

Hanagotchi: aplicación para monitoreo y cuidado de plantas domésticas

Feijoo Sofia, Firmapaz Agustin, Pacheco Federico, Pérez Andrade
Violeta

August 13, 2024



Introducción y motivación

Agenda

- ▶ Estado del arte
- ▶ Solución implementada
- ▶ Pruebas y Validaciones
- ▶ Demo
- ▶ Riesgos
- ▶ Trabajo futuro
- ▶ Conclusiones

Jardines inteligentes



Automatización de tareas

Monitoreo y control sobre
condiciