



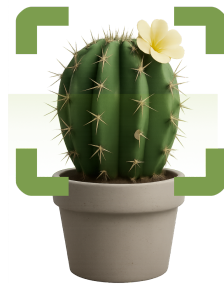
UNIVERSIDAD TECNOLÓGICA NACIONAL

Facultad Regional Córdoba

Ingeniería en Sistemas de Información

PROYECTO FINAL

PLANTITAPP



CURSO: 5K3

CICLO LECTIVO: 2025

PROFESORES

ING. SERGIO QUINTEROS

Mg. ING. VIRGINIA SANTOS

AUTORES

81907 - AMORMINO, MILENA JULIETA

85721 - CAPDEVILA, JUAN MANUEL

82454 - CARRERA ZANETTA, SEBASTIAN LUCA

87423 - DINIA JORGE, HASSAN

85920 - GOR, LEANDRO GABRIEL



PLANTITAPP

SOY PLANTITAPP, TU NUEVO ASISTENTE VERDE.
UN PRODUCTO QUE TE PERMITIRÁ IDENTIFICAR TUS PLANTAS, CONOCER
SUS CUIDADOS Y LLEVAR UN HISTORIAL DE SALUD.

Colección:

OBJETIVOS

Reconoce al instante cualquier planta de tu hogar usando **inteligencia artificial**.

Accede a **recomendaciones de riego, luz y tratamientos** contra enfermedades, mientras llevas un historial completo de la salud de tus plantas.

Conéctate con otros amantes de la botánica, localiza viveros cercanos y compartí experiencias, productos y **servicios** en una comunidad activa.



Monstera

COCINA



BENEFICIOS

PRACTICIDAD DIARIA 

 **CONEXIÓN**

APRENDIZAJE CONSTANTE 

 **ORGANIZACIÓN Y SEGUIMIENTO**

TECNOLOGÍAS



 **Universidad Tecnológica Nacional, Facultad Regional Córdoba**
Ingeniería en Sistemas de Información Cátedra: Proyecto Final. Curso: 5k4 - Año 2025

ALUMNOS

AMORMINO, MILENA JULIETA
CAPDEVILA, JUAN MANUEL
CARRERA ZANETTA, SEBASTIAN LUCA
DINIA, HASSAN
GOR, LEANDRO

CONTACTO

julietamormino@gmail.com
JMACapdevila@gmail.com
sebaluca.carrera@gmail.com
diniaghassan@gmail.com
leandrogabrielgor@gmail.com

Maestro M. López esq, Cruz Roja Argentina, Córdoba

PROFESORES

ING. VIRGINIA SANTOS
ING. SERGIO QUINTEROS



PLANTITAPP- Aplicación para el cuidado y gestión de plantas

Amormino Milena Julieta - Capdevila Juan Manuel - Carrera Zanetta

Sebastian Luca - Dinia Jorge Hassan - Gor Leandro Gabriel

Universidad Tecnológica Nacional, Facultad Regional Córdoba

Abstract

En este proyecto se desarrolló PlantitApp, una solución tecnológica destinada a facilitar el cuidado de plantas en entornos domésticos y urbanos para personas con conocimientos limitados en botánica o con escaso tiempo para su mantenimiento. El objetivo fue proporcionar una herramienta integral que combinara identificación de especies, recomendaciones personalizadas, registro de eventos y conexión con una comunidad activa. La metodología de desarrollo se basó en el marco ágil Scrum[1], definiendo roles, ceremonias y artefactos, y utilizando herramientas como Angular Ionic, Express.js y Supabase para la implementación y almacenamiento de datos. Entre los principales resultados, se logró integrar un sistema de reconocimiento vegetal mediante inteligencia artificial, un historial de salud por planta, recordatorios de cuidados, espacios de interacción comunitaria y acceso a viveros y prestadores de servicios. Estas funcionalidades contribuyeron a mejorar la organización, el seguimiento y la conservación de las plantas, así como a fomentar el intercambio de conocimientos y la colaboración entre usuarios. Se concluyó que PlantitApp representó una solución innovadora y accesible que potenció el vínculo de las personas con la naturaleza, ofreciendo una experiencia completa que combinó tecnología, educación y comunidad.

Palabras Clave

Plantas, Inteligencia artificial, Jardinería, Comunidad, Salud vegetal, Viveros, Producto.

Introducción

En la actualidad, muchas personas buscan incorporar plantas en sus hogares como una forma de conectar con la naturaleza, mejorar su bienestar y embellecer sus espacios. Sin embargo, el desconocimiento sobre los cuidados específicos que cada especie requiere, sumado al ritmo acelerado

de vida urbana, dificulta el mantenimiento adecuado de estas plantas. Esta situación conduce con frecuencia al deterioro de las mismas y a la frustración de los usuarios.

PlantitApp surge como una herramienta educativa y organizativa que permite a cualquier persona, sin importar su nivel de experiencia, aprender a cuidar sus plantas de forma efectiva. La aplicación identifica especies mediante inteligencia artificial, ofrece recomendaciones personalizadas, permite llevar un historial de salud vegetal, y organiza la información de cada planta según su ubicación, tipo y necesidades.

Además, promueve la creación de una comunidad activa en la que los usuarios comparten experiencias, intercambian esquejes y acceden a información de viveros y servicios cercanos. De esta manera, PlantitApp no solo facilita el cuidado individual, sino que también fortalece el vínculo entre la tecnología, la educación ambiental y la vida urbana.

Elementos de trabajo y metodología

Para el desarrollo de PlantitApp, se adoptó el marco de trabajo ágil Scrum debido a su capacidad de adaptación y a la necesidad de fomentar la colaboración constante entre los integrantes del equipo. Esta elección fue especialmente útil considerando que era la primera vez que el equipo trabajaba en conjunto en un proyecto de esta magnitud.

El proceso comenzó con un Sprint Cero, donde se definieron los roles, las ceremonias (Daily, Planning, Review y Retrospective), los artefactos (Product

Backlog, Sprint Backlog, Incremento) y las pautas para la documentación, estimaciones y codificación. Se planificaron sprints de 15 días, iniciando cada uno con una ceremonia de planificación donde se seleccionaron ítems del Product Backlog y se estimaron esfuerzos mediante la técnica “Poker Planning”[2].

Al finalizar cada sprint, se realizaba una revisión del trabajo completado y una retrospectiva para identificar oportunidades de mejora. El rol de Scrum Master fue rotativo entre los miembros del equipo, permitiendo que todos adquirieron experiencia en la gestión del proyecto.

La comunicación entre los miembros del equipo de desarrollo para las distintas reuniones se dió a través de WhatsApp y Discord[3]. Para la gestión del proyecto se utilizó la herramienta Jira[4], que permitió al equipo organizarse de forma autónoma y llevar un seguimiento detallado del trabajo. Con el fin de mantener el proyecto organizado e íntegro, se optó por Google Drive para almacenar la documentación y GitHub[5] para el control de versiones. El desarrollo del producto como aplicación web se llevó a cabo utilizando el editor de código Visual Studio Code[6]. Para el diseño de la interfaz gráfica se utilizó Angular con Ionic, mientras que el backend fue desarrollado con Express.js[7]. Finalmente, se eligió Supabase[8] como plataforma de backend-as-a-service[9], por su integración de base de datos, autenticación y almacenamiento en tiempo real.

Resultados

Durante el desarrollo de PlantitApp, se avanzó en la implementación de múltiples módulos funcionales que responden a las necesidades tanto de los usuarios aficionados al cuidado de plantas

(plantiamigos) como de los prestadores de servicios relacionados. A continuación, se detallan los principales resultados alcanzados:

Identificación y guardado de plantas: Se desarrolló un sistema de reconocimiento de especies vegetales mediante inteligencia artificial, que permitió a los usuarios identificar plantas con la cámara del celular y agregarlas a su colección personal.

Gestión del historial de salud: Para cada planta registrada se cuenta con un historial donde se pueden registrar eventos como riegos, podas, fertilizaciones y enfermedades. Además, se incorporó un sistema de diagnóstico preliminar basado en síntomas visibles, que sugiere posibles enfermedades y cuidados asociados.

Recomendaciones personalizadas: A partir de la especie, ubicación y condiciones ambientales, la aplicación genera recomendaciones específicas de riego, luz y cuidados generales, adaptadas a cada planta.

Gestión de recordatorios: Se implementó un sistema de notificaciones que recuerda a los usuarios cuándo regar, fertilizar o revisar el estado de sus plantas, tanto de forma general como específica por especie.

Participación y gestión de comunidades: Se habilitó un espacio para que los usuarios puedan unirse a comunidades temáticas, compartir experiencias, intercambiar esquejes y resolver dudas. Además, se desarrollaron herramientas de moderación para mantener un entorno seguro y colaborativo.

Administración de productos y servicios de prestadores: Los prestadores de servicios pueden registrar sus productos y servicios, gestionando su disponibilidad y visibilidad dentro de la app.

Consulta y calificación de prestadores: Los usuarios pueden buscar viveros y

prestadores cercanos mediante un mapa interactivo, consultar sus perfiles, contactarlos directamente y calificarlos según su experiencia.

Discusión

A lo largo del desarrollo de PlantitApp, se abordaron múltiples desafíos técnicos y conceptuales relacionados con el reconocimiento vegetal, la personalización del cuidado y la interacción comunitaria. Uno de los principales interrogantes que surgieron fue la precisión en la identificación de plantas y enfermedades, se reconoce que la precisión puede variar según la calidad de la imagen, la iluminación y la similitud entre especies.

Otro aspecto clave es la diferenciación frente a otras aplicaciones existentes. A diferencia de soluciones que se enfocan únicamente en la identificación de plantas, PlantitApp ofrece una experiencia integral: permitió organizar y gestionar colecciones personales, llevar un historial de salud vegetal, recibir recomendaciones personalizadas, interactuar con una comunidad activa y acceder a servicios locales.

Conclusión

PlantitApp se presenta como una solución innovadora y accesible para mejorar el cuidado de plantas en entornos urbanos. Su objetivo principal *“potenciar la forma en que las personas descubren, organizan y cuidan sus plantas tanto a nivel personal como dentro de una comunidad”* se ve reflejado en los módulos implementados y en la experiencia de usuario diseñada.

La aplicación no solo facilita el acceso a información botánica confiable, sino que también promueve la interacción entre aficionados, especialistas y comercios del rubro. A través de herramientas

tecnológicas modernas, PlantitApp contribuye a democratizar el conocimiento sobre jardinería, fortalecer redes de colaboración y mejorar la calidad de vida de las personas mediante el vínculo con la naturaleza.

Agradecimientos

Queremos agradecer a la cátedra de Proyecto Final por ser parte de este último tramo de la carrera, con una mención especial a Virginia Santos por acompañarnos en todo momento y por los valiosos consejos. A nuestras familias y amigos por el apoyo incondicional. Por último, a nuestra querida UTN FRC por darnos la posibilidad de formarnos como profesionales.

Referencias

- [1] Schwaber, K. y Sutherland, J. – Guía de Scrum. [Scrum.org](https://scrumguides.org/). 2020. Disponible en: <https://scrumguides.org/>.
- [2] Mountain Goat Software. Disponible en: <https://www.planningpoker.com/>.
- [3] Discord. Disponible en: <https://discord.com/>.
- [4] Atlassian. Disponible en: <https://www.atlassian.com/es/software/jira>.
- [5] GitHub. Disponible en: <https://github.com/about>.
- [6] Microsoft – Visual Studio Code. Disponible en: <https://code.visualstudio.com/>.
- [7] Express.js. Disponible en: <https://expressjs.com/es/>.
- [8] Supabase. Disponible en: <https://supabase.com/docs>.
- [9] Amazon Web Services. Disponible en: <https://aws.amazon.com/es/what-is/backend-as-a-service/>.

Datos de contacto:

Amormino Milena Julieta. Universidad Tecnológica Nacional- Facultad Regional Córdoba.

E-mail: julietamormino1@gmail.com

Capdevila Juan Manuel. Universidad Tecnológica Nacional- Facultad Regional Córdoba.

E-mail: jmacapdevila@gmail.com

Carrera Zanetta Sebastian Luca. Universidad Tecnológica Nacional- Facultad Regional Córdoba.

E-mail: sebaluca.carrera@gmail.com

Dinia Jorge Hassan. Universidad Tecnológica Nacional- Facultad Regional Córdoba.

E-mail: hassandinia02@gmail.com

Gor Leandro Gabriel. Universidad Tecnológica Nacional- Facultad Regional Córdoba.

E-mail: leandrogabrielgor@gmail.com

PLANILLA PARA CATALOGAR EL PROYECTO FINAL

AÑO	2025	CURSO Y NRO. DE GRUPO	5K3 - G15
NOMBRE DEL SISTEMA / PROYECTO			
PLANTITAPP			
CATEGORÍA (Solución De Negocio / Producto / Proyecto De Impacto Social)			
Producto			
HERRAMIENTAS Y TECNOLOGÍAS UTILIZADAS			
ÁMBITO DE APLICACIÓN		NOMBRE Y VERSIÓN	
ENTORNO DE DESARROLLO		Visual Studio Code v 1.103 Node.js v 20.17	
REPOSITORIOS Y VERSIONADO		Git en plataforma GitHub.	
PROGRAMACIÓN		Frontend: Angular v19, Backend: Express v4	
BASE DE DATOS		Supabase.	
COMUNICACIÓN INTERNA		Discord. Whatsapp.	
CAPACITACIÓN		Páginas web, Chat GPT, Gemini, Youtube, Documentación de tecnologías.	
PRUEBAS DE SISTEMA		Pruebas de caja blanca y caja negra. Pruebas manuales.	
GESTIÓN DEL PROYECTO		Jira.	
DOCUMENTACIÓN		Google Drive, Figma.	
MODELOS		UML 2.0.	