realizado	45
5.3. Instrumento de Evaluación de la Usabilidad Percibida	46
5.4. Instrumento de Evaluación de la Utilidad Percibida	47
5.5. Resultados obtenidos	48
5.5.1. Tiempos y correctitud de las operaciones realizadas	48

5.5.2. Usabilidad percibida	49
5.5.3. Utilidad percibida	50
5.5.4. Servicios sugeridos	51
6. Conclusiones y Trabajo a Futuro	52
Bibliografía	54

Índice de Tablas

5.1. Tiempos de ejecución de las operaciones del sketch.	49
5.2. Correctitud de los participantes en las operaciones del sketch.	49
5.3. Resultados de la evaluación de usabilidad de la aplicación.	50
5.4. Usabilidad percibida por cada participante.	50
5.5. Resultados de la evaluación de utilidad de la aplicación.	50

Índice de Ilustraciones

1.1. Esquema de la arquitectura general del sistema de gestión semiautomática de cultivos en ambientes interiores.	3
1.2. Esquema de interacción entre ambiente interior con sensores, dispositivos y cámara.	4
3.1. Diagrama de flujo para proceso de agregar ambiente.	14
3.2. Diagrama de flujo para proceso de agregar maceta a ambiente.	15
3.3. Diagrama de flujo para proceso de agregar planta.	16
3.4. Diagrama de flujo para proceso de agregar comunidad.	16
3.5. Diagrama de contexto para el sistema según el modelo C4.	17
3.6. Diagrama de contenedor para la aplicación móvil según el modelo C4.	18
3.7. Diagrama de contenedor para el Servicio en la nube según el modelo C4.	19
3.8. Diagrama de componente para Complementos según el modelo C4.	20
3.9. Diagrama de componente para la aplicación móvil según el modelo C4.	21
3.10. Modelo de datos relacional.	22
4.1. Mapa de navegación para visitar las principales secciones del sistema.	31
4.2. Interfaz Inicio.	31
4.3. Interfaz Ambientes.	32
4.4. Mapa de navegación para agregar un Ambiente.	32
4.5. Interfaz final agregar Ambiente (continuación de la anterior).	33
4.6. Interfaz inicial agregar Ambiente.	33
4.7. Interfaz agrega Maceta a Ambiente.	34
4.8. Interfaz agrega Actuador a Ambiente.	34
4.9. Mapa de navegación de un Ambiente.	36
4.10. Interfaz detalle de Sensor en Ambiente.	37
4.11. Interfaz detalle de Ambiente.	37
4.12. Interfaz detalle de Maceta en Ambiente.	38
4.13. Interfaz detalle de Actuador en Ambiente.	38
4.14. Interfaz Macetas.	40
4.15. Mapa de navegación para agregar una Maceta.	40
4.16. Interfaz agrega Planta a Maceta.	41
4.17. Mapa de navegación de una Maceta.	42
4.18. Interfaz detalle de Planta en Maceta.	43
4.19. Interfaz Plantas.	43
4.20. Mapa de navegación para agregar Planta a Maceta.	44

4.21. Interfaz del Catálogo de Plantas.	44
4.22. Interfaz de Comunidades.	44

Capítulo 1

Introducción

1.1. Motivación

La motivación de este trabajo de memoria proviene del interés del estudiante por el desarrollo de sistemas que incorporen el paradigma de Internet de las Cosas (*IoT*), como apoyo al cultivo de plantas en zonas urbanas. Esta memoria nace a partir del interés de desarrollar un sistema *IoT* que ha futuro pueda ser un producto con potencial comercial. En esta labor este memorista trabajó en conjunto con el Sr. Benjamín Zamora, compañero de carrera y también memorista, quién está desarrollando un sistema complementario al descrito en este documento.

En los cultivos urbanos se ha identificado la necesidad de contar con un sistema que ayude a lograr cultivos exitosos, que apoye a usuarios inexpertos en esta materia, y también minimice el tiempo requerido para llevar a cabo esta actividad. Este tipo de ocupación se ha visto en aumento en los últimos años, ya que gran parte de la población está retomando la tradición de cultivar sus alimentos (hortalizas principalmente) y plantas medicinales en el lugar donde viven [25], siendo una alternativa a los productos que son comercializados en el mercado. Además, el cultivo urbano permite al consumidor mantener un control de los pesticidas usados (en términos de calidad y cantidad), brindando acceso a productos más limpios, a un menor costo y con mayor variedad, particularmente cuando lo cosechado no tiene una amplia oferta en el mercado [38, 37].

1.2. Contexto y Problema a Abordar

En esta memoria se desarrolló un sistema *IoT* de apoyo al cultivo urbano (por ejemplo, en balcones, patios o al interior del hogar), el cual a grandes rasgos estará compuesto por un sistema de control (que monitorea y controla la humedad y temperatura del cultivo), y una aplicación móvil que permite al usuario comunicarse con éste. Particularmente se desarrolló una aplicación de software, mientras que el sistema de control fue realizado en la memoria del Sr. Benjamín Zamora (reportado en [54]).

Para entender qué apoyo brinda este sistema al cultivador, y en qué tipo de cultivos puede

utilizarse, es necesario antes definir el término “cultivo urbano”. Éste se refiere a toda siembra o plantación que tiene lugar en zonas urbanas, o relacionadas con éstas. Usualmente esto se da donde las áreas verdes son escasas, o en algunos casos inexistentes [50, 28].

Este tipo de cultivos generalmente se sitúa en terrazas, balcones, jardines y en ambientes interiores (por ejemplo, en una habitación). Esta labor implica nuevas tareas dentro del quehacer diario de las personas, y requiere seguir una rutina para obtener un resultado exitoso; es decir, lograr completar el ciclo de cultivo de manera satisfactoria, así como también de producción esperada. Es por esto que contar con una herramienta que minimice el tiempo de trabajo, y que también garantice un resultado óptimo, se vuelve un recurso muy relevante y valioso para el cultivador [33, 44].

Las problemáticas más comunes asociadas a esta actividad son principalmente la falta de tiempo para atender el cultivo, carencia de conocimientos, la poca conscientización del cuidador acerca del estado actual del cultivo, la falta de acceso al cultivo en el momento que es requerido, y el manejo ineficiente de recursos (por ejemplo, el agua). Todo esto impacta negativamente en los resultados del proceso de cultivo.

Hoy en día esta actividad se realiza principalmente de forma personal, dependiendo de la voluntad y memoria del cultivador para sus cuidados, pues no existe una conexión entre éste y el cultivo, que le permita tener registro sobre todos los eventos importantes o recibir alertas sobre posibles contingencias que afecten a éste. Particularmente, en el cultivo urbano el riego es una actividad fundamental que debe ser mantenida bajo control, así como también la exposición a la luz y a la temperatura. Dependiendo del tipo de cultivo, hay variables adicionales que requieren ser monitoreadas y controladas. Cuando los valores de esas variables salen de los rangos aceptables, se requiere que alguien (por ejemplo, el cuidador del cultivo) realice una acción correctiva para normalizar la situación. Para enfrentar este tipo de situaciones la gestión remota del ciclo de cultivo y la automatización de parte de dicha actividad, cobra gran importancia [36, 41].

1.3. Objetivos de la Memoria

El objetivo general de este trabajo de memoria es desarrollar una aplicación móvil que permita mantener una conexión entre el cultivador y su cultivo, y así mantener el proceso de cultivo bajo control, minimizando el tiempo destinado a tareas repetitivas, permitiendo al cultivador dedicar más tiempo a labores indispensables o de mayor prioridad [42, 46]. Para alcanzar el objetivo general, se han definido los siguientes objetivos específicos:

1. Desarrollar un módulo de gestión de cultivos urbanos (creación, modificación y eliminación de cultivos), que permita además acceder a un catálogo de plantas y a las comunidades de cultivadores urbanos.
2. Desarrollar un módulo de monitoreo automático de variables medioambientales del cultivo. Este módulo también incluye la consulta a demanda de estas variables, y del estado general del cultivo, así como el envío de notificaciones automáticas al cuidador cuando corresponda.
3. Desarrollar una aplicación móvil que integre los componentes anteriores, y que genere reportes según un periodo establecido. Estos reportes le permitirán al cuidador generar

nuevas configuraciones para los equipos usados en el ambiente de cultivo, y además realizar una gestión más eficiente de los recursos hídricos, energéticos y de trabajo presencial.

4. Establecer e implementar la comunicación entre la aplicación móvil y el sistema de control desarrollado por el otro memorista que participa en este proyecto.

1.4. Estructura de la Solución

A continuación se presentan dos figuras que ilustran la estructura de la solución, bajo el escenario de un cultivo urbano en ambiente interior (indoor). La Figura 1.1 muestra un esquema del sistema completo, desde la aplicación móvil hasta el ambiente de cultivo. En ésta se observa una arquitectura típica de un sistema *IoT*, el cual consiste en cuatro componentes principales: hardware, software, nube y conectividad [53].

La Figura 1.2 presenta un esquema de interacción entre el ambiente de cultivo y los sensores, actuadores y dispositivos de control. En ambas figuras se añadieron números junto a cada elemento para tener una referencia en su posterior explicación.

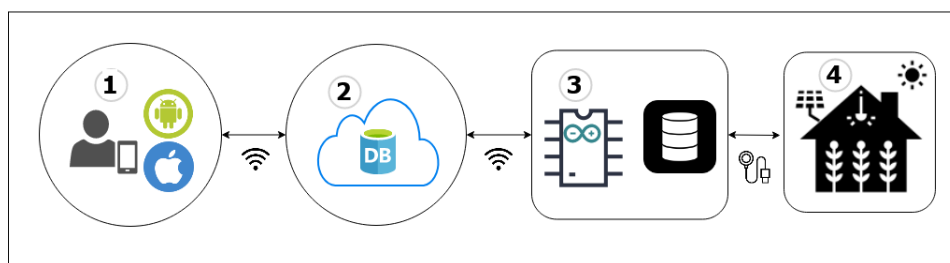


Figura 1.1: Esquema de la arquitectura general del sistema de gestión semiautomática de cultivos en ambientes interiores.

En la Figura 1.1, las flechas indican comunicación bidireccional entre los elementos dentro de cada componente. Estos componentes se describen a continuación:

1. *Aplicación móvil* disponible para Android e iOS, que mantiene comunicación bidireccional con la nube a través de Internet.
2. *Servicio en la nube* conocido como plataforma *IoT* [43], el cual es la base para que dispositivos estén interconectados y se genere un ecosistema propio. Dicho de otra forma, este servicio en la nube es el software que conecta hardware, puntos de acceso y redes de datos, con la aplicación móvil que utiliza el cultivador.
3. *Hardware del sistema* que se conecta con el servicio en la nube, para almacenar datos y recibir órdenes de control desde la aplicación móvil. El hardware incluye un controlador, el cual es el encargado de manejar los sensores y actuadores del sistema. Un sensor es un dispositivo que detecta un cambio físico o químico en el entorno y responde con un voltaje analógico medible por el controlador [24]. Por otro lado, un actuador es básicamente un dispositivo que recibe un señal, y en base a esta, ejecuta una acción con la finalidad de generar un efecto sobre un proceso automatizado [16].

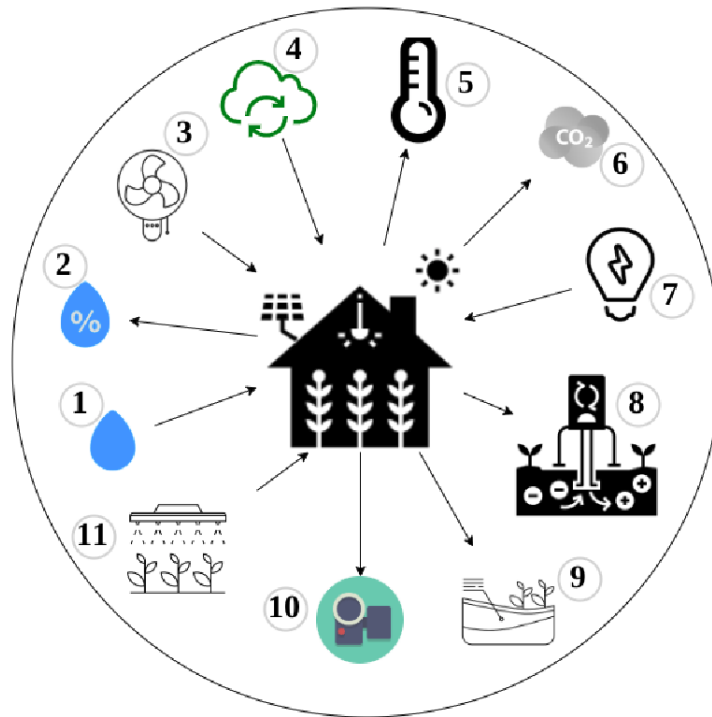


Figura 1.2: Esquema de interacción entre ambiente interior con sensores, dispositivos y cámara.

4. *Cultivo en ambiente interior*, en éste se desplegarán sensores y módulos que controlan los actuadores utilizados en el ciclo de cultivo. El detalle de estos se presentan en la Figura 1.2.

La simbología de la Figura 1.2 indica lo siguiente; si la flecha sale desde el ambiente de cultivo, entonces el símbolo referenciado es un sensor; en otro caso, es un actuador. Estos símbolos se describen a continuación:

1. *Humidificador*. Este dispositivo se encarga de elevar la humedad del aire en el ambiente de cultivo. Su funcionamiento se activa mediante un módulo de relé que es manejado por el controlador del sistema IoT.
2. *Sensor de humedad relativa del aire*. Este sensor mide la humedad del aire e informa sus mediciones al controlador.
3. *Ventilador*. Su función es hacer circular el aire dentro del ambiente; éste es activado también con un módulo de relé.
4. *Extractores e inyectores de aire*. Estos tienen la tarea de hacer circular y renovar el aire del ambiente de cultivo. También son activados con relés.
5. *Sensor de temperatura del aire*. Este mide la temperatura del aire en su posición, y envía sus lecturas al controlador.
6. *Sensor de concentración de dióxido de carbono*. Mide la concentración de este gas, pues en un agente importante en el proceso de fotosíntesis de las plantas. Este sensor también informa sus mediciones al controlador.
7. *Luces*. Estos dispositivos son esenciales para la iluminación artificial que reemplaza la acción del Sol. Son accionados también por el microcontrolador en ciclos ajustados al

cultivo.

8. *Sensor de electro-conductividad del sustrato*. Éste mide un parámetro importante, ya que la electro-conductividad influye en la tasa de absorción de nutrientes de las plantas. También proporciona sus lecturas al controlador.
9. *Sensor de humedad relativa del sustrato*. Éste también proporciona mediciones de humedad relativa para el controlador.
10. *Cámara*. La cámara permite visualizar el cultivo en todo momento desde la aplicación móvil.
11. *Electroválvula*. Este dispositivo permite abrir o cerrar una válvula de agua para activar el riego automático. Ésta se acciona mediante un relé.

1.4.1. Tareas realizadas y división del trabajo

Para crear este sistema de apoyo se realizaron una amplia variedad de tareas, las cuales se detallan a continuación. Para una explicación ordenada de éstas, se utilizará como referencia los números de componente mostrados en la Figura 1.1. En adelante se referirá a cada figura del esquema como una componente, siendo cuatro en total. Además, al final del detalle de cada tarea se menciona a cuál de los memoristas le correspondió llevarla a cabo (Rodrigo Soria o Benjamín Zamora).

- Tareas de la Componente 1:
 - Crear modelo de datos de la aplicación móvil. Tarea de Rodrigo Soria.
 - Diseñar interfaces de la aplicación. Tarea de Rodrigo Soria.
 - Programar la aplicación móvil. Tarea de Rodrigo Soria.
 - Conectar la aplicación con el servicio en la nube mediante una API. Tarea de Rodrigo Soria.
- Tareas de la Componente 2:
 - Crear API para comunicar la nube con la aplicación. Tarea de Rodrigo Soria.
 - Crear API para comunicar la nube el hardware del sistema de control. Tarea de Rodrigo Soria.
 - Establecer modelos de datos que incorpora los modelos de la aplicación y del hardware del sistema de control. Tarea de Rodrigo Soria.
- Tareas de la Componente 3:
 - Definir modelo de datos para el sistema de control. Tarea de Benjamín Zamora.
 - Programar el sistema de control utilizando lógica difusa. Tarea de Benjamín Zamora.
 - Crear API para comunicar el sistema de control con la nube. Tarea de Benjamín Zamora.
- Tareas de la Componente 4:
 - Establecer conexión entre el controlador y los sensores en conjunto con los módulo de relés. Tarea de Benjamín Zamora.
 - Programar la recepción de los datos recopilados por los sensores. Tarea de Benjamín Zamora.
 - Programar la activación de los módulos de relés. Tarea de Benjamín Zamora.

1.5. Estructura del Documento

A continuación se listan los capítulos de esta memoria y se hace una breve referencia al contenido de los mismos.

- Capítulo 1: “Introducción”. En este capítulo se muestra una mirada general del presente trabajo y su motivación. Además, se expone el contexto del problema a abordar, y los objetivos del trabajo a realizar.
- Capítulo 2: “Estado del Arte”. En éste se presenta la situación actual de sistemas *IoT* que se asemejan, o tienen algunas funciones similares a la solución propuesta en esta memoria. También se revisan plataformas *IoT* para la implementación del backend de la aplicación, y frameworks para el frontend de la aplicación. Además, se plantean los desafíos que implica hacer un sistema *IoT* de estas características.
- Capítulo 3: “Concepción de la Solución”. En este capítulo se detalla cómo se concretó la solución obtenida, y para ello se muestran los siguientes aspectos:
 - Identificación de los perfiles de usuario del sistema.
 - Principales requisitos de usuario.
 - Diseño de la aplicación.
 - Procesos apoyados por el sistema.
 - Arquitectura del sistema.
 - Modelo de datos del sistema.
- Capítulo 4: “Implementación del Sistema de Gestión de Información de Cultivos”. Aquí se presentan las interfaces, el funcionamiento de la aplicación y sus características.
- Capítulo 5: “Evaluación del Sistema”. Se muestra cómo se evaluó la funcionalidad y usabilidad del sistema desarrollado. Además, se presentan y discuten los resultados obtenidos en esta actividad.
- Capítulo 6: “Conclusiones y Trabajo a Futuro”. En este capítulo se entrega un resumen de la memoria, incluyendo el trabajo realizado, la evaluación de la solución y los resultados obtenidos. Allí se muestra que los objetivos planteados inicialmente fueron alcanzados. Finalmente, se presenta el trabajo a futuro que se abordará en la siguiente versión de la aplicación.

Capítulo 2

Estado del Arte

En este capítulo se presentan sistemas y tecnologías que pretenden dar al lector una visión general acerca del estado del arte de las soluciones existentes, las cuales resuelven algunos aspectos del problema expuesto. Particularmente se revisarán una muestra de las aplicaciones móviles de apoyo al cultivador, los frameworks multi-plataforma para desarrollar estos sistemas, las plataformas *IoT* para este tipo de soluciones, y los dispositivos de apoyo al cultivo urbano. Finalmente, se discuten los desafíos técnicos que usualmente se deben abordar en el desarrollo de este tipo de soluciones.

2.1. Aplicaciones Móviles de Apoyo al Cultivador

La situación actual de las aplicaciones móviles para el apoyo al cultivador es bastante diversa, y se encuentra viviendo una gran evolución debido al crecimiento de Internet y a la irrupción de la quinta generación de redes móviles, más conocida como el *5G*. Estas redes móviles permiten navegar a velocidades del orden de 10 GBps (gigabytes por segundo) y reducir significativamente la latencia (el tiempo de respuesta de la red) hasta los 5 milisegundos, lo cual es imperceptible para los humanos [35]. Estas nueva tecnologías permiten además aumentar exponencialmente el número de dispositivos conectados a la red, lo que minimiza las limitaciones para agregar dispositivos a una solución de IoT por problemas de direccionamiento.

Debido a esto, y a la diversificación de los componentes con capacidad de cómputo, los sistemas *IoT* cuentan ya con la infraestructura requerida para desarrollar herramientas confiables e indispensables para apoyar esta actividad [39, 32]. Es por esto que las aplicaciones móviles pensadas para este propósito han ido incorporando cada vez más capacidades.

Las primeras aplicaciones móviles de apoyo al cultivo urbano contaban con funciones sólo del tipo enciclopedia de plantas, y bitácoras para registrar eventos del cultivo. Las aplicaciones actuales permiten controlar y monitorizar el cultivo, tanto in-situ como de forma remota. A continuación se mencionan algunos ejemplos destacados de aplicaciones de apoyo al cultivo urbano, seguido de una breve descripción de cada iniciativa:

- **LEAF** [10], **Grobo** [6] y **Gro.io** [7] son sistemas de cultivo hidropónicos automá-

ticos controlados por *smartphones* para plantas medicinales y de alimento. Desde la aplicación móvil que incluye cada sistema, se realiza el control y monitoreo de todos los parámetros relevantes para el cultivo. Sobre la base de esta información, las aplicaciones muestran sugerencias y alertas sobre mejores configuraciones para el funcionamiento de los dispositivos del sistema, como por ejemplo, la potencia de las luces o la cantidad de agua a usar en el riego.

- **Grow with Jane [8]** es una herramienta que ayuda al cultivador a tener control desde la siembra de la semilla, hasta la cosecha de la planta. Para ello, este sistema implementa *logs*, recordatorios y sugerencias, basándose en la información registrada por el usuario.
- **Garden Answers [4]** es una aplicación que identifica plantas de manera instantánea. Para ello, sólo se necesita tomar una foto de la planta en cuestión con la aplicación.
- **myGrow [13]** es una herramienta enfocada en el cultivo de cannabis con fines medicinales. Para esta labor, la aplicación permite mantener un registro exhaustivo de principio a fin, y también entrega información sobre tipos cepas medicinales.
- **Niwa [14]** es un sistema *IoT* pensado para el cultivo de hortalizas en ambiente interior, que está compuesto por un equipo de cultivo y una aplicación móvil. A través de ésta, se monitoriza y configura los dispositivos del equipo de cultivo.
- **Growpal One [40]** es un sistema diseñado para ser usado en un ambiente interior, el cual automatiza el funcionamiento de los equipos conectados a él, según la configuración establecida por el usuario. Este sistema es aplicable a cultivos en huertos, maceteros, cultivos verticales y carpas *indoor*. Incluye sensores y un sistema de riego automático. Su diseño destaca por ser intuitivo y de fácil configuración mediante una pantalla táctil.

La aplicación móvil propuesta en esta memoria tendrá las siguientes ventajas sobre los ejemplos antes mencionados: su diseño que permite al usuario integrar al sistema sus equipos de cultivo y automatizar su funcionamiento, la capacidad de controlar y monitorear tantos cultivos como sistemas de control se tengan, entrega la posibilidad de gestionar tanto en ambientes exteriores como interiores e incluye un módulo de comunidades online de cultivadores que le facilitará al usuario la resolución preguntas. Para el desarrollo de esta aplicación, fue necesaria la selección de varias tecnologías de apoyo a la implementación del software, las que se presentan a continuación.

2.2. Frameworks Multi-Plataforma

Para llevar acabo este trabajo de memoria se debió escoger un framework para el desarrollo del frontend la aplicación. Además, para abarcar la mayor cantidad de plataformas posibles, se pretendía que éste fuera multi-plataforma. La ventaja que entrega un framework de ese tipo, consiste en que permite obtener aplicaciones nativas para distintas plataformas (como por ejemplo, Android e iOS, entre otros), a partir de un único código fuente [52].

A continuación se presentan algunos ejemplos de frameworks multi-plataforma, junto a una pequeña reseña [48]:

- **Flutter [3]** es un kit de herramientas UI de Google, para construir aplicaciones nativas, compiladas para plataforma web, móvil y de escritorio desde un mismo código base.

Flutter es un framework adecuado para desarrollar aplicaciones de alto rendimiento, ya que se basa en el Lenguaje Dart, el cual ofrece funciones de escritura y ejecución de código más rápidas para los desarrolladores. Algunos de los casos de uso más destacables de esta tecnología son Google Ads y Alibaba.

- **React Native [15]** es un Cross-platfrom framework, de código abierto, creado por la compañía Facebook. Este framework permite crear aplicaciones móviles nativas para las plataformas móviles Android e iOS. Es uno de los más usados en el último tiempo, se basa en Javascript y es una herramienta ideal para desarrollar aplicaciones de nivel empresarial. Casos de uso destacables son: Facebook e Instagram.
- **Ionic [9]** es un framework de código abierto, que ofrece una amplia biblioteca de herramientas y componentes UI, optimizadas para dispositivos móviles, con el objetivo de poder crear aplicaciones rápidas y altamente interactivas para Android e iOS usando Javascript. Casos de uso destacables son: Pacifica, Market Watch y McDonald's.
- **Xamarin [19]** es un framework multi-plataforma de Microsoft, de código abierto, que sirve para construir aplicaciones para Android e iOS, usando la plataforma .NET y el lenguaje de programación C#. La creación de aplicaciones de alto rendimiento con Xamarin es rápida, ya que tiene una función TestCloud que prueba automáticamente la aplicación, y garantiza la reutilización de código en su totalidad. Algunos de los casos de uso más destacables son los portales de las siguientes empresas: Alaska Airlines, Oro, MRW y AOX.
- **Cordova [27]** es un framework de código abierto, de propiedad de Adobe Systems, que permite crear aplicaciones para múltiples plataformas, mediante el uso de complementos y tecnologías web estándar como HTML5, CSS3 y Javascript. Casos de uso destacable de este framework son los portales de Hogangnono, Telpert y Walmart.

2.3. Plataformas *IoT*

Como parte de este trabajo también se debió escoger una plataforma *IoT*, que integrara y comunicara a la aplicación con el hardware seleccionado. La elección de la mejor plataforma de *IoT* depende de diversos factores como la escalabilidad, el protocolo de comunicación utilizado, el ancho de banda de los canales de comunicación, etc. Las plataformas más destacadas para estos fines son las siguientes:

- **Firestore [27]** es una plataforma de desarrollo de aplicaciones Backend-as-a-Service (BaaS) de Google, que entrega servicios de alojamiento de backend, como el caso de una base de datos en tiempo real, almacenamiento en la nube, y autenticación, entre otros.
- **Thingier.io [17]** es una plataforma que contiene librerías de código abierto, que permite gestionar una multitud de dispositivos a través de Internet. Esta aplicación permite almacenar y visualizar información recibida desde sensores, enviar información o instrucciones a dispositivos, e interactuar con otro tipo de aplicaciones, como por ejemplo, con IFTTT (if this, then that) services. Este último es un tipo de servicio web que permite crear y programar acciones para automatizar diferentes tareas y acciones en Internet.
- **DeviceHive [2]** es una plataforma de datos *IoT* de código abierto con una amplia

gama de posibilidades de integración con otras plataformas.

- **Mainflux [11]** es una plataforma *IoT* de código abierto segura y de alto rendimiento, con capacidades para el desarrollo a escala de soluciones, aplicaciones *IoT* y productos inteligentes conectados.
- **Thingspeak [18]** es otra plataforma *IoT* que permite recoger y almacenar datos de sensores en la nube, y desarrollar aplicaciones. También ofrece aplicaciones que permiten analizar y visualizar datos en MATLAB, entre otros.
- **Zetta [20]** es una plataforma de código abierto construida sobre *Node.js*, la cual permite crear servidores *IoT* que se ejecutan en computadoras distribuidas geográficamente y en la nube. Esta plataforma combina API REST, WebSockets y programación reactiva, perfecta para conectar muchos dispositivos en aplicaciones con datos en tiempo real.
- **AWS IoT Analytics [26]** es un servicio administrado, que facilita la ejecución e instrumentación del análisis de grandes volúmenes de datos de *IoT*. Este servicio permite al desarrollador despreocuparse del costo y la complejidad que involucra la creación de una plataforma de análisis de data de sistemas *IoT* (que usualmente son muy voluminosos).
- **Cayenne myDevices [1]** es una plataforma de prototipado de dispositivos de *IoT*. Una de sus mayores ventajas es que es una solución visual, que permite arrastrar y soltar *widgets*, es decir, permite crear y configurar el sistema sin tener la necesidad de programar. Sin embargo, como suele suceder con este tipo de herramientas, esto limita bastante el tipo de soluciones que se pueden construir, ya que solo se cuenta con las funcionalidades que la plataforma tiene disponible.

2.4. Dispositivos de Apoyo al Cultivo Urbano

En un cultivo urbano, los dispositivos usualmente utilizados para su instrumentación son de tres clases: controladores, actuadores y sensores. Estos se describen a continuación:

- **Controlador** es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Los controladores son ampliamente utilizados insertos en placas de desarrollo. Estos permiten ser conectados a sensores y actuadores de distintas maneras, con el objetivo de obtener los datos recogidos por un sensor, y en base a estos datos tomar decisiones sobre el funcionamiento de dispositivos manejados por un actuador. Las placas de desarrollo (controladores) más usadas para esta tarea son las siguientes:
 - **Arduino Uno [22]** es una plataforma de desarrollo basada en una placa electrónica de hardware libre, que incorpora un microcontrolador re-programable y una serie de pines hembra. Estos permiten establecer conexiones entre el microcontrolador y los diferentes sensores y actuadores de una manera muy sencilla (principalmente con cables dupont).
 - **Nodemcu [34]** es una placa de desarrollo totalmente abierta, a nivel de software y de hardware. Al igual que ocurre con Arduino, en NodeMCU todo está dispuesto para facilitar la programación de un microcontrolador o MCU (del inglés Microcontroller Unit).
 - **Raspberry Pi [23]** es una computadora de bajo costo, de tamaño compacto (del porte de una tarjeta de crédito), que puede ser conectada a un monitor de

computador o un TV, y usarse con un mouse y teclado estándar. Es un pequeño dispositivo capaz de permitirle a las personas de todas las edades explorar la computación y aprender a programar lenguajes como Scratch y Python. Es capaz de hacer la mayoría de las tareas típicas de un computador de escritorio, desde navegar en internet, reproducir videos en alta resolución, manipular documentos de ofimática, y hasta reproducir juegos.

- **Actuador** es básicamente un dispositivo que recibe una señal, y en base a ella realiza una acción. Un ejemplo típico de un actuador es el relé, el cual es un interruptor eléctrico que recibe una señal para cambiar su estado, el cual puede ser encendido o apagado. Este dispositivo es ampliamente utilizado, ya que permite automatizar el funcionamiento de cualquier aparato. Se denominará “dispositivo actuador” al aparato conectado a un relé, para un mayor entendimiento. Los dispositivos actuadores usualmente utilizados en un cultivo urbano son: luces(típicamente focos), humidificador, ventilador, extractor e inyector de aire, y electroválvulas para el riego.
- **Sensor** es un dispositivo compuesto por un transductor, capaz de detectar, medir o indicar los cambios que se producen en una magnitud física específica (por ejemplo, temperatura, humedad, presión, etc.) y transformarlos en una señal eléctrica, y un acondicionador de señal, que traduce a esa señal a un formato fácil de leer [21]. Los sensores típicamente usados en el cultivo son de temperatura, de humedad relativa, medición de gases, de agua y de luz.

Combinando las tres clases de dispositivos antes mencionadas es posible crear un sistema *IoT* de apoyo al cultivo urbano, donde los controladores tienen un rol fundamental, ya que permiten la programación necesaria tanto para recibir los datos de los sensores, como para controlar el funcionamiento de los dispositivos actuadores.

2.5. Discusión

Para la resolución del problema planteado, se requiere mantener (a grandes rasgos) una comunicación bidireccional entre el sistema y el usuario, una recopilación de datos constante para el monitoreo del cultivo, y automatizar los procesos de cultivo tanto como se pueda. Para ese escenario de operación, aparece como muy conveniente el desarrollo de una solución *IoT*, pensada para el cultivo en interior. Para esto, la solución debe adaptarse a las necesidades de estos usuarios, de los que se asume que cultivan en tierra (en macetas) y cuentan con un mínimo de equipamiento para realizar esta actividad, por ejemplo, luces, extractores, ventiladores, electroválvulas para el riego, etc. También se asume que estos usuarios cultivan en su gran mayoría en tierra por su facilidad, y no en hidroponía por ejemplo, debido a los altos costos en equipamiento que se requiere, y también a la gran limitación de no poder cultivar plantas del tipo tubérculo [51]. Teniendo en cuenta estas consideraciones, la solución tendrá que automatizar el funcionamiento de los equipos y entregar información útil al usuario para el control de estos.

Las aplicaciones antes presentadas tienen limitaciones para dar cumplimiento a los requerimientos enunciados para apoyar el cultivo urbano. Algunas de estas herramientas, como LEAF [10] o Grobo [6] son sistemas completos y de alto costo, lo que condiciona el cultivo según las prestaciones que cada uno de estos ofrece. La aplicación Growpal One [40] es una

de las más destacas por su enfoque complementario para la automatización del cultivo; sin embargo, no cuenta con una aplicación móvil que permita el monitoreo.

Dado lo antes expresado, existe una oportunidad para desarrollar una solución de bajo costo, pensada para apoyar el cultivo urbano de personas que no necesariamente tienen tiempo y/o conocimiento para realizar dicha actividad. En el diseño de dicha solución hay que analizar cuidadosamente las tecnologías que permiten el envío de datos, y el manejo de estos en la nube (donde se implementaría la lógica de negocio). Además, para la comunicación entre el usuario con el sistema es necesario el desarrollo de una herramienta que presente la información de manera visual, para facilitar su entendimiento. Para esto se escogió utilizar el smartphone, debido a que es un dispositivo portátil y masivo, para el que se desarrollará una aplicación móvil que servirá como conexión entre el usuario y el cultivo en cuestión. [45, 49].

Los desafíos que involucra desarrollar este sistema son los siguientes:

1. Integración entre el dispositivo móvil, el hardware del sistema y la nube.
2. Mantener un correcto control y funcionamiento de los dispositivos conectados al sistema.
3. Seleccionar formas de visualización de datos, de manera que sean entendibles y útiles para el usuario.
4. Establecer comunicación entre el usuario y el cultivo con baja latencia.
5. Proveer al usuario de reportes que le permitan entender cómo está el cultivo, y establecer mejores configuraciones de funcionamiento del sistema.
6. Alertar al usuario ante eventos que pudiesen afectar al cultivo.
7. Garantizar la seguridad, confidencialidad e integridad de los datos, lo cual siempre es un reto para sistemas *IoT* [47].

Capítulo 3

Concepción de la solución

En este capítulo se presenta la solución formulada para el problema antes presentado, y el contexto en el cual éste está inmerso. Para ello, se exponen a continuación diversos aspectos de la solución como: perfiles de usuario, principales requisitos, diagramas de flujo para los procesos a apoyar, arquitectura del sistema y el modelo de datos de la solución.

3.1. Perfiles de Usuario del Sistema

Para este sistema se identificaron tres tipos de usuario: “cultivador”, “miembro de una comunidad” y “público en general”. A continuación se describe a cada uno de estos:

- *Cultivador*: Este usuario utiliza la aplicación para controlar y monitorear su cultivo. Para esto, se debe registrar en la aplicación e ingresar los datos de los ambientes de cultivos que serán manejados por esa persona.
- *Miembro de una comunidad*: Este usuario crea comunidades o participa de una o más de ellas, a través del foro de la aplicación. Para ello, éste debe registrarse en la aplicación y postular a una comunidad, o crear una nueva.
- *Público*: Este usuario accede a la aplicación y solo puede acceder a la información pública del sistema.

3.2. Principales Requisitos

Los principales requisitos de usuario fueron definidos en base a lo discutido con el memorista Benjamín Zamora, el cuál por su trabajo es un conocedor de las necesidades y requerimientos de los cultivos urbanos, y también teniendo en consideración información encontrada en la literatura sobre cultivos. Los requisitos son los siguientes:

1. La aplicación debe exhibir sus características mediante una interfaz de bienvenida. Si el usuario desea acceder a alguna de las funciones mostradas, éste debe registrarse.
2. La aplicación debe permitir ingresar información del ambiente de cultivo donde se usará. Este menú permite crear y conectar ambientes, macetas y plantas, para su posterior

control y monitoreo.

3. La aplicación debe permitir crear y participar en comunidades online, las cuales tienen como objetivo compartir información referente al cultivo de una planta en particular.
4. La aplicación debe presentar al usuario visualizaciones de datos útiles para el cultivo, así como también el historial de funcionamiento de actuadores.
5. La aplicación debe generar notificaciones ante eventos ocurridos en el ambiente de cultivo.

3.3. Proceso a Apoyar

La aplicación apoyará los siguientes procesos de administración de: ambientes, macetas, plantas y comunidades. Para explicar estos procesos, se presentan a continuación diagramas de flujo que ilustran cómo agregar cada uno de estos componentes al sistema. No se muestra la modificación o eliminación de estos componentes, pues siguen un flujo muy similar al aquí descrito.

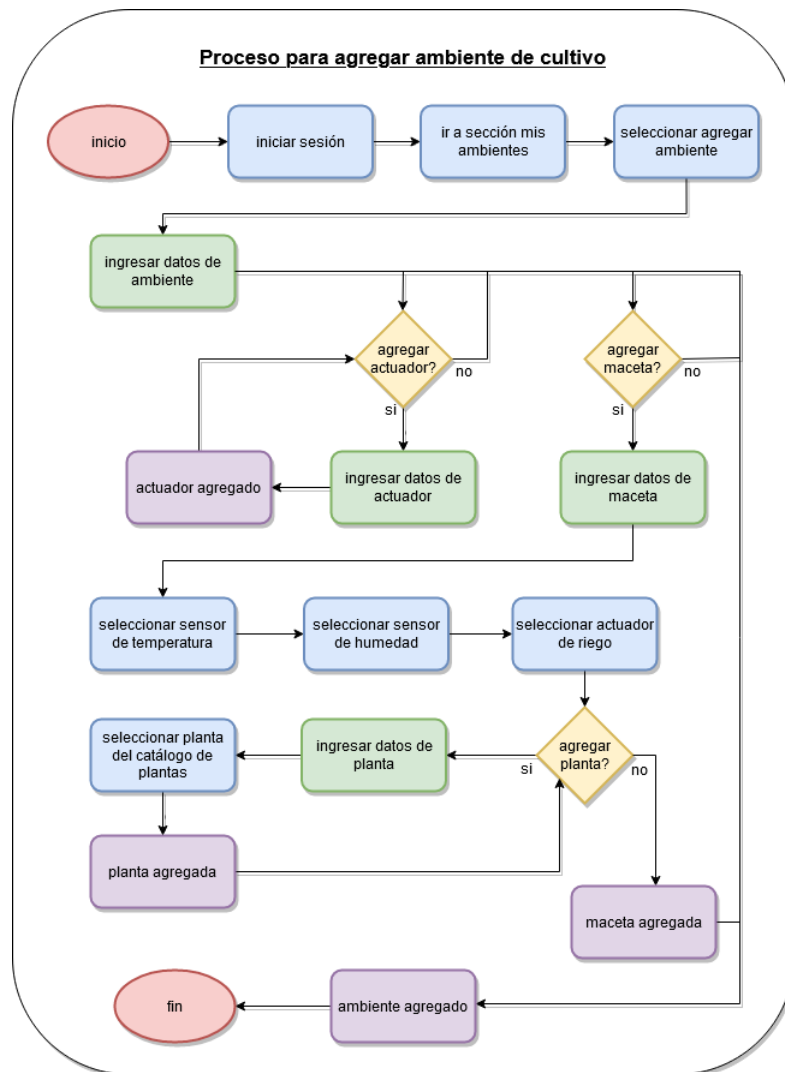


Figura 3.1: Diagrama de flujo para proceso de agregar ambiente.

El proceso de agregar ambiente de la Figura 3.1 comienza con el inicio de sesión del usuario, luego se dirige a la sección “mis ambientes” y selecciona “agregar ambiente”. Enseguida se requiere el ingreso de datos del ambiente, por ejemplo su nombre, dimensiones, si es un ambiente de cultivo en interior o exterior, un código alfanumérico para conectar la aplicación con el sistema de control (dispositivo compuesto por un controlador, sensores y actuadores). Sólo después de haber ingresado estos datos, se presentan al usuario las siguiente opciones:

- *Agregar actuador*: Esta acción requiere que el usuario ingrese datos del actuador, tales como su nombre, consumo, régimen de operación (manual, automático o diario), tipo de enchufe al cual ha sido conectado, y el tipo de actuador al que corresponde (por ejemplo: luz, ventilador, electroválvula, etc.). El sistema permite agregar múltiples actuadores para un ambiente.
- *Agregar maceta*: Esta acción solicita el ingreso de datos de la maceta, como la composición del sustrato y volumen de la maceta. Luego se presentan las siguientes opciones:

- *Seleccionar sensor de temperatura*: Esta acción habilita al sensor para comenzar a funcionar y recopilar datos en la maceta donde se medirá la temperatura del sustrato.
- *Seleccionar sensor de humedad*: Esta acción habilita al sensor para comenzar a funcionar y recopilar datos en la maceta donde se medirá la humedad relativa del sustrato.
- *Seleccionar actuador de riego*: Esta acción habilita al actuador de riego para comenzar a funcionar en la maceta.
- *Agregar planta*: Esta acción requiere ingresar datos de la planta, como la fecha de inicio de la fase de germinación, vegetación o floración. Para identificar la planta que se está agregando, ésta se debe buscar en el catálogo de plantas y seleccionarla. Una vez listo este último paso, la planta es agregada. Igual que en los casos anteriores, es posible añadir múltiples plantas a una maceta. En la Figura 3.3 también se explica el proceso de agregar una planta, pero con un flujo diferente.

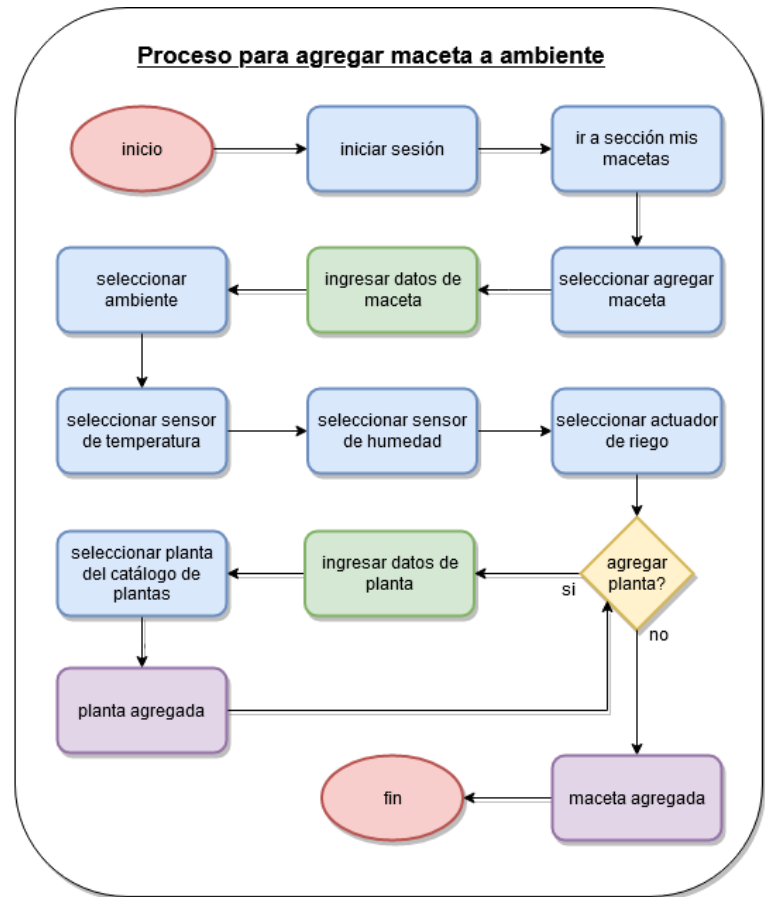


Figura 3.2: Diagrama de flujo para proceso de agregar maceta a ambiente.

Como es de esperar, un ambiente puede contener múltiples macetas. En la Figura 3.2 también se explica el proceso de agregar una maceta, pero siguiendo un flujo diferente.

Finalmente, seguidas o no la opciones antes mostradas, se agrega el ambiente y se termina el proceso. El usuario podrá acceder a la información de sus ambientes de cultivo desde la sección “mis ambientes”.

El proceso de agregar maceta expuesto en la Figura 3.2 es muy similar al mostrado en la Figura 3.1, con la diferencia de que en este último, el proceso se inicia desde la sección “mis macetas”. En esta se selecciona agregar maceta y se ingresan los datos de ésta. Luego se selecciona el ambiente al cual ésta pertenece. Después se sigue un flujo idéntico al mostrado en la Figura 3.1, hasta que la maceta es agregada.

El proceso de “agregar planta” presentado en la Figura 3.3, es también similar al exhibido en la Figura 3.1, con la diferencia de que en éste, el proceso comienza desde la sección “mis plantas”. En esta se selecciona agregar planta, y luego se selecciona el ambiente y maceta para ésta. Continúa con el ingreso de los datos de ésta, para luego seleccionar la planta correspondiente desde el catálogo de plantas. Realizado este el último paso, la planta es agregada.

El proceso de agregar comunidad mostrado en la Figura 3.4 comienza con el inicio de sesión del usuario, luego se dirige a la sección “mis comunidades” y se selecciona “agregar comunidad”.

Finalmente se requiere que se ingresen datos de la comunidad, por ejemplo, su nombre y propósito. Hecho este paso, la comunidad ya puede ser agregada. Con esto, la comunidad ya puede ser vista por otros usuarios desde el catálogo de comunidades, al que se accede también

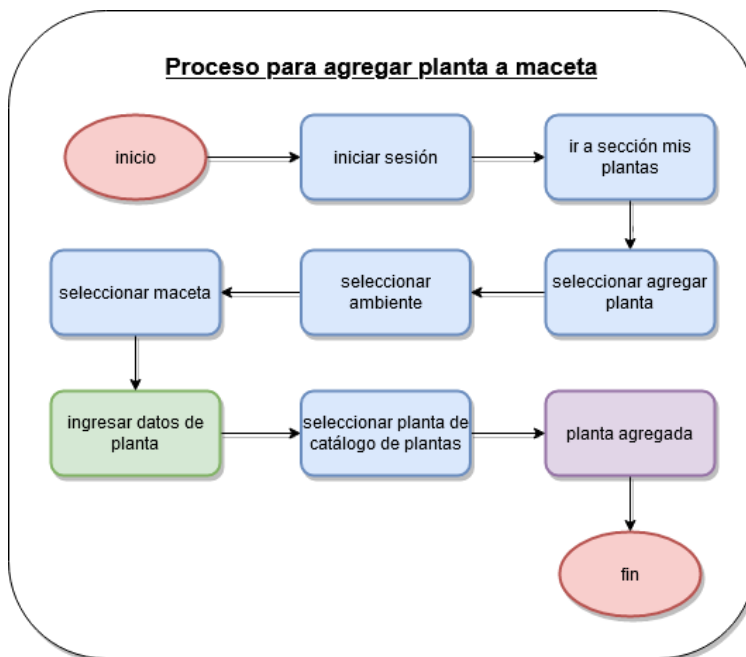


Figura 3.3: Diagrama de flujo para proceso de agregar planta.

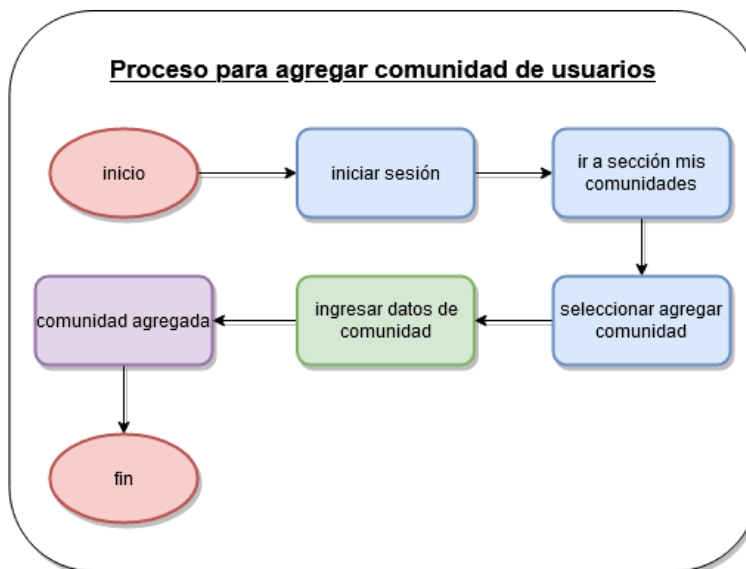


Figura 3.4: Diagrama de flujo para proceso de agregar comunidad.

desde esta sección.

3.4. Arquitectura del Sistema

A continuación se describe la arquitectura de sistema de la Aplicación de Gestión Semiautomática de Cultivos Urbanos, el servicio en la nube, la interacción entre estos y el usuario, y también con el Sistema de Control desarrollado por el memorista Benjamín Zamora (trabajo complementario a esta memoria). La descripción de la arquitectura se hará utilizando la nomenclatura del modelo C4 [31]. Este modelo está compuesto por cuatro niveles de abstracción, comenzando con una mirada general representada con el diagrama de contexto de la aplicación en la Figura 3.5.

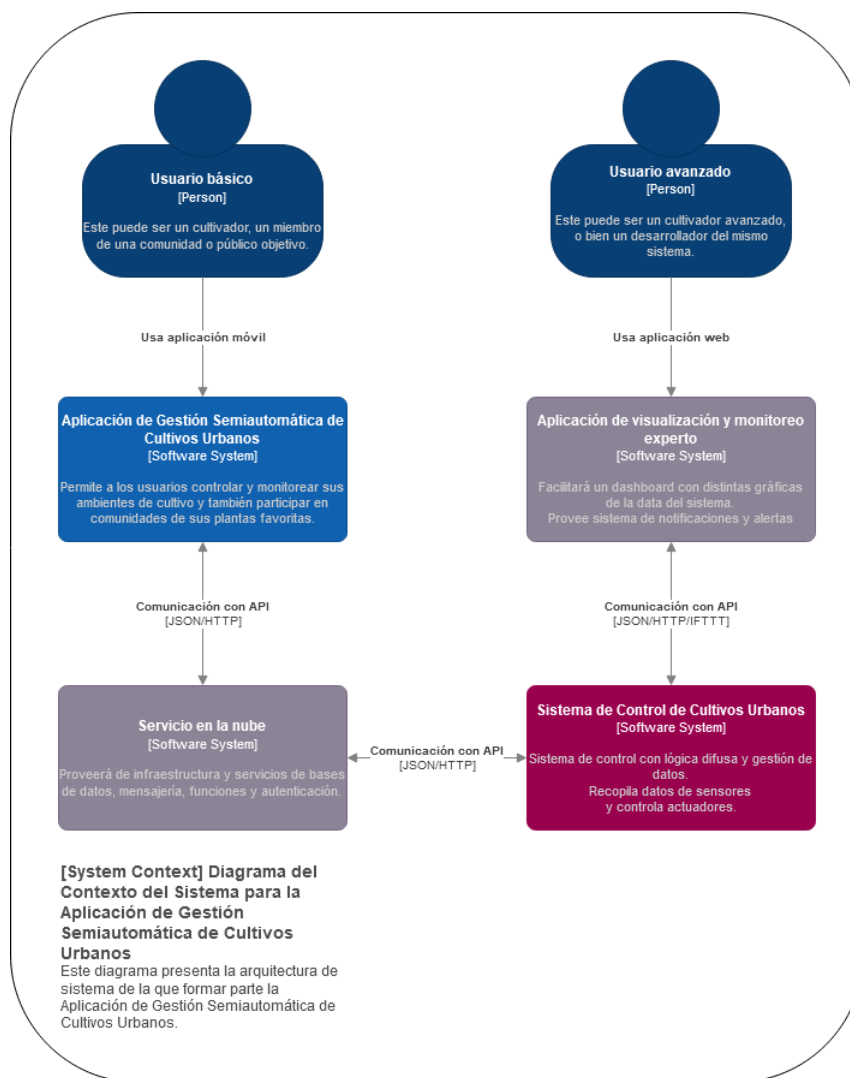


Figura 3.5: Diagrama de contexto para el sistema según el modelo C4.

En la Figura 3.5 se muestra el nivel 1 del modelo C4; es decir, el diagrama de contexto del sistema. Éste muestra cómo interactúa el software que se está construyendo, con el resto de los componentes del ecosistema (ambiente operativo) en el que está inserto. Allí se pueden

observar dos clases de usuarios: básico y avanzado. Los usuarios básicos fueron descritos en la sección 3.1. Los usuarios avanzados son perfiles manejados por el sistema de control desarrollado en la otra memoria aquí referenciada [54]. Estos usuarios avanzados son dos: cultivador avanzado y desarrollador del sistema.

Los usuarios básicos (cultivador, miembro de una comunidad y público objetivo) usan la aplicación móvil (caja azul) para la gestión de su cultivo. La aplicación se comunica con el servicio en la nube (caja gris), el que simultáneamente interactúa con el sistema de control (caja roja).

Los usuarios avanzados (cultivador avanzado y desarrollador del sistema) usan la aplicación de visualización y monitoreo nivel experto (caja gris) a través de una interfaz web. Esta aplicación también interactúa con el sistema de control, y provee a los usuarios un amplio espectro de visualizaciones de los datos generados con la herramienta *Grafana* [5]. Esta herramienta permite que los desarrolladores puedan perfeccionar el sistema, y que los cultivadores avanzados tomen decisiones y tenga información extra sobre su cultivo.

La aplicación de visualización y monitoreo nivel experto, así como el sistema de control son parte de la memoria de Benjamín Zamora. Por esa razón, en los siguientes niveles del modelo C4 sólo se detallará el trabajo realizado en esta memoria. En las Figuras 3.6 y 3.7 se presenta el nivel 2 del modelo C4, es decir, los diagramas de contenedores. Este tipo de diagrama amplía los sistemas de software de la Figura 3.5, exhibiendo los contenedores (aplicaciones, almacenamientos de datos, microservicios, etc.) que componen cada sistema de software.

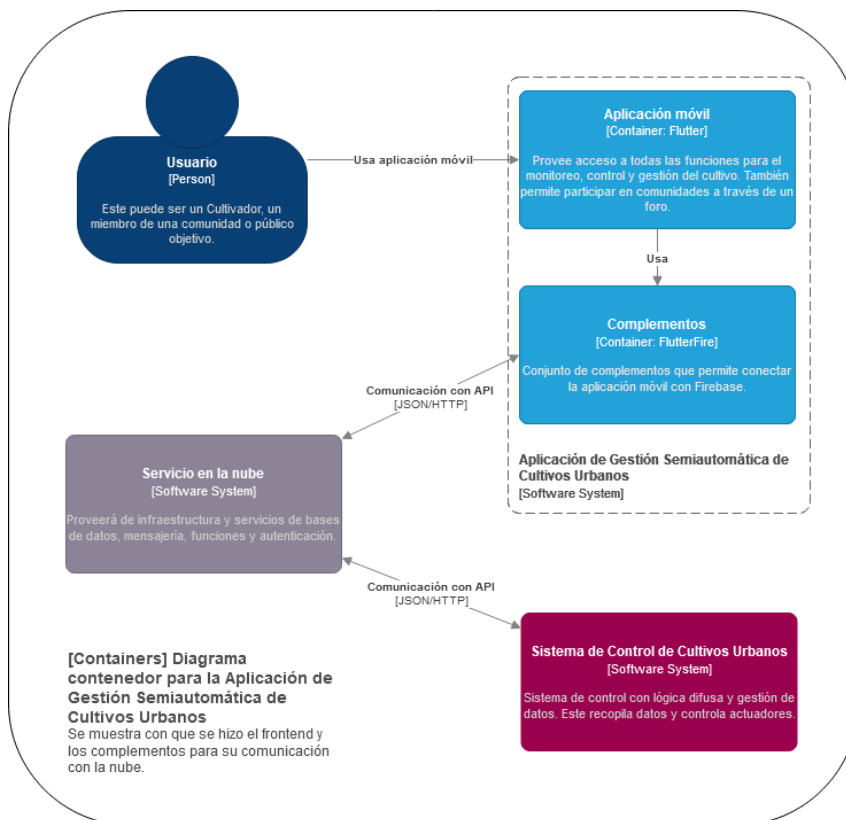


Figura 3.6: Diagrama de contenedor para la aplicación móvil según el modelo C4.

El diagrama contenedor de la Figura 3.6 muestra la aplicación móvil y sus complementos. Esta aplicación entrega acceso a todas las funciones de monitoreo, control y gestión del cultivo. También permite al usuario crear comunidades. El frontend de la aplicación ha sido programado usando el framework Flutter [3], y que a través del uso del complemento FlutterFire, se logra la establecer comunicación con el backend usando una API JSON/HTTPS.

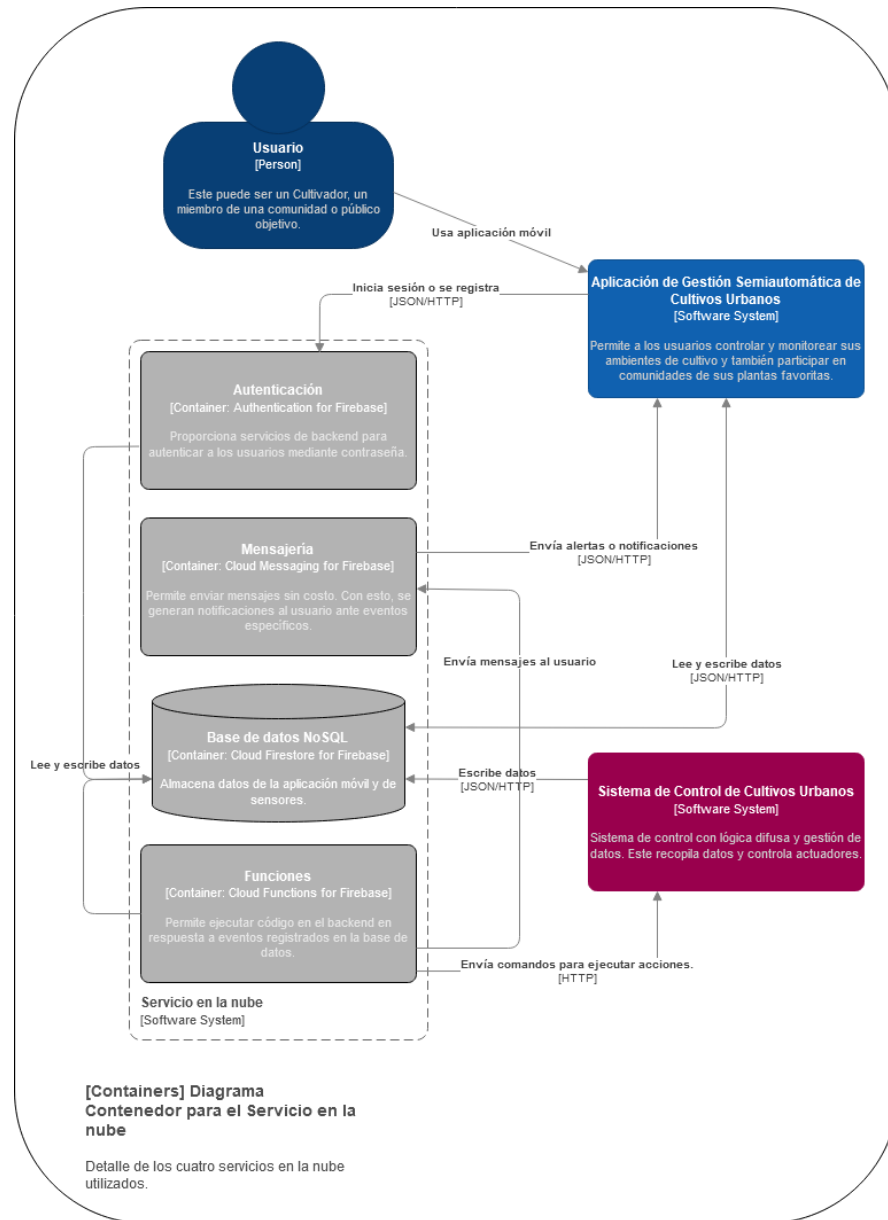


Figura 3.7: Diagrama de contenedor para el Servicio en la nube según el modelo C4.

El diagrama contenedor de la Figura 3.7 expone los microservicios entregados por el servicio en la nube (Firebase), el cual actúa como un BaaS, es decir, backend como servicio. Los microservicios usados son: autenticación para Firebase usando Authentication, mensajería para Firebase usando con Cloud Messaging, una base de datos NoSQL (Cloud Firestore), y diversas funciones disponibles en Cloud Functions para Firebase. A continuación se describen estos microservicios:

- *Authentication*: Proporciona servicios de backend y herramientas para el desarrollador, las cuales son fáciles de usar para autenticar a los usuarios en la aplicación. Admite autenticación mediante contraseñas, números de teléfono y también con proveedores de identidad populares, tales como Google, Facebook y otros. Este servicio es usado para realizar el inicio de sesión y el registro de los usuarios.
- *Cloud Messaging*: Este es un servicio de mensajería multiplataforma, que permite notificar a aplicaciones cliente, indicando que hay datos disponibles para sincronizar. Este servicio es usado en la aplicación para mostrar al cliente visualizaciones de datos en tiempo real, y para mandar alertas.
- *Cloud Firestore*: Esta es una base de datos NoSQL en la nube, flexible y escalable, la cual mantiene los datos sincronizados en las aplicaciones cliente. Cloud Firestore es usada para almacenar toda la información de la aplicación.
- *Cloud Functions*: Este servicio permite ejecutar código automáticamente en el backend, como respuesta a eventos activados por Firebase y solicitudes HTTPS. Este código es almacenado en la nube de Google y se ejecuta en un entorno administrado. Cloud Functions se usa para activar el envío de alertas, avisos y notificaciones utilizando el servicio de mensajería antes mencionado.

En las Figuras 3.8 y 3.9 se presenta el nivel 3 del modelo C4, es decir, los diagramas de componente. Estos diagramas expanden un contenedor individual para mostrar los componentes que contiene. Una componente es una abstracción real del código, por ejemplo, una agrupación de códigos. A continuación se ampliarán los contenedores de la Figura 3.6 y se describirán sus componentes.

El diagrama de componente de la Figura 3.8 muestra los complementos usados para conectar la aplicación móvil con el servicio en la nube. Estos complementos son parte de un conjunto de complementos FlutterFire que, como se mencionó antes, es una extensión diseñada para conectar aplicaciones móviles de Flutter, con el servicio Firebase en la nube. Estos complementos son los antes presentados.

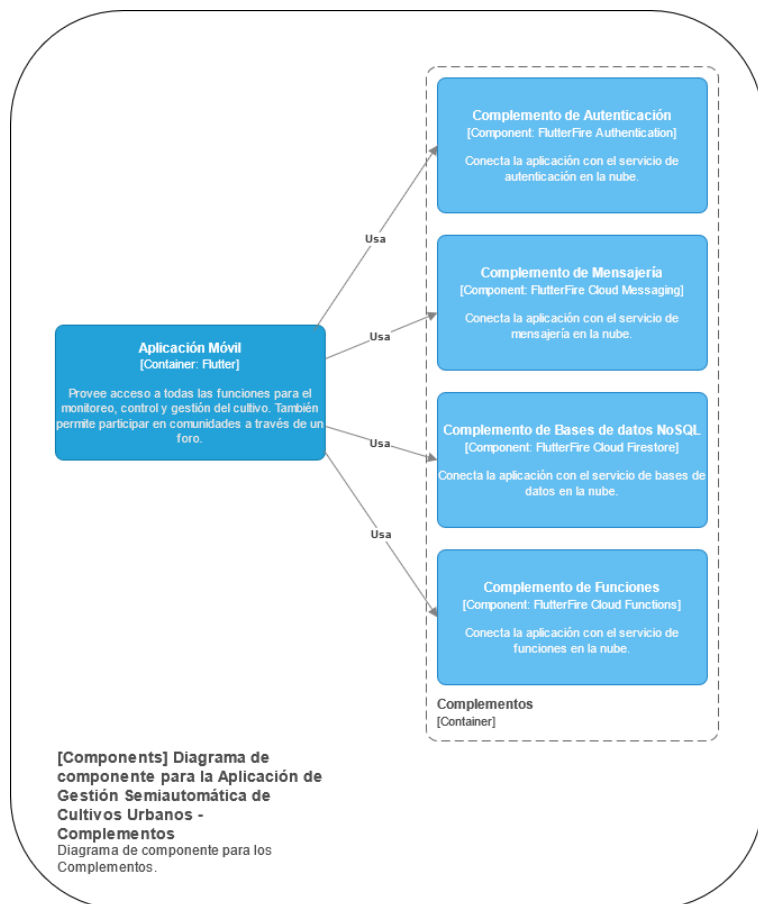


Figura 3.8: Diagrama de componente para Complementos según el modelo C4.

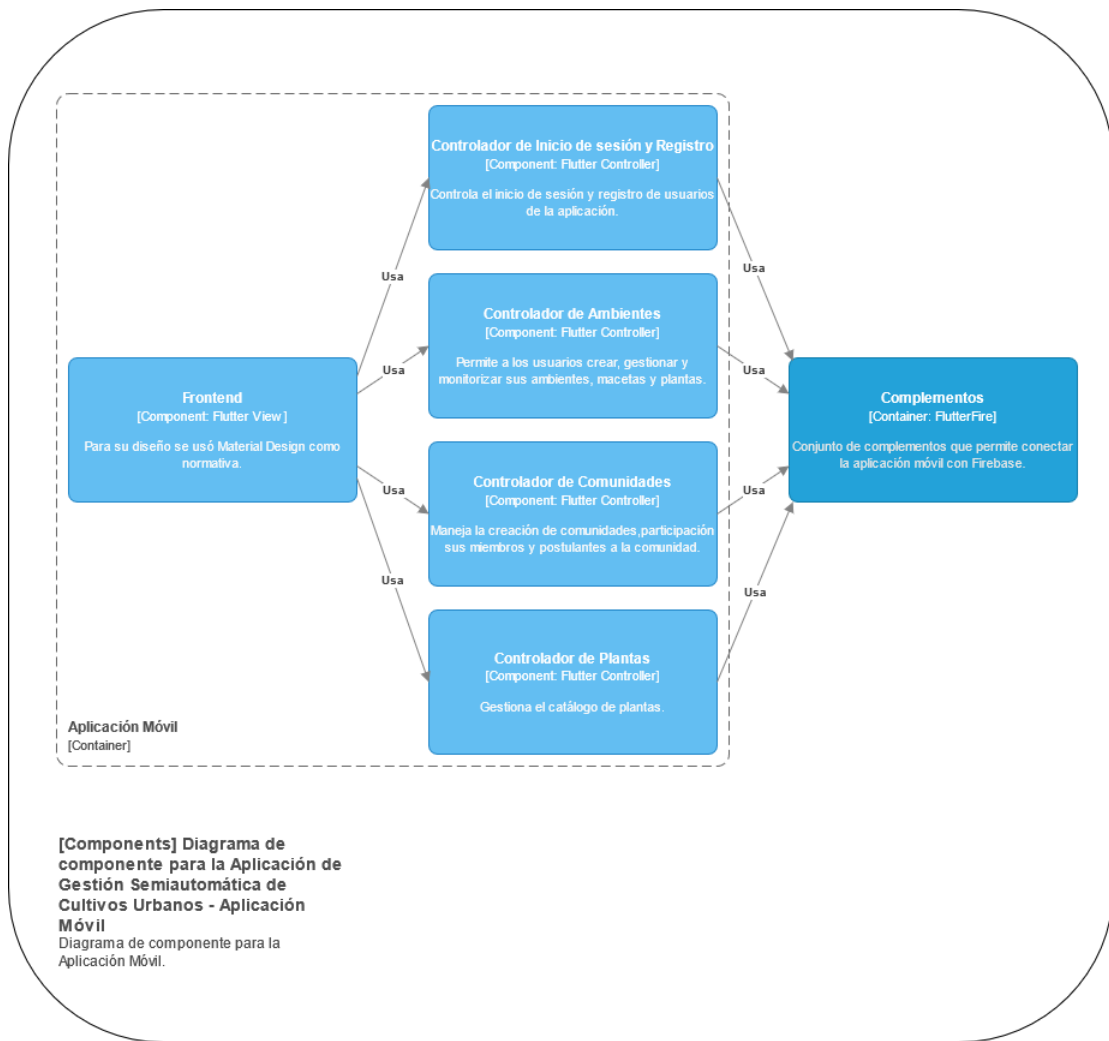


Figura 3.9: Diagrama de componente para la aplicación móvil según el modelo C4.

El diagrama de componente de la Figura 3.9 presenta las principales componentes de la aplicación móvil. Para cada funcionalidad de la aplicación existe una componente *controlador*. La componente *frontend* usa las cuatro componentes de controladores para apoyar el funcionamiento y la comunicación con el backend, a través de los complementos usados por cada controlador. El nivel 4 del modelo C4 no será presentado, ya que no es necesario entrar en el detalle del código de la aplicación.

3.5. Modelo de Datos del Sistema

A continuación se presentan dos modelos de datos. El primero, expuesto en la Figura 3.10, es el modelo de datos relacional que se utilizará con fines explicativos para un mayor entendimiento del lector. El segundo, mostrado en los códigos 8, 9 y 10, corresponde a la versión no relacional del modelo anterior. Este último es el que usa la aplicación mediante el servicio de bases de datos NoSQL en la nube.

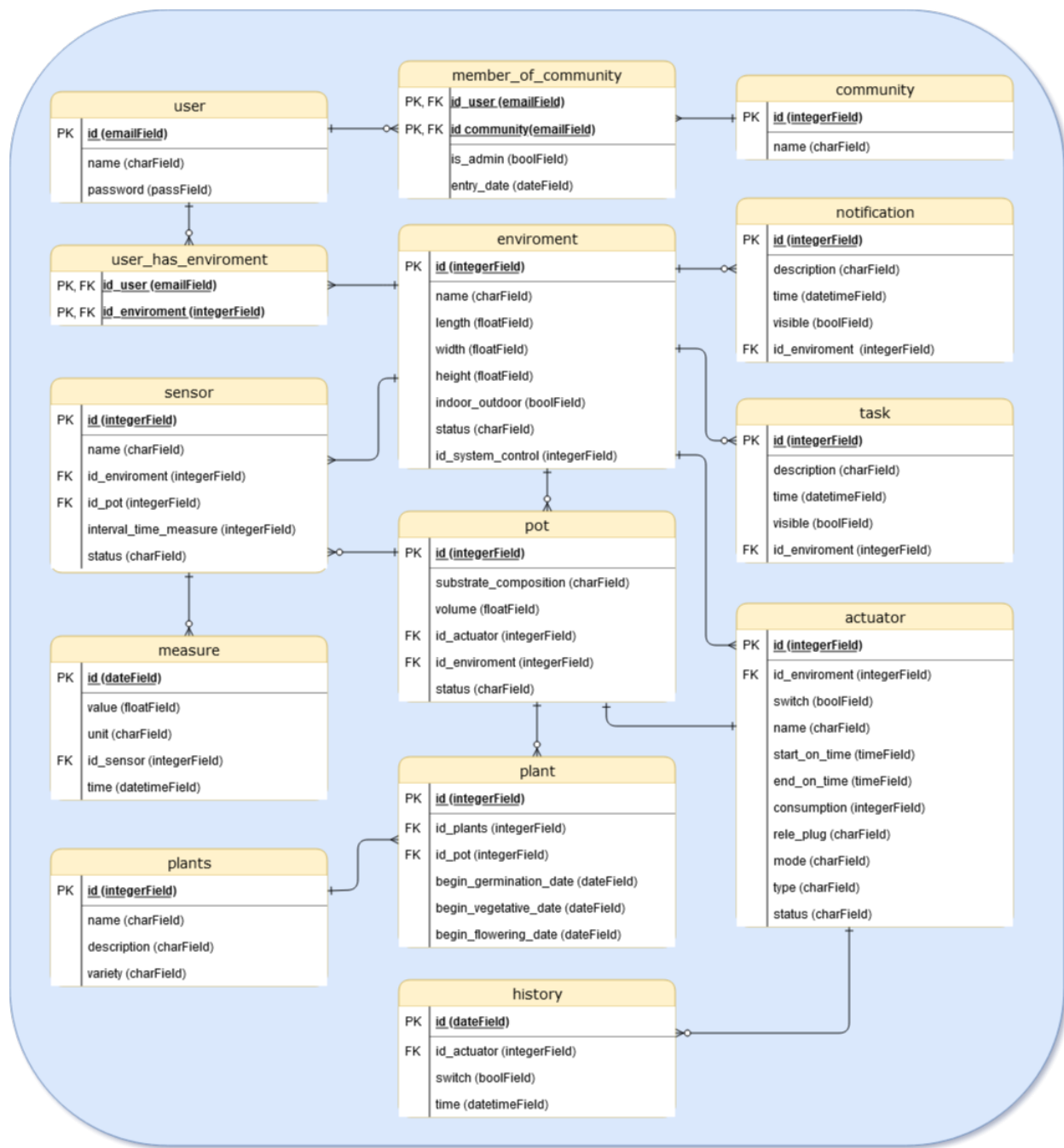


Figura 3.10: Modelo de datos relacional.

El modelo de datos de la Figura 3.10 es relacional, y se compone de las siguientes tablas, junto con su descripción:

- Tabla **user**: Tiene los datos de los usuarios tales como su nombre, email y contraseña. Sus atributos son:
 - *id*: Correo del usuario. Este atributo es llave primaria.
 - *name*: Nombre del usuario.
 - *password*: Contraseña del usuario.

- Tabla **member_of_community**: Contiene a los usuarios que son miembros de una comunidad en particular. Un usuario puede ser miembro de muchas comunidades, y una comunidad puede tener muchos usuarios. Sus atributos son:
 - *id_user*: Llave foránea que identifica un usuario y llave primaria de esta tabla.
 - *id_community*: Llave foránea que identifica una comunidad y llave primaria de esta tabla.
 - *is_admin*: Indica si el usuario es administrador de la comunidad.
 - *entry_date*: Fecha de ingreso del usuario a la comunidad.
- Tabla **community**: Posee los datos de comunidades. Sus atributos son:
 - *id*: Identificador llave primaria.
 - *name*: Nombre de la comunidad.
- Tabla **user_has_enviroment**: Relaciona usuarios con ambientes. Un usuario puede tener muchos ambientes, y un ambiente puede ser accedido por muchos usuarios. Sus atributos son:
 - *id_user*: Llave foránea que identifica un usuario y llave primaria de ésta tabla.
 - *id_enviroment*: Llave foránea que identifica un ambiente y llave primaria de ésta tabla.
- Tabla **enviroment**: Representa un ambiente de cultivo. Sus atributos son:
 - *id*: Identificador llave primaria.
 - *name*: Nombre del ambiente.
 - *length*: Largo del ambiente en metros.
 - *width*: Ancho del ambiente en metros.
 - *height*: Alto del ambiente en metros.
 - *indoor_outdoor*: Es verdadero si el ambiente esta con luz artificial, y falso en caso contrario.
 - *status*: Estado del ambiente.
 - *id_system_control*: Identificador del sistema de control. Este parámetro permite asignar un sistema de control al ambiente agregado por el usuario.
- Tabla **actuator**: Posee los datos un actuador, el cual está asociado al ambiente en el cual funciona. Un ambiente puede tener muchos actuadores. Sus atributos son:
 - *id*: Identificador llave primaria.
 - *id_enviroment*: Llave foránea que identifica un ambiente.
 - *switch*: Indica el estado del switch del actuador. Este puede estar encendido o apagado.
 - *name*: Nombre del actuador.
 - *start_on_time*: Hora de encendido del actuador, cuando éste se encuentra en modalidad de funcionamiento diario(*daily*).
 - *end_on_time*: Hora de apagado del actuador, cuando éste se encuentra en modalidad diaria de funcionamiento.
 - *consumption*: Consumo del actuador en watt.
 - *rele_plug*: Identificador del enchufe donde ha sido conectado el actuador.

- *mode*: Modo de funcionamiento del actuador. Éste puede ser “Manual”, “Automático.” o “Diario”. Las descripción de cada modo es la siguiente:
 - * Modo “Manual”: En este modo el usuario decide cuándo prender o apagar el actuador.
 - * Modo “Automático”: En este modo el sistema de control antes mencionado (descrito en [54], se encarga del funcionamiento del actuador mediante una configuración que utiliza lógica difusa.
 - * Modo “Diario”: En este modo se usa las horas ingresadas en *start_on_time* y *end_on_time* para encender y apagar respectivamente el actuador diariamente.
- *type*: Nombre del tipo de actuador conectado. Los tipo existentes son los siguientes: luz, electroválvula, humidificador, ventilador, y otros. Estos tipos son reconocidos por el sistema de control para su correcto funcionamiento en caso de usar una modalidad automática para el actuador.
- *status*: Estado del actuador.
- Tabla **history**: Contiene el historial de funcionamiento de un actuador en el ambiente, y por ello, un actuador tendrá muchos registros en esta tabla. Los atributos de esta tabla son:
 - *id*: Identificador llave primaria.
 - *id_actuator*: Llave foránea que identifica un actuador.
 - *switch*: Indica el estado del switch del actuador.
 - *time*: Fecha y hora del registro.
- Tabla **sensor**: Posee la información de los sensores usados. Un sensor puede ser usado en un ambiente o en una maceta, pero no en ambos simultáneamente. Sus atributos son:
 - *id*: Identificador llave primaria.
 - *name*: Nombre del sensor.
 - *id_environment*: Llave foránea que identifica el ambiente del sensor.
 - *id_pot*: Llave foránea que identifica la maceta del sensor.
 - *interval_time_measure*: Intervalo de tiempo entre cada medida del sensor.
 - *status*: Estado del sensor.
- Tabla **measure**: Contiene las medidas registradas por los sensores, y por lo tanto, un sensor tendrá muchas medidas. Esta tabla es poblada por el sistema de control, y sus atributos son los siguientes:
 - *id*: Identificador llave primaria.
 - *value*: Valor medido.
 - *unit*: Unidad de medida del valor.
 - *id_sensor*: Llave foránea que identifica un sensor.
 - *time*: Hora y fecha del registro.
- Tabla **pot**: Posee las macetas de un ambiente. Un ambiente puede tener una o más macetas, y sus atributos son:
 - *id*: Identificador llave primaria.

- *substrate_composition*: Detalle de la composición del sustrato usado.
- *volume*: Volumen de la maceta en litros.
- *id_actuator*: Llave foránea que identifica el actuador de riego de la maceta.
- *id_enviroment*: Llave foránea que identifica un ambiente.
- *status*: Estado de la maceta.
- Tabla **plant**: Indica la(s) variedad(es) de planta que contiene una maceta. Cada planta se identifica con una perteneciente al catálogo registrado en la tabla **plants**.
 - *id_plants*: Llave foránea que identifica a la planta, la cual hace match con una del catálogo.
 - *id_pot*: Llave foránea que identifica la maceta que contiene a la planta.
 - *begin_germination_date*: Fecha de inicio de la germinación de la semilla.
 - *begin_vegetative_date*: Fecha de comienzo del ciclo vegetativo de la planta.
 - *begin_flowering_date*: Fecha de inicio del ciclo de floración de la planta.
- Tabla **plants**: Catálogo de plantas existentes, el cual sirve para identificar las plantas que se cultivan en las macetas.
 - *id*: Identificador llave primaria.
 - *name*: Nombre de la planta.
 - *description*: Descripción de la planta.
 - *variety*: Variedad de la planta.
- Tabla **notification**: Posee las notificaciones generadas para un determinado ambiente, maceta o planta.
 - *id*: Identificador llave primaria.
 - *description*: Descripción de la notificación.
 - *visible*: Este es un parámetro booleano; si su valor es verdadero, la notificación es visible para el usuario.
 - *time*: Fecha y hora de recepción de la notificación.
 - *id_enviroment*: Llave foránea que identifica un ambiente del usuario.
- Tabla **task**: Contiene tareas ingresadas por el usuario para ejecutar en una fecha determinada, en un ambiente, maceta o planta.
 - *id*: Identificador llave primaria.
 - *description*: Descripción de la tarea.
 - *time*: Fecha de ejecución de la tarea.
 - *visible*: Este es un parámetro booleano; si su valor es verdadero, la tarea es visible para el usuario.
 - *id_enviroment*: Llave foránea que identifica un ambiente del usuario.

El modelo relacional antes expuesto fue transformado a su versión no relacional para ser implementado en la nube. Este nuevo modelo está compuesto por las siguientes colecciones: **user**, **community**, **plants**, **enviroment**, **pot**, **plant**, **actuator**, **task** y **notification**. Cada colección fue nombrada según la tabla que representan del modelo relacional. A continuación se presentarán ejemplos de documentos que ilustran la estructura usada en cada colección.

```

1  {
2    "id": "r@r.com",
3    "uid_usuario": "228VavF7HYNdh0FsjMhfqzGMfd73",
4    "password": "#####",
5    "enviroments": ["enviroment/avCUmy1EJvPChZuwb9Kb",],
6    "communities": ["community/dDMk0xFJBZzfMwVpixjU",],
7  }

```

Listing 1: Documento de la colección **user**.

```

1  {
2    "id": "avCUmy1EJvPChZuwb9Kb",
3    "name": "Ambiente de Prueba 1",
4    "height": "2",
5    "length": "3",
6    "width": "2",
7    "indoor_outdoor": "true"
8    "id_system_control": "cwekjhf2378fasdbkj",
9    "pots": ["/pots/p85nHo8KzRJjqY750BYs",],
10   "actuators": ["/actuators/ISdVLQZR0wGbMsq5kIOL",],
11   "sensors": ["/sensors/6SvNpLHOCJDyPMpyyJdu",],
12   "tasks": ["/tasks/WBXje9sgNNUYEQjUbdUX",],
13   "notifications": ["/notifications/neCaE3vEeNtuucygfamN",],
14   "status": "good",
15 }

```

Listing 2: Documento de la colección **enviroment**.

```

1  {
2    "id": "p85nHo8KzRJjqY750BYs",
3    "name": "Maceta 1",
4    "volume": "11",
5    "actuators": ["/actuators/234L2KNDSGbMsq5kIOL",],
6    "plants": ["/plant/gnvAuKiyp5tIHEbCQmj0",],
7    "sensors": ["/sensors/neCaE3vEeNtuucygfamN",],
8    "substrate_composition": "Sustrato básico",
9    "status": "good",
10 }

```

Listing 3: Documento de la colección **pot**.

```

1  {
2      "id": "gnvAuKiyp5tIHEbCCmj0",
3      "name": "Tomate",
4      "variety": "Fruta",
5      "description": "una descripcion de ejemplo...",
6      "begin_germination_date": "1 de enero de 2020, 0:00:00 UTC-3",
7      "begin_vegetative_date": "10 de enero de 2020, 0:00:00 UTC-3",
8      "begin_flowering_date": "20 de enero de 2020, 0:00:00 UTC-3",
9
10 }

```

Listing 4: Documento de la colección **plant**.

```

1  {
2      "id": "ISdVLQZROwGbMsq5kIOL",
3      "name": "foco led",
4      "consumption": "10",
5      "mode": "Automático",
6      "type": "Luz",
7      "rele_plug": "Enchufe 1",
8      "switch": false,
9      "type": "Luz",
10     "status": "good",
11     "start_on_time": "",
12     "end_on_time": "",
13     "history": [{"switch": true,
14                 "time": "1 de enero de 2020, 0:00:00 UTC-3"}],
15 }

```

Listing 5: Documento de la colección **actuador**.

```

1  {
2      "id": "6SvNpLHOCJDyPMpyyJdu",
3      "interval_time_measure": "5",
4      "measure": [{"value": "25",
5                  "time": "11 de enero de 2020, 0:00:00 UTC-3",
6                  "unit": "Grados Celsius"},],],
7      "status": "good",
8  }

```

Listing 6: Documento de la colección **sensor**.

```

1  {
2      "id": "neCaE3vEeNtuucygfamN",
3      "description": "Humedad elevada en Maceta de Ambiente de Prueba 1",
4      "name": "Humedad elevada en Maceta",
5      "visible": true,
6  }

```

Listing 7: Documento de la colección **notification**.

```

1  {
2      "id": "WBXje9sgNNUYEQjUbdUX",
3      "description": "Cambio de maceta en Maceta 1, Ambiente de Prueba 1",
4      "name": "Cambiar maceta a una de 20 Litros",
5      "visible": true,
6  }

```

Listing 8: Documento de la colección **task**.

```

1  {
2      "id": "pPaTquucDug0qf6pKfa1",
3      "name": "Comunidad de Frutillas",
4      "members": [
5          {
6              "user": "user/228VavF7HYNdh0FsJMhfqzGMfd73",
7              "is_admin": true,
8              "entry_date" : "08-01-2020"
9          },
10         {
11             "user": "user/cNb700Wxu3yNEUkY6c4X",
12             "is_admin": false,
13             "entry_date" : "07-11-2020"
14         }
15     ]
16 }

```

Listing 9: Documento de la colección **community**.

```
1 {
2   "id": "7CT7J0jXgdCGjDRf9bEq"
3   "name": "Frambuesa",
4   "variety": "Heritage",
5   "description": "This plant grows wild in various regions of Europe,
6   although it is also cultivated, its cultivation being quite
7   important. It is a small, conical or rounded fruit with a velvety
8   red or yellowish skin.",
9 }
```

Listing 10: Documento de la colección **plants**.

Capítulo 4

Implementación del Sistema de Gestión de Información de Cultivos

En este capítulo se presentan los mapas de navegación para las principales funciones del sistema, y se describen en detalle las interfaces más relevantes. En las interfaces se ubicaron marcadores numéricos, para facilitar la referencia a esa área en la explicación de las mismas. Todas las interfaces fueron hechas siguiendo las recomendaciones de diseño de Material Design [12].

En la Figura 4.1 se muestra el mapa de navegación para visitar las secciones del sistema. Los nombres de cada interfaz se ubican junto a cada una de éstas en el mapa. En este mapa el flujo es de izquierda a derecha y descendente. Se comienza por la identificación del usuario. Éste inicia sesión en el sistema, o bien, se registra con su correo y accede. Una vez dentro, el sistema lo conduce a la interfaz de Inicio (ver Figura 4.2). Además, se cuenta con una barra de navegación inferior, que permite al usuario desplazarse entre las secciones del sistema, las cuales son las siguientes: Inicio de Sesión, Inicio, Ambientes, Macetas, Plantas y Comunidades. Estas secciones se explican a continuación.

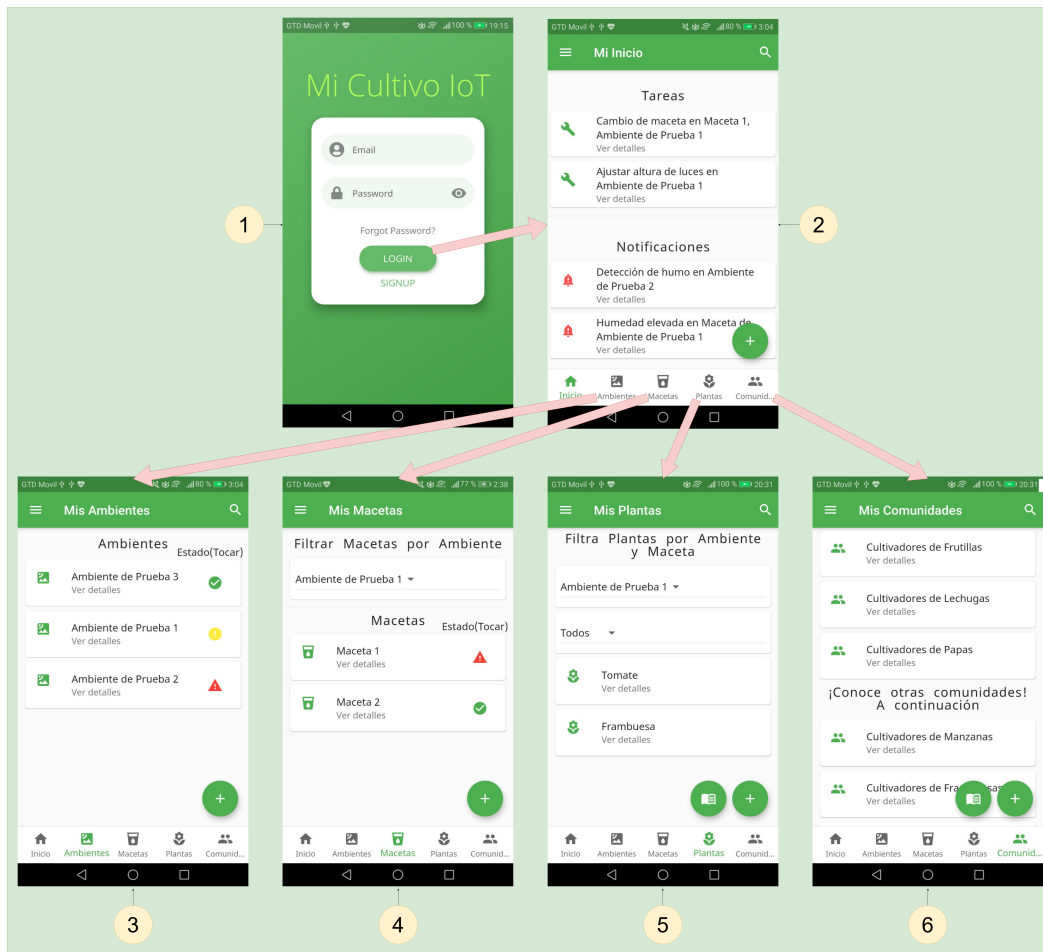


Figura 4.1: Mapa de navegación para visitar las principales secciones del sistema.

4.1. Inicio

En la Figura 4.2 se presenta la interfaz Inicio del sistema, la cual:

1. Despliega un menú con la opción de cerrar sesión.
2. Permite realizar búsquedas sobre las tareas o notificaciones mostradas en esta interfaz.
3. Gestiona tarea ingresada por el usuario. Las tareas serán exhibidas en la interfaz Inicio, desde la fecha para la cual ha sido programada, hasta que el usuario la marque como realizada. Al presionar una tarea, se despliega una ventana de diálogo que informa al usuario sobre más detalles de ésta, y también permite marcarla como realizada.
4. Revisar la notificación del sistema. Las notificaciones serán exhibidas en la interfaz Inicio, hasta que sean marcadas como no visible desde dicha interfaz. Al presionar una notificación, se muestra una ventana de Diálogo que informa al usuario más detalles de ésta, y también permite marcarla como no visible.



Figura 4.2: Interfaz Inicio.

5. Despliega una interfaz con un formulario para agregar una tarea nueva. En éste se ingresa una descripción de la tarea, fecha de ejecución, y el Ambiente, Maceta y/o Planta sobre la cual se realizará dicha tarea.

4.2. Ambientes

En la Figura 4.3 se presenta la interfaz de Ambientes. En ésta se muestra un listado con los ambientes del usuario, donde:

1. Despliega un menú con opciones.
2. Permite realizar una búsqueda de un ambiente por su nombre.
3. Permite ver el detalle de un ambiente. Esta interfaz será expuesta más adelante.
4. Icono indicador del estado del ambiente. Al tocarlo despliega una ventana de diálogo con información. En este caso se indica que el ambiente se encuentra en correctas condiciones.



Figura 4.3: Interfaz Ambientes.

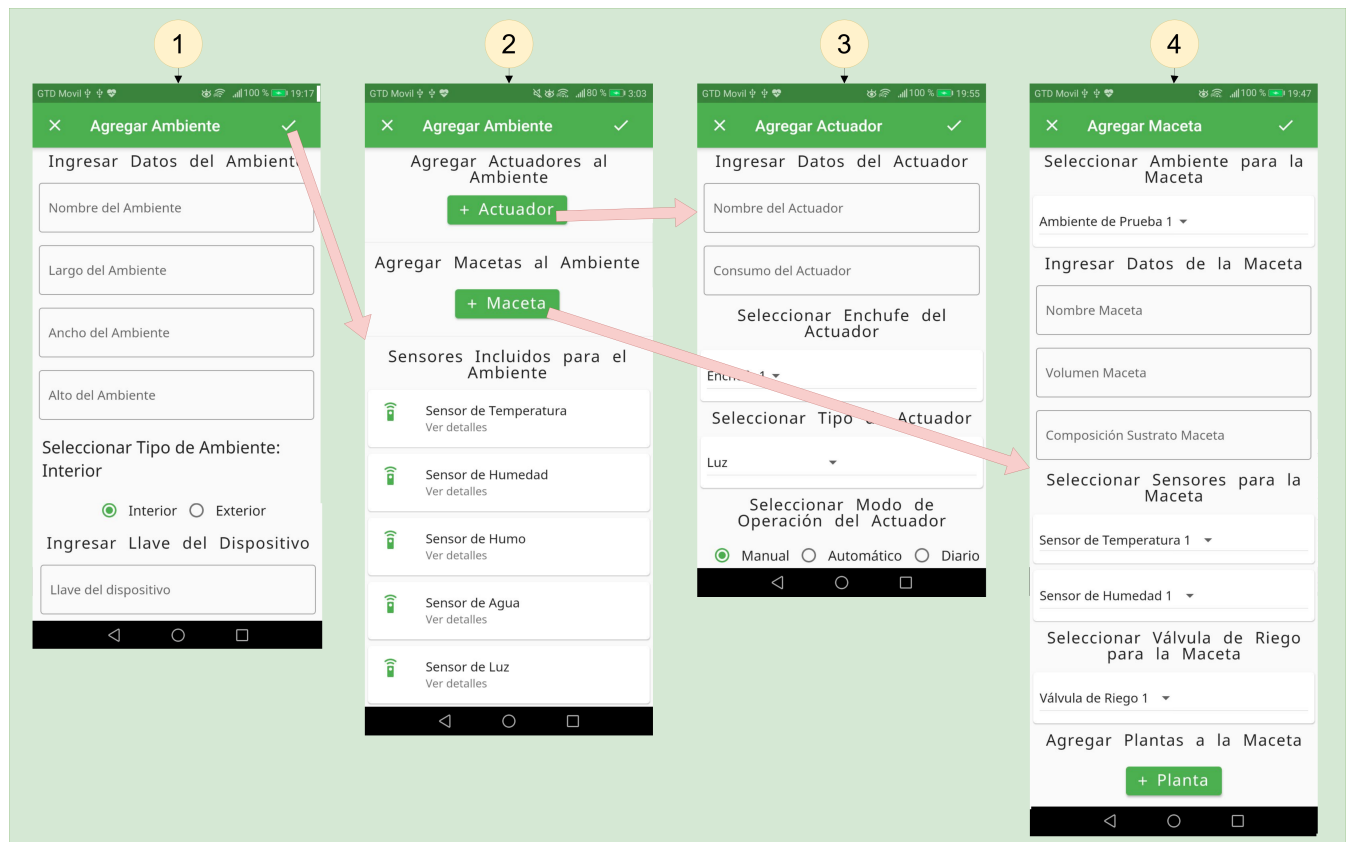


Figura 4.4: Mapa de navegación para agregar un Ambiente.

5. Icono indicador de estado del ambiente. En este caso el color amarillo notifica problemas de carácter semi grave en el ambiente.
6. Icono indicador de estado del ambiente. En este caso el color rojo indica problemas de carácter grave en el ambiente.
7. Despliega una interfaz con un formulario para agregar un ambiente nuevo. Además, en la Figura 4.4 se presenta el mapa de navegación para agregar un ambiente.

Las interfaces del mapa de navegación mostrado en la Figura 4.4, son las siguientes:

1. *Interfaz inicial para agregar un Ambiente.* Aquí se ingresan datos del Ambiente, siendo la más destacable una llave alfanumérica para asociar el Ambiente con el sistema de control instalado en él.



Figura 4.5: Interfaz final agregar Ambiente (continuación de la anterior).

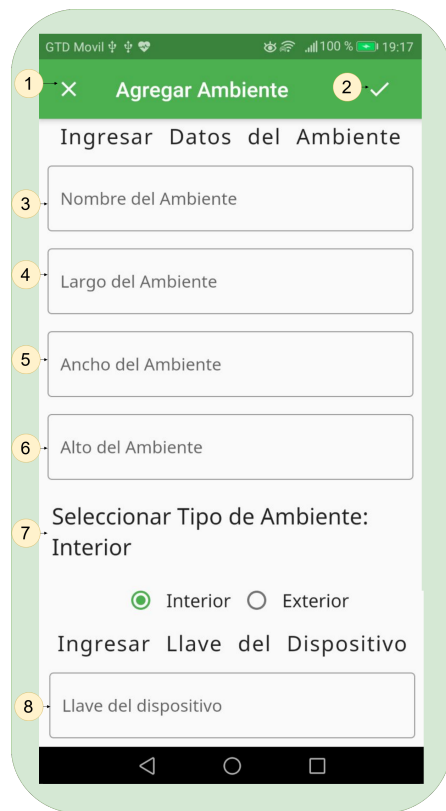


Figura 4.6: Interfaz inicial agregar Ambiente.

2. *Interfaz final para agregar un Ambiente.* Aquí se cargan los sensores incluidos en el sistema de control y es posible agregar macetas y actuadores. Ésta se presentará en detalle en la Figura 4.5.
3. *Interfaz que permite agregar un Actuador al Ambiente.* Ésta se mostrará en detalle en la Figura 4.8.
4. *Interfaz para agregar Maceta a un Ambiente.* Ésta se expondrá en detalle en la Figura 4.7.

En la Figura 4.6 se presenta la interfaz inicial agregar Ambiente, la cual:

1. Permite regresar a la interfaz de Ambientes.
2. Permite continuar el proceso en la interfaz final agregar Ambiente.
3. Permite ingresar un campo de texto para ingresar el nombre del Ambiente.
4. Permite ingresar un campo de texto indicando el largo del Ambiente en metros.

5. Permite ingresar el ancho del Ambiente en metros.
6. Permite ingresar el alto del Ambiente en metros.
7. Aquí se debe seleccionar un tipo de Ambiente, éste puede ser *indoor* si es un Ambiente interior, o *outdoor* si es exterior.
8. Ingresar la llave alfanumérica, para la vinculación con el sistema de control que se usará en este Ambiente.

En la Figura 4.5 se presenta la continuación de la interfaz agrega Ambiente. En ésta se muestra un listado de los Sensores incluidos en el sistema de control que funcionarán en el ambiente. También se tienen dos botones que permiten agregar actuadores y macetas:

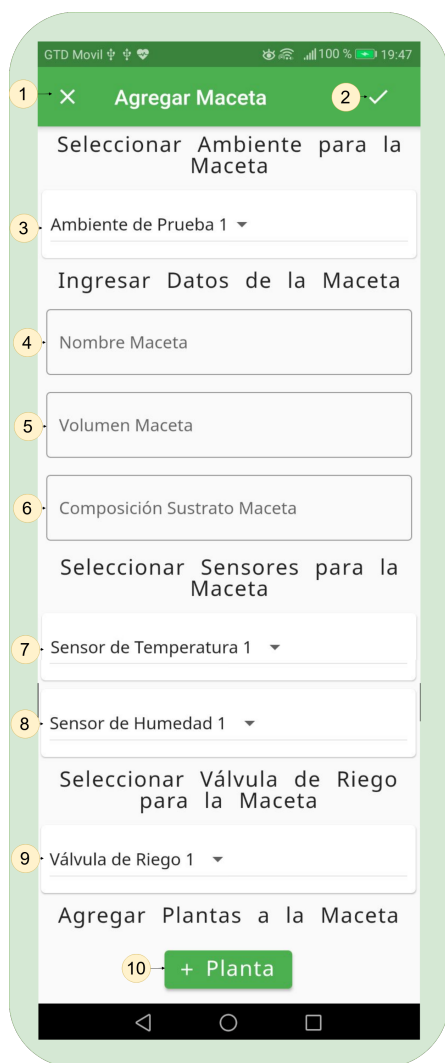


Figura 4.7: Interfaz agrega Maceta a Ambiente.

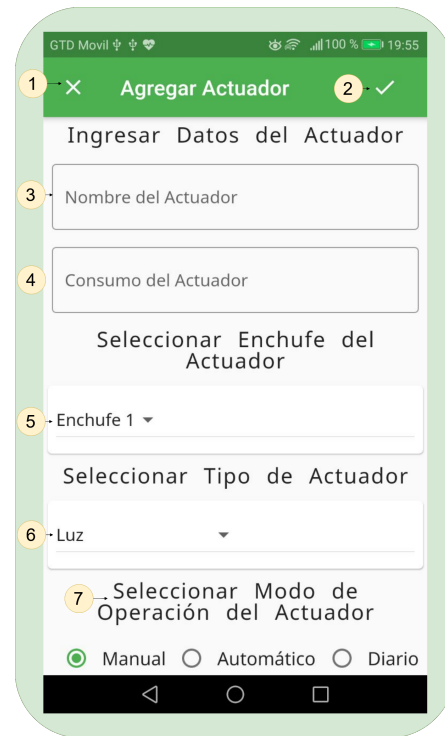


Figura 4.8: Interfaz agrega Actuador a Ambiente.

1. Este icono permite regresar a la interfaz agrega Ambiente.
2. Este icono agrega el nuevo Ambiente a la lista de Ambientes del usuario. Luego, traslada al usuario a la interfaz de Ambientes.
3. Despliega la interfaz para agregar un Actuador al Ambiente. Ésta será detallada más adelante en la Figura 4.8.
4. Despliega la interfaz para agregar una maceta al Ambiente. Esta será detallada más adelante en la Figura 4.7.
5. Título indicador para los sensores incluidos en el sistema de control para el Ambiente.
6. Cuando esta tarjeta es presionada muestra una ventana de diálogo con información sobre dicho Sensor.

A continuación, en la Figura 4.8 se presenta la interfaz para agregar un Actuador al Ambiente:

1. Esta opción permite regresar a la interfaz agrega Ambiente.
2. Agrega el Actuador al Ambiente.
3. Este es un campo de texto para ingresar el nombre del Actuador.
4. Campo de texto para ingresar el consumo en watt del Actuador.
5. Este es un menú desplegable que muestra como opciones los enchufes disponibles. Se debe seleccionar el escogido para este Actuador.
6. Esta también es un menú desplegable. Sus opciones son las categorías de Actuadores existentes en el sistema de control; por ejemplo, si se conecta un extractor de aire, su tipo sería ventilación. El sistema utiliza este parámetro para establecer el funcionamiento adecuado para cada tipo de Actuador.
7. Permite indicar el modo de operación. Se tienen tres opciones para escoger:
 - Manual: En este modo el usuario manualmente enciende y apaga el Actuador.
 - Automático: En este modo el Actuador es manejado por el sistema de control para determinar cuándo debe encenderse y apagarse.
 - Diario: En este modo el usuario ingresa la hora de encendido y apagado del Actuador. Este funcionamiento es diario.

En la Figura 4.7 se presenta la interfaz para agregar una maceta al Ambiente:

1. Permite regresar a la interfaz agrega Ambiente.
2. Agrega la Maceta al ambiente.
3. Menú desplegable para seleccionar el Ambiente de la Maceta.
4. Campo de texto para ingresar el nombre de la Maceta.
5. Campo de texto para ingresar el volumen de la Maceta en litros.
6. Campo de texto para ingresar la composición del sustrato.
7. Menú desplegable para seleccionar un Sensor de temperatura para la Maceta.
8. Menú desplegable para seleccionar un Sensor de humedad para la Maceta.
9. Menú desplegable para seleccionar un Actuador tipo válvula de riego para la Maceta.
10. Despliega la interfaz para agregar una planta a la Maceta. Esta interfaz se mostrará en detalle en la Figura 4.16. En el mapa de navegación para agregar una Maceta de la Figura 4.20 también se explicará el proceso completo a seguir para agregar una planta.

En la Figura 4.9 se muestra el mapa de navegación para ver el detalle de un ambiente. Las interfaces de este mapa de navegación son las siguientes:

1. Interfaz que muestra el detalle de un Sensor. Ésta será detallada en la Figura 4.10.
2. Interfaz que muestra el detalle de un ambiente. Se puede acceder a sus Sensores, Actuadores, Macetas, Tareas y Notificaciones. Esta será explicada con la Figura 4.11.
3. Interfaz que exhibe el detalle de un Actuador. Ésta será explicada en la Figura 4.13.
4. Interfaz que expone el detalle de una Maceta. Ésta será detallada en la Figura 4.12.

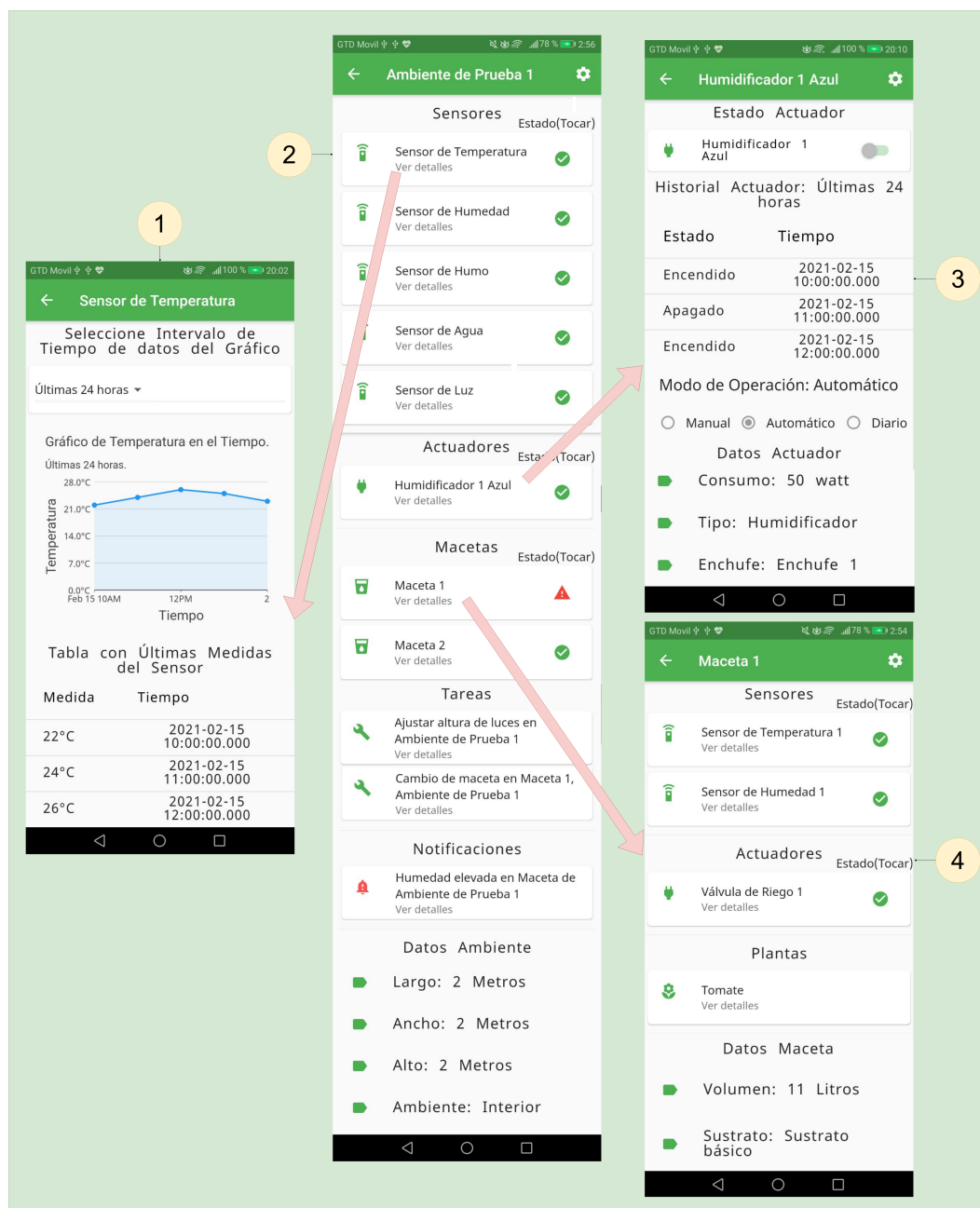


Figura 4.9: Mapa de navegación de un Ambiente.

A continuación, en la Figura 4.11 se presenta en detalle la interfaz de Ambiente, la cual:

1. Permite regresar a la interfaz Ambientes.
2. Despliega un menú con opciones. Las opciones disponibles son: editar Ambiente, agregar Actuador, agregar Maceta, agregar Tarea y eliminar Ambiente.
3. Sensor del Ambiente. Al hacer click sobre éste se accede a una interfaz con visualizaciones de sus datos recopilados. Esta interfaz se muestra en la Figura 4.10.
4. Icono indicador del estado del Sensor. El color verde indica que funciona correctamente, al tocarlo se muestra una ventana de diálogo con información del estado del Sensor.

5. Actuador del Ambiente. Al hacer click sobre éste, se accede a una interfaz con el detalle del Actuador, como se muestra en la Figura 4.13.
6. Icono indicador del estado del Actuador. El color verde indica que funciona correctamente, al tocarlo se muestra una ventana de diálogo con información del estado del Actuador.
7. Maceta del Ambiente. Al hacer click sobre ésta se accede a una interfaz con el detalle de la Maceta, éste se muestra en la Figura 4.12.
8. Icono indicador del estado de la Maceta. El color rojo indica que tiene problemas, al tocarlo se muestra una ventana de diálogo con información del estado de la Maceta.

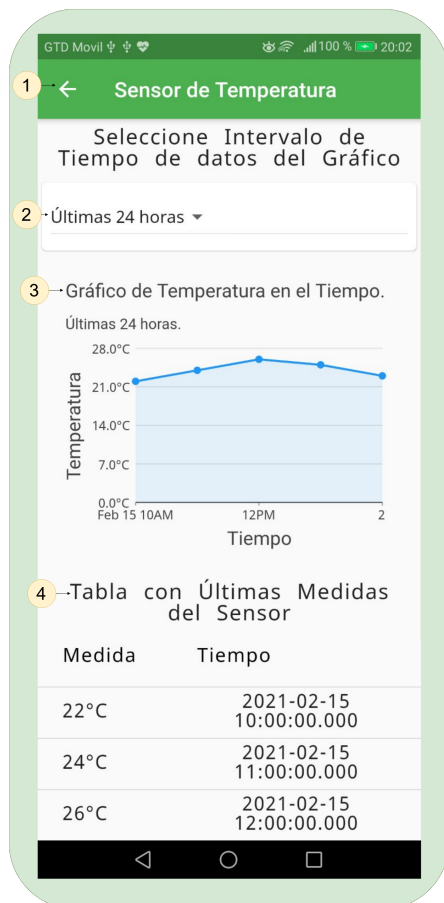


Figura 4.10: Interfaz detalle de Sensor en Ambiente.



Figura 4.11: Interfaz detalle de Ambiente.

9. Tarea del Ambiente. Al hacer click sobre éste se abre una ventana de diálogo con la descripción de la tarea.
10. Notificación del Ambiente. Al hacer click sobre ésta, se abre una ventana de diálogo con la descripción de la notificación.
11. Largo del Ambiente en metros.
12. Ancho del Ambiente en metros.
13. Alto del Ambiente en metros.
14. Tipo de Ambiente, es de interior en este caso.

En la Figura 4.10 se presenta la interfaz que detalla un Sensor de un Ambiente:

1. Permite regresar a la interfaz detalle de Ambiente.
2. Menú desplegable que permite seleccionar el intervalo de tiempo de los datos mostrados en el gráfico. Sus opciones son: últimas 24 horas, última semana y último mes.



Figura 4.12: Interfaz detalle de Maceta en Ambiente.

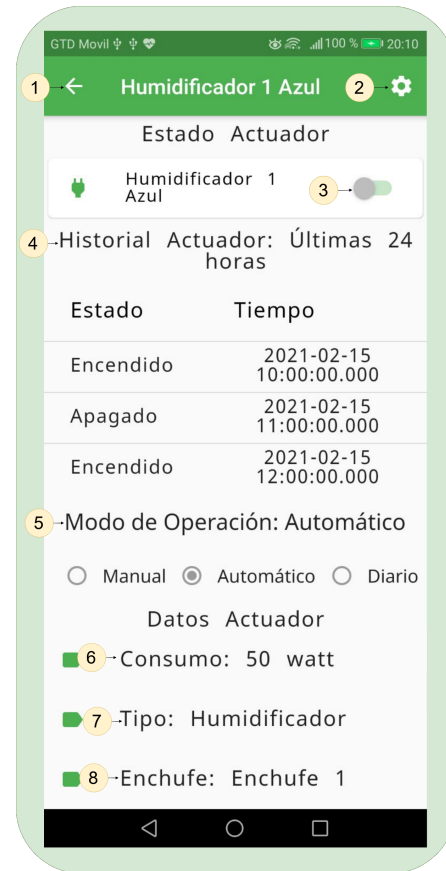


Figura 4.13: Interfaz detalle de Actuador en Ambiente.

3. Gráfico que muestra la temperatura en el tiempo del intervalo seleccionado en el menú antes descrito.
4. Tabla con últimas cinco medidas registradas por el Sensor.

En la Figura 4.13 a continuación, se presenta la interfaz que detalla un Actuador en un Ambiente:

1. Permite regresar a la interfaz detalle de Ambiente.
2. Despliega un menú con las siguientes opciones: editar Actuador, borrar Actuador, historial del último día, última semana, última mes, o seleccionar fechas.
3. Muestra el estado del Actuador, indicando si se encuentra encendido o apagado.

4. Tabla con el historial de funcionamiento del Actuador para el último día. La Columna Estado muestra el historial de funcionamiento del Actuador, y sus posibles valores son Encendido o Apagado. La Columna Tiempo también muestra el historial de funcionamiento del Actuador, y los valores que presenta son la hora y fecha en la cual se produce el cambio de estado.
5. Indica el modo de operación del Actuador. En este caso se encuentra en modo manual. Para cambiar el modo, el usuario debe seleccionar editar Actuador en el menú antes mencionado.
6. Muestra el consumo en watt declarado por el usuario para el Actuador.
7. Consumo del actuador en watt.
8. Tipo de actuador.
9. Enchufe al cual ha sido conectado el Actuador.

En la Figura 4.12 se presenta la interfaz que detalla una Maceta en un Ambiente:

1. Permite regresar a la interfaz detalle de Ambiente.
2. Despliega un menú con las siguientes opciones: editar Maceta, agregar Planta y borrar Maceta.
3. Sensor de la Maceta. Al hacer click sobre éste, se accede a una interfaz con visualizaciones de sus datos recopilados. Esta interfaz se muestra en la Figura 4.10.
4. Icono indicador del estado del Sensor. Si es verde significa que el Sensor está operativo, al tocarlo se muestra una ventada de tipo diálogo con más información.
5. Actuador tipo válvula de riego de la Maceta. Al hacer click sobre éste, se accede a una interfaz con sus datos e historial de funcionamiento. Esta interfaz se muestra en la Figura 4.13.
6. Icono indicador del estado del Actuador. Si es verde significa que el Actuador está operativo, al tocarlo se muestra una ventada de tipo diálogo con más información.
7. Planta de la Maceta. Al hacer click sobre ésta se accede a una interfaz con el detalle de la planta. Ésta se presenta en la Figura 4.18.
8. Volumen de la Maceta en litros.
9. Descripción de la composición del sustrato usado en esta Maceta.

4.3. Macetas

En la Figura 4.14 se presenta la interfaz de Macetas del usuario. En ella se muestra un listado de todas las Macetas del usuario. Cabe destacar que una Maceta puede ser creada en conjunto con un Ambiente, o bien, desde la sección Macetas, donde puede ser asignada a un Ambiente:

1. Despliega un menú con opciones.
2. Menú desplegable. Permite filtrar las Macetas mostradas según el Ambiente al que pertenecen.
3. Al hacer click aquí, se accede a una interfaz con el detalle de la Maceta, el cual fue presentado en la Figura 4.12.

4. Icono indicador del estado de la Maceta. El color rojo indica que tiene problemas, al tocarlo se muestra una ventana de diálogo con información del estado de la Maceta.
5. Icono indicador del estado de la Maceta. El color verde indica que está es condiciones normales, al tocarlo se muestra una ventana de diálogo con información del estado de la Maceta.
6. Despliega la interfaz para agregar una Maceta al Ambiente. Ésta se mostró en la figura 4.7.

En la Figura 4.15 se muestra el mapa de navegación para agregar una Maceta a un Ambiente; particularmente se indican:

1. Interfaz para agregar una Maceta. Ésta fue presentada en la Figura 4.7.
2. Interfaz para agregar Planta a la Maceta. Ésta será mostrada en detalle en la Figura 4.16.



Figura 4.14: Interfaz Macetas.



Figura 4.15: Mapa de navegación para agregar una Maceta.

En la Figura 4.16 se presenta la interfaz para agregar una Planta a una Maceta:

1. Permite regresar a la interfaz agrega Maceta.
2. Agrega la variedad de Planta a la Maceta.
3. Menú desplegable que permite seleccionar el Ambiente para la Planta.
4. Menú desplegable que permite seleccionar la Maceta para la Planta.
5. Botón despliega la interfaz de catálogo de plantas. En ella el usuario debe seleccionar la planta que corresponda con la que será plantada. Este flujo será expuesto en detalle en el mapa de navegación de la Figura 4.20.
6. Botón que abre una ventana, en la cual el usuario selecciona una fecha de un calendario. Esta fecha será considerada como el inicio de la fase germinación de la planta.
7. Ídem al caso anterior. La fecha seleccionada será considerada como el inicio de la fase vegetación de la planta.
8. Ídem al anterior, pero en este caso la fecha indica el inicio de la fase floración de la planta.

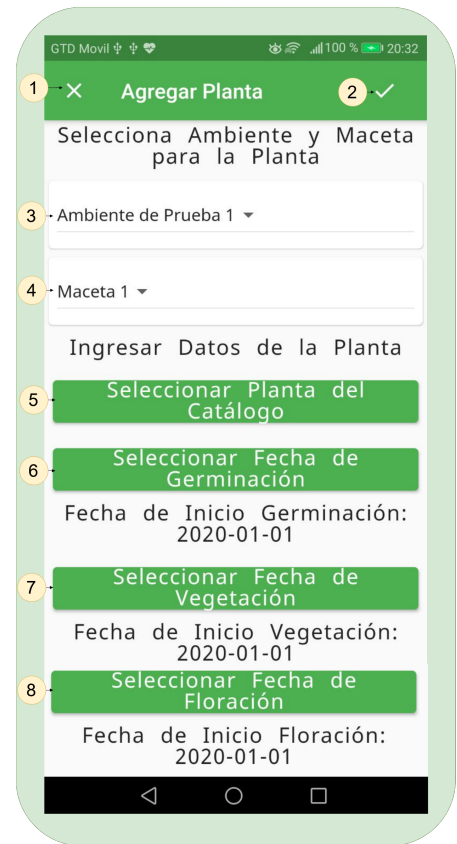


Figura 4.16: Interfaz agrega Planta a Maceta.

En la Figura 4.17 se muestra el mapa de navegación, donde se puede ver el detalle de una Maceta. Las interfaces de este mapa de navegación son las siguientes:

1. Interfaz del detalle de un Sensor. Ésta fue descrita en la Figura 4.10.
2. Interfaz del detalle de una Maceta. Ésta fue descrita en la Figura 4.12.
3. Interfaz del detalle del Actuador tipo válvula de riego. Ésta fue descrita en la Figura 4.13.
4. Interfaz del detalle de una Planta. Ésta será descrita en la Figura 4.18.

En la Figura 4.18 se presenta la interfaz del detalle de una Planta:

1. Permite regresar a la interfaz detalle de la Maceta que la contiene.
2. Despliega un menú con las siguientes opciones: editar Planta y borrar Planta.
3. Despliega una ventana de diálogo con la descripción de la Planta.
4. Fecha de inicio de la fase de germinación de la Planta.
5. Fecha de inicio de la fase de vegetación de la Planta.
6. Fecha de inicio de la fase de floración de la Planta.

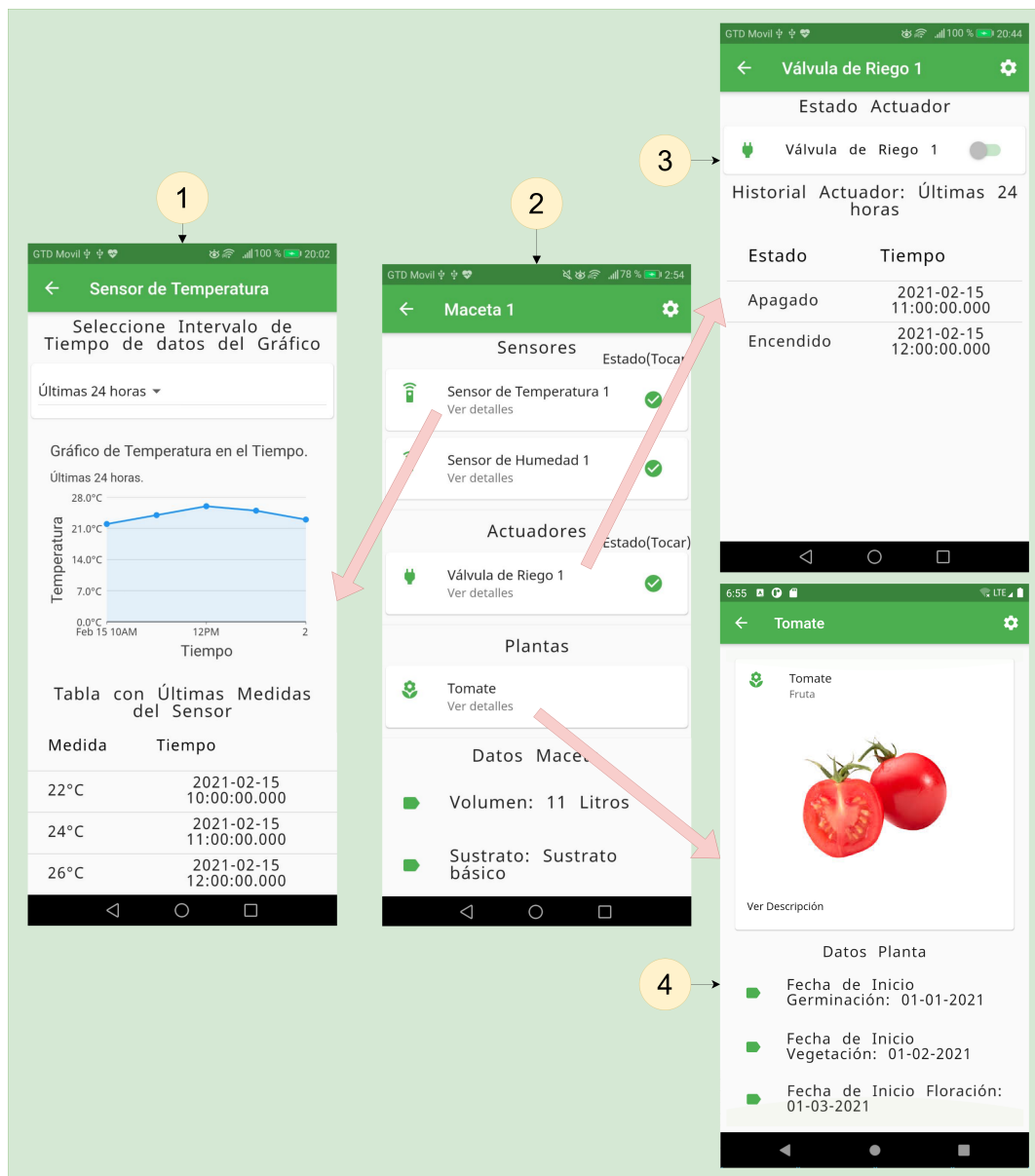


Figura 4.17: Mapa de navegación de una Maceta.

4.4. Plantas

En la Figura 4.19 se presenta la interfaz de la sección Plantas para un usuario, donde se muestra un listado de todas sus Plantas. La descripción de esta interfaz es la siguiente:

1. Despliega un menú con opciones.
2. Permite realizar una búsqueda de plantas por su nombre.
3. Menú desplegable que permite filtrar las plantas mostradas según el Ambiente al que pertenecen.
4. Menú desplegable que permite filtrar las plantas mostradas según la Maceta que las contiene.

- Al hacer click aquí, se accede a una interfaz con el detalle de la Planta, el cual fue presentado en la Figura 4.18.
- Despliega la interfaz del catálogo de plantas. Ésta se expone más en detalle en la Figura 4.21.
- Despliega la interfaz para agregar una Planta. Ésta fue explicada en la figura 4.16. El flujo completo para agregar una Planta se describirá con el mapa de navegación de la Figura 4.20.

En la Figura 4.20 se presenta el mapa de navegación para agregar una Planta a una Maceta, las interfaces que lo componen son las siguientes:

- Interfaz para agregar Planta, fue mostrada en la Figura 4.16.
- Interfaz del catálogo de Plantas. Ésta se describe en la Figura 4.21.
- Interfaz del detalle de la Planta. Ésta fue descrita en la Figura 4.18.

En la Figura 4.21 se presenta la interfaz del catálogo de Plantas. En ella se muestra el listado de todas las Plantas existentes en la base de datos del sistema. A través de este catálogo se puede:

- Regresar a la interfaz desde la cual se accedió.
- Realizar una búsqueda de una planta por su nombre en el catálogo.
- Al hacer click aquí, se accede a una interfaz con el detalle de la Planta, que muestra una imagen y una descripción de ésta.

4.5. Comunidades

En la Figura 4.22 se presenta la interfaz de comunidades del usuario; en ésta se muestra un listado de todas las comunidades de las cuales es miembro (en la figura se observan tres). Además, ahí mismo se le sugieren otras al usuario. A continuación se presenta una descripción de dicha interfaz:

- Despliega un menú con opciones.
- Permite realizar una búsqueda de comunidades por su nombre.
- Al hacer click aquí se accede a una interfaz con el detalle de la Comunidad.



Figura 4.18: Interfaz detalle de Planta en Maceta.

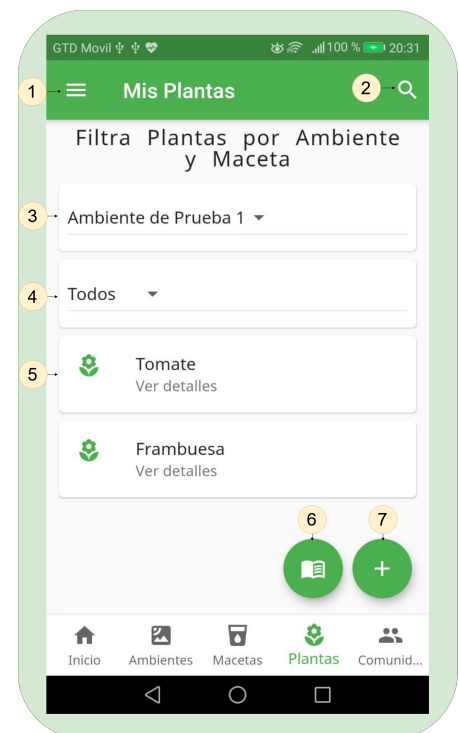


Figura 4.19: Interfaz Plantas.

4. Se invita al usuario a conocer y a postular a nuevas comunidades.
5. Accede a una interfaz con el detalle de la Comunidad.
6. Accede al catálogo de comunidades.
7. Despliega una interfaz con un formulario para crear una Comunidad.

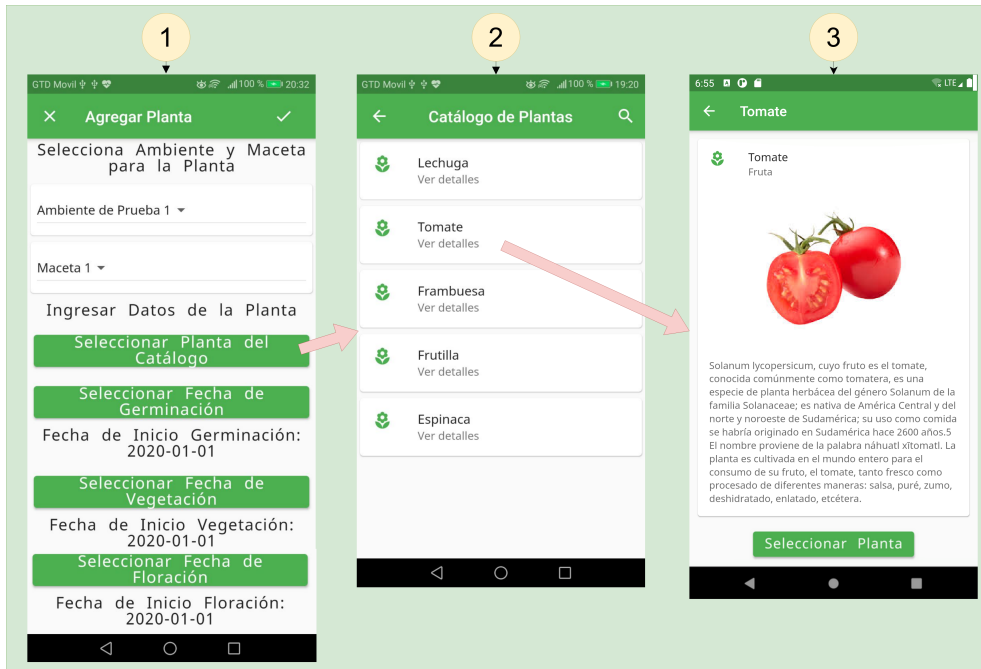


Figura 4.20: Mapa de navegación para agregar Planta a Maceta.

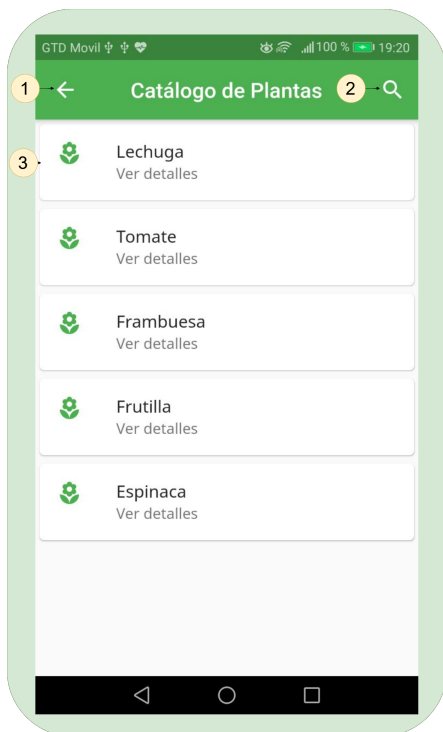


Figura 4.21: Interfaz del Catálogo de Plantas.



Figura 4.22: Interfaz de Comunidades.

Capítulo 5

Evaluación de la Solución

Para determinar la usabilidad y utilidad percibida de la solución, por parte de los potenciales usuarios de la aplicación, se realizó una prueba en base a un sketch que contenía las operaciones típicas para crear, modificar y monitorear un cultivo. A continuación se describe a los participantes del proceso de evaluación, la dinámica de evaluación realizada, el instrumento utilizado y los resultados obtenidos en términos de la usabilidad y utilidad percibida por los usuarios.

5.1. Participantes

En la evaluación participaron 8 personas con una edad promedio de 29 años, de los cuales 5 tenían experiencia en cultivo urbano, y los 3 restantes planeaban tener un cultivo pero sin experiencia previa. La experiencia promedio de los 5 participantes experimentados fue de 4,4 años.

5.2. Descripción del Proceso de Evaluación Realizado

El proceso inicia con una introducción sobre la aplicación. En ella se les indica a los usuarios que ésta permite controlar y monitorear un ambiente de cultivo, el cual debe estar instrumentado; es decir, con sensores y actuadores instalados en él.

Como primera acción se les pidió crear un nuevo ambiente de cultivo (ficticio), el cual debía incluir los siguientes componentes:

- Los siguientes cinco sensores: de temperatura, de humedad relativa, de luz, de agua y de humo.
- Cinco enchufes para conectar actuadores para al ambiente de cultivo. Los actuadores pueden ser: luces, ventiladores, extractores, calefactores, sistemas de aire acondicionado y humidificadores.
- Diez sensores de temperatura del sustrato, a ser instalado en las diez macetas que forman parte del ambiente de cultivo.

- Diez sensores de humedad relativa para el sustrato contenido en las diez macetas.
- Diez actuadores (válvulas de riego); uno para cada una de las diez macetas.

Luego de crear el ambiente de cultivo, los evaluadores utilizaron un usuario de prueba con datos cargados previamente, y un código alfanumérico que debían ingresar al momento de crear un ambiente, y así poder conectarlo con su respectivo sistema de control.

Una vez hecho eso, se le pidió a los usuarios que siguieran un sketch utilizando la aplicación. A continuación se indica el sketch utilizado.

1. *Agregar dos ambientes de cultivo.* El primero debe incluir un actuador del tipo luz, y el segundo debe incluir un actuador del tipo humidificador.
2. *Agregar dos macetas al primer ambiente creado.* Cada maceta debe incluir sus respectivos sensores (humedad y temperatura) y un actuador para el riego (electroválvula).
3. *Agregar dos variedades de plantas a una de las macetas agregadas en el paso anterior.* En cada una indicar fecha de germinación.
4. *Ver historial de funcionamiento del actuador tipo humidificador del ambiente de prueba incluido.*
5. *Ver gráficos del sensor de temperatura en el ambiente de prueba incluido.*
6. *Ver historial de funcionamiento del actuador tipo válvula de riego de la maceta 1.*
7. *Monitorear y verificar el estado del cultivo del ambiente de prueba.*
8. *Inscribirse en una comunidad para participar en ella.*

Durante la prueba, a cada usuario se le tomó el tiempo que le insumió completar cada operación, y se computó también la correctitud en la realización de las operaciones. Luego, cada usuario tuvo unos minutos para revisar la aplicación, para después completar una encuesta que buscó capturar la usabilidad y utilidad de la herramienta, de acuerdo a la percepción de los usuarios.

5.3. Instrumento de Evaluación de la Usabilidad Percibida

Para la evaluación de la usabilidad de la herramienta se utilizó la System Usability Scale (SUS), o Escala de Usabilidad de Sistemas, en español [29, 30]. Este instrumento, que fue propuesto para medir usabilidad de una aplicación Web, de acuerdo a la percepción de los usuarios. La escala SUS consta de 10 ítems (aseveraciones), donde el usuario debe indicar su nivel de acuerdo o desacuerdo con ellas utilizando una escala Likert de 5 puntos (siendo 1 totalmente en desacuerdo, y 5 totalmente de acuerdo). Una vez que todos los participantes han completado la encuesta, se calcula el promedio de las respuestas, y se obtiene una calificación final que define el nivel de usabilidad de la aplicación Web.

El cuestionario SUS utilizado incluyó los siguientes 10 ítems:

1. Creo que me gustaría usar este sistema frecuentemente.
2. Encuentro al sistema innecesariamente complejo.

3. Creo que el sistema es fácil de usar.
4. Creo que necesitaría ayuda de una persona con conocimientos técnicos para usar este sistema.
5. Las funciones de este sistema están bien integradas.
6. Creo que el sistema es muy inconsistente.
7. Imagino que la mayoría de la gente aprendería a usar este sistema en forma muy rápida.
8. Encuentro que el sistema es muy engorroso de usar.
9. Me siento seguro/a al usar este sistema.
10. Necesité aprender muchas cosas antes de aprender a usar este sistema.

Para el cálculo del nivel de usabilidad se deben sumar los resultados entregados por los usuarios, donde las respuestas se computarán de la siguiente manera:

- Los ítems impares (1, 3, 5, 7 y 9) tomarán el valor de la respuesta y se les restará 1.
- Los ítems pares (2, 4, 6, 8 y 10) tomarán el valor de 5, menos el valor de la respuesta.

Una vez obtenido la suma total, a ese resultado se lo multiplica por 2,5, así la fórmula de cálculo queda como se muestra a continuación:

$$\text{Nivel de usabilidad: } ((r(1) - 1) + (5 - r(2)) + (r(3) - 1) + (5 - r(4)) + (r(5) - 1) + (5 - r(6)) + (r(7) - 1) + (5 - r(8)) + (r(9) - 1) + (5 - r(10))) * 2,5$$

El valor final obtenido respecto a la usabilidad del sistema se evalúa con los siguientes tracks:

- Si no llega a 25 puntos, entonces el escenario es “lo peor imaginable”.
- Si va desde 25 a 38, entonces la usabilidad es considerada “pobre”.
- Si va desde 39 a 52, entonces la usabilidad es considerada “aceptable”.
- Si va desde 53 a 73, entonces la usabilidad es considerada “buena”.
- Si va desde 74 a 85, entonces la usabilidad es considerada “excelente”.
- Si va desde 86 a 100, entonces es “lo mejor posible”, que es la aspiración de toda aplicación móvil.

5.4. Instrumento de Evaluación de la Utilidad Percibida

La escala SUS fue extendida para incluir ítems que evalúan la utilidad del sistema, según la percepción del propio usuario. Esto incluye el nivel de suficiencia de la funcionalidad entregada por la herramienta, y también los servicios que potencialmente podrían faltar en la aplicación. Los ítems agregados fueron los siguientes:

11. Los servicios ofrecidos por la aplicación son suficientes como para crear y gestionar un cultivo.
12. Hay servicios que deben ser agregados a la aplicación para realizar esta labor de manera más apropiada.

13. Hay servicios que, aunque no son mandatorios, podrían ser agregados a la herramienta para hacerla más efectiva.
14. Comentario abierto: Indicar los servicios que a su juicio deben ser agregados a la aplicación, y también aquellos que usted considera como deseable de agregar.

Respecto al cómputo de la utilidad percibida por el usuario (ítems 11 a 13), ésta se computa de la siguiente manera:

$$\text{Utilidad percibida: } 0,5 * (r(11) - 1) + 0,35 * (5 - r(12)) + 0,15 * (5 - (r(13)))$$

La evaluación del ítem 11 pesa un 50 % del total de la utilidad, ya que ahí se determina si la herramienta llega a ser un MVP (Minimum Viable Product). El ítem 12 representa una funcionalidad que es necesaria, pero no fundamental; por esa razón, dicho ítem pesa un 35 %. Finalmente, el ítem 13 que representa a la funcionalidad deseable, pesa un 15 %.

El valor final obtenido respecto a la utilidad del sistema se evalúa con los siguientes tracks:

- Si es 0, entonces el escenario es “lo peor imaginable”.
- Si está entre 0 y 1, entonces la utilidad es considerada “pobre”.
- Si está entre 1 y 2, entonces la utilidad es considerada “aceptable”.
- Si está entre 2 y 3, entonces la utilidad es considerada “buena”.
- Si está entre 3 y 4, entonces la utilidad es considerada “excelente”.
- Si es 4, entonces es “lo mejor posible”.

5.5. Resultados obtenidos

A continuación se presentan los resultados obtenidos respecto a los tiempos de ejecución y la correctitud de las operaciones realizadas. Luego se indican los resultados de la encuesta SUS, y un posterior análisis de estos.

5.5.1. Tiempos y correctitud de las operaciones realizadas

En las tablas 5.1 y 5.2 se indican los tiempos de ejecución y la correctitud de las operaciones realizadas por cada participante. En la tabla 5.2 se indica con la letra L (logrado) si la tarea fue ejecutada con éxito, y NL (no logrado) en caso contrario.

Operación \ Participante	1	2	3	4	5	6	7	8	Total
Participante 1	4:03	3:05	1:12	0:17	0:46	0:25	0:12	0:35	10:35
Participante 2	3:43	2:09	1:48	0:22	0:12	0:15	0:58	0:28	9:55
Participante 3	3:04	1:40	0:40	0:30	0:20	0:30	2:12	0:38	9:34
Participante 4	3:02	2:40	0:55	0:26	0:08	0:11	0:06	0:20	7:48
Participante 5	2:46	1:29	1:08	0:10	0:04	0:10	1:41	0:15	7:43
Participante 6	3:49	3:01	1:37	0:35	0:36	0:08	1:34	0:45	10:25
Participante 7	3:02	1:37	1:00	0:20	0:07	0:14	1:00	0:32	7:52
Participante 8	2:44	2:21	0:50	0:17	0:30	0:17	1:00	0:58	8:57
Promedio	3:16	2:15	1:08	0:22	0:20	0:16	1:05	0:34	9:06

Tabla 5.1: Tiempos de ejecución de las operaciones del sketch.

Operación \ Participante	1	2	3	4	5	6	7	8	Total
Participante 1	L	L	L	L	L	L	L	L	8L
Participante 2	L	L	L	L	L	L	L	L	8L
Participante 3	L	L	L	L	L	L	NL	L	7L
Participante 4	L	L	L	L	L	L	L	L	8L
Participante 5	L	L	L	L	L	L	L	L	8L
Participante 6	L	L	L	L	L	L	L	L	8L
Participante 7	L	L	L	L	L	L	L	L	8L
Participante 8	L	L	L	L	L	L	L	L	8L

Tabla 5.2: Correctitud de los participantes en las operaciones del sketch.

El tiempo promedio de los participantes en la ejecución del sketch fue de 9 minutos con 6 segundos, y la correctitud promedio fue de 98,4% en las operaciones realizadas.

5.5.2. Usabilidad percibida

En la siguiente tabla se muestran los resultados recopilados en la evaluación de la usabilidad, de acuerdo a la percepción de los usuarios.

Item \ Participante	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6	Item 7	Item 8	Item 9	Item 10
Participante 1	5	3	2	2	5	1	3	3	5	4
Participante 2	5	3	3	2	4	1	4	2	4	1
Participante 3	5	1	5	4	5	1	5	1	5	4
Participante 4	4	1	3	2	5	1	4	1	5	1
Participante 5	5	3	4	2	5	4	5	2	5	2
Participante 6	5	2	4	1	4	1	5	1	5	1
Participante 7	4	2	4	2	4	1	4	2	4	1
Participante 8	2	2	2	1	4	2	2	3	2	2

Tabla 5.3: Resultados de la evaluación de usabilidad de la aplicación.

Participantes	Nivel de usabilidad
Participante 1	67,5
Participante 2	77,5
Participante 3	85
Participante 4	87,5
Participante 5	77,5
Participante 6	92,5
Participante 7	80
Participante 8	55

Tabla 5.4: Usabilidad percibida por cada participante.

El valor final promedio obtenido para el nivel de usabilidad es de 77,8 puntos, por lo que la usabilidad de la aplicación se considera entonces como “excelente”.

5.5.3. Utilidad percibida

En la siguiente tabla se muestran los resultados recopilados en la evaluación de la utilidad, de acuerdo a la percepción de los usuarios.

Usuario \ Item	Item 1	Item 2	Item 3
Participante 1	5	2	2
Participante 2	4	3	2
Participante 3	5	1	1
Participante 4	5	1	1
Participante 5	5	4	1
Participante 6	5	4	4
Participante 7	3	3	4
Participante 8	2	5	5

Tabla 5.5: Resultados de la evaluación de utilidad de la aplicación.

El valor final promedio obtenido para el nivel de utilidad es de 2,74 puntos, por lo que la utilidad de la aplicación se considera entonces como “buena”.

5.5.4. Servicios sugeridos

Los servicios deseables, mencionados por los participantes en la sección de comentarios abiertos, fueron los siguientes:

- Permitir la compra y venta de plantas (por ej., de hortalizas) a través de las comunidades.
- Permitir configurar funcionamientos específicos del sistema de control, para cultivar plantas que requieran condiciones distintas al resto.
- Añadir un calendario lunar para saber cuándo ejecutar algunas tareas, como por ejemplo, podas o trasplantes.
- Contar con servicio de asistencia técnica para el usuario a través de la aplicación.
- Permitir la inclusión de sensores de PH para el sustrato de las macetas.
- Permitir el monitoreo visual remoto (a través de imágenes) para el control de plagas en las plantas.
- Agregar al catálogo de plantas, imágenes referenciales que indiquen deficiencias nutricionales en la planta como guía para el usuario.

Capítulo 6

Conclusiones y Trabajo a Futuro

Este trabajo de memoria se llevó a cabo con el objetivo de contar con una herramienta de apoyo al cultivo urbano, que permitiese el monitoreo remoto y el control automático del cultivo. Esta necesidad surge tanto por el aumento de personas que cultivan sus alimentos, como también por la reducción de los espacios para realizar esta actividad en las grandes urbes. A estos factores, se le suma el gran interés del alumno por los sistemas basados en internet de la cosas (*IoT*). Dicho esto, el realizar un proyecto de esta índole ha sido una meta personal para este memorista desde algunos años.

En el trabajo de memoria se desarrolló el MVP (Minimum Viable Product) de una aplicación móvil, la cual forma parte de un proyecto mayor que se desarrolla en conjunto con el alumno memorista Benjamín Zamora. Este último desarrolló el sistema de control para los cultivos, que complementa a la aplicación móvil aquí reportada.

La aplicación funciona integrada al sistema de control, y en conjunto permiten la gestión y control de tantos cultivos como sistemas de control se tengan. La aplicación móvil permite crear y monitorear ambientes de cultivos, recibir notificaciones, agregar tareas, ver en tiempo real el estado los elementos que componen el ambiente como macetas, plantas, sensores y actuadores. También permite hacer modificaciones a estos elementos. Además, cuenta con un catálogo de plantas y de comunidades para fomentar la interacción entre usuarios.

La aplicación móvil fue sometida a una evaluación que buscó medir su usabilidad y utilidad. En ésta participaron 8 personas, todos interesados en utilizar una aplicación de estas características para apoyar sus cultivos actuales o previstos. En la evaluación los usuarios siguieron un sketch (conjunto de operaciones), y luego completaron un formulario que evalúa la usabilidad y utilidad de la herramienta según la percepción de los usuarios.

Los resultados de la evaluación indicaron que la aplicación, tiene una usabilidad considerada “excelente” y una utilidad “buena”. Estos resultados son bastante satisfactorios para el tiempo de desarrollo empleado, y dado que esta es la primera versión del sistema. No obstante a lo anterior, los evaluadores sugirieron incluir algunos servicios bastante interesantes y útiles, lo que demuestra que aún queda bastante por desarrollar para lograr un software suficientemente completo y robusto como para comercializarlo.

El trabajo a futuro es amplio, y en primera instancia involucra el ajuste de los componentes para facilitar la puesta en producción del sistema. También se completará el desarrollo de funcionalidades que quedaron pendientes para una siguiente iteración, y se realizarán mejoras en la usabilidad de la aplicación que permitan llegar a usuarios poco habituados al uso de aplicaciones móviles. Finalmente, se analizará la adición de los nuevos servicios sugeridos por los evaluadores.

Las lecciones aprendidas durante este trabajo de memoria fueron varias, pues el alumno no contaban con experiencia previa en el desarrollo de aplicaciones móviles. La primera lección a mencionar ocurrió cuando éste se encontraba realizando la transición entre la etapa de diseño y de desarrollo del sistema, en la cual se enfrentó con el problema de no poder llevar a código los elementos definidos en el diseño, pues el framework utilizado para implementar no soportaba varias de las construcciones especificadas en el diseño. En ese sentido, queda claro que hay que considerar el framework de implementación al momento de realizar el diseño del sistema.

Otra lección aprendida fue que el uso de un framework relativamente nuevo, tiene sus implicancias negativas en el desarrollo. Por ejemplo, es usual que por ser un framework nuevo haya poca disponibilidad de información para resolver determinados problemas utilizando los recursos de dicho framework.

Por último, hubo también una lección aprendida referente a la estimación del tiempo de desarrollo, la cual falló rotundamente. Sin embargo, esto ayudó a entender que cuando se realiza la planificación del trabajo considerando el uso de tecnologías desconocidas, esto aumenta mucho la incertidumbre en los tiempos, por lo que se debe tener mucha cautela al momento de definirlos.

Bibliografía

- [1] Cayenne mydevices. <https://developers.mydevices.com/cayenne/features/>. Última visita: 4 de Enero de 2021.
- [2] Devicehive. <https://devicehive.com>. Última visita: 4 de Enero de 2021.
- [3] Flutter. <https://flutter-es.io>. Última visita: 4 de Enero de 2021.
- [4] Garden answers. <http://www.gardenanswers.com/intro>. Última visita: 4 de Enero de 2021.
- [5] Grafana. <https://grafana.com/>. Última visita: 5 de Enero de 2021.
- [6] Grobo. <https://www.grobo.io/>. Última visita: 4 de Enero de 2021.
- [7] Gro.io. <https://www.youtube.com/watch?v=cxLubGhclow>. Última visita: 4 de Enero de 2021.
- [8] Grow with jane. <https://play.google.com/store/apps/details>. Última visita: 4 de Enero de 2021.
- [9] Ionic. <https://ionicframework.com/>. Última visita: 2 de Diciembre de 2021.
- [10] Leaf. <https://www.getleaf.co/>. Última visita: 4 de Enero de 2021.
- [11] Mainflux. <https://www.mainflux.com>. Última visita: 4 de Enero de 2021.
- [12] Material design. <https://material.io/>. Última visita: 5 de Enero de 2021.
- [13] mygrow. <https://www.lamota.org/es/blog/apps-cultivo-marihuana/>. Última visita: 4 de Enero de 2021.
- [14] Niwa. <https://startupscolaborativas.com/niwa-huertos-urbanos/>. Última visita: 4 de Enero de 2021.
- [15] React native. <https://reactnative.dev>. Última visita: 2 de Diciembre de 2020.
- [16] Sensores (sensors) vs actuadores (actuators). https://www.tuataratech.com/2015/06/sensores-sensors-vs-actuadores-actuators_8.html. Última visita: 26 de Marzo de 2021.

- [17] Thingier.io. <https://thingier.io>. Última visita: 4 de Enero de 2021.
- [18] Thingspeak. <https://thingspeak.com>. Última visita: 4 de Enero de 2021.
- [19] Xamarin. <https://dotnet.microsoft.com/apps/xamarin>. Última visita: 2 de Diciembre de 2020.
- [20] Zetta. <https://www.zettajs.org>. Última visita: 4 de Enero de 2021.
- [21] ¿cuáles son las funciones de los sensores que incorporan los objetos con tecnología iot? <https://enviraiot.es/funciones-sensores-iot-cuales-son/>. Última visita: 5 de Enero de 2021.
- [22] ¿qué es arduino? <https://arduino.cl/que-es-arduino/>. Última visita: 5 de Enero de 2021.
- [23] ¿qué es raspberry pi? <https://raspberrypi.cl/raspberry/>. Última visita: 5 de Enero de 2021.
- [24] ¿qué es un sensor y qué hace? <https://dewesoft.com/es/daq/que-es-un-sensorwhat-do-sensors-do>. Última visita: 26 de Marzo de 2021.
- [25] Futuro 360. Huertos urbanos: La tendencia de cultivar en la ciudad. *Futuro360 - Desafío Tierra*, 2019. <https://www.futuro360.com/desafiotierra/huertos-urbanos-la-tendencia-de-cultivar-en-la-ciudad20190925/>. Última visita : 4 de Enero de 2021.
- [26] Amazon. Aws iot analytics. <https://aws.amazon.com/es/iot-analytics/>. Última visita: 4 de Enero de 2021.
- [27] Apache. Cordova. <https://cordova.apache.org/>. Última visita: 2 de Diciembre de 2020.
- [28] Fernanda Monasterio Blanco. La desigualdad del verde en Chile. *Pauta*, 2019. <https://www.pauta.cl/calidad-de-vida/estadisticas-parques-espacios-verdes-ciudades-regiones-de-chile>. Última visita: 4 de Enero de 2021.
- [29] John Brooke. Sus: A quick and dirty usability scale. *Usability Eval. Ind.*, 189, 11 1995.
- [30] John Brooke. Sus: a retrospective. *Journal of Usability Studies*, 8:29–40, 01 2013.
- [31] Simon Brown. The c4 model for software architecture. *InfoQ*, 2018. <https://www.infoq.com/articles/C4-architecture-model/>. Última visita: 1 de Marzo de 2021.
- [32] James Dean. How 5g technologies can be implemented more efficiently. *TechRadar*, December 2018. <https://www.techradar.com/news/how-5g-technologies-can-be-implemented-more-efficiently>. Última visita: 4 de Enero de 2021.
- [33] Jason Deign. Cannabis growers embrace iot for energy efficiency. *Greentech Media*, 2018. <https://www.greentechmedia.com/articles/read/cannabis-growers-embrace-iot-for-energy-efficiencygs.8Msu76tN>. Última visita: 4 de Enero de 2021.

- [34] Luis del Valle Hernández. Nodemcu tutorial paso a paso desde cero. <https://programarfacil.com/podcast/nodemcu-tutorial-paso-a-paso/>. Última visita: 5 de Enero de 2021.
- [35] Javier Flores. ¿qué es el 5g y cómo nos cambiará la vida? *National Geographic España*, 2020. https://www.nationalgeographic.com.es/ciencia/que-es-5g-y-como-nos-cambiara-vida_14449.htm *ltimavisita* : 4deEnero2021.
- [36] Niu G. and Masabni J. Plant production in controlled environments. *Horticulturae*, 4(28), September 2018.
- [37] Constanza Martínez Gaete. Estos huertos urbanos están haciendo florecer un santiago más verde. *El Definido*, 2018. <https://eldefinido.cl/actualidad/pais/9736/Estos-huertos-urbanos-estan-haciendo-floreecer-un-Santiago-mas-verde/>. Última visita: 4 de Enero de 2021.
- [38] N. Gondchawar and R.S. Kawitkar. Iot based smart agriculture. *International Journal of Advanced Research in Computer and Communication Engineering*, 5(8), June 2016.
- [39] María Gracia. Iot - internet of things. *Deloitte Touche Tohmatsu Limited*, 2021. <https://www2.deloitte.com/es/es/pages/technology/articles/IoT-internet-of-things.html>. Última visita: 4 de Enero de 2021.
- [40] Equipo Growpaltech. Growpal tech, brochure informativo growpal one. *Sensors*, 2019. <https://growpaltech.com/pages/manual-de-usuario>. Última visita: 4 de Enero de 2021.
- [41] Celina Gómez, Christopher Currey, Ryan Dickson, Hye-Ji Kim, Ricardo Hernández, Nadia Sabeh, Rosa Raudales, Robin Brumfield, Angela Laury-Shaw, Adam Wilke, Roberto Lopez, and Stephanie Burnett. Controlled environment food production for urban agriculture. *HortScience*, 54:1448–1458, 09 2019.
- [42] N. Harun, M. R. M. Kassim, I. Mat, and S. S. Ramli. Precision irrigation using wireless sensor network. *Proc. of the International Conference on Smart Sensors and Application (ICSSA)*, Kuala Lumpur:71–75, 2015.
- [43] Christopher Holloway. Plataformas iot: ¿qué son y cómo elegir la mejor para el negocio? *Itmastersmag*, 2018. <https://itmastersmag.com/noticias-analisis/plataformas-iot-que-son-y-como-elegir-la-mejor-para-el-negocio/>. Última visita: 4 de Enero de 2021.
- [44] T. Mercado A. Fernández J. García J. Gómez, S. Castaño. Sistema de internet de las cosas (iot) para el monitoreo de cultivos protegidos. *Ingeniería e Innovación*, 5(1), 2017.
- [45] C. Kamienski, J.P. Soininen, M. Taumberger, R. Dantas, A. Toscano, T.S. Cinotti, and A.T. Neto. Smart water management platform: Iot-based precision irrigation for agriculture. *Sensors*, 19(2):276, 2019.
- [46] S. Li. Application of the internet of things technology in precision agriculture irrigation systems. *2012 International Conference on Computer Science and Service System*, Nanjing:1009–1013, 2012.

- [47] Steff Manriquez. Iot el nuevo paradigma. *blogspot.com*, 2015. <http://iotgovernance.blogspot.com/2015/03>, Última visita: 4 de Enero de 2021.
- [48] Sophia Martin. 7 popular cross-platform app development tools that will rule in 2020. *Medium*, 2020. <https://medium.com/datadriveninvestor/7-popular-cross-platform-app-development-tools-that-will-rule-in-2020-349c80fb51>. Última visita: 3 de Diciembre de 2020.
- [49] S. Mohanty, D. Hughes, and M. Salathé. Using deep learning for image-based plant disease detection. *Frontiers in Plant Science* 7, April 2016.
- [50] Emiliano Carrizo Ortiz. La nueva lucha de las áreas verdes con el avance de la ciudad. *La Tercera*, 2019. <https://www.latercera.com/pulso/noticia/la-nueva-lucha-las-areas-verdes-avance-la-ciudad/728501/>. Última visita: 4 de Enero de 2021.
- [51] Nieves Pérez. Ventajas y desventajas del cultivo hidropónico frente al cultivo tradicional. *LeroyMerlin*, 2017. <https://comunidad.leroyermerlin.es/t5/Blog-de-la-Comunidad/Ventajas-y-desventajas-del-cultivo-hidrop-C3-B3nico-frente-al-cultivo/ba-p/194391>. Última visita: 4 de Enero de 2021.
- [52] Ramón García Soto. ¿es el desarrollo cross-platform la tendencia de 2014? *Exevi*, 2014. <https://www.exevi.com/desarrollo-cross-platform/>. Última visita: 2 de Diciembre de 2020.
- [53] Akash Takyar. Iot app development: Impact, challenges, and process. *Dzone*, 2021. <https://dzone.com/articles/iot-app-development-impact-challenges-and-process>. Última visita: 4 de Enero de 2021.
- [54] Benjamín Zamora. Sistema semi-automático de asistencia para cultivos en huertos urbanos y periurbanos. *Memoria de Ingeniería Civil en Computación (en evaluación por parte de la comisión de memoria)*, Departamento de Ciencias de la Computación, Universidad de Chile, 2021.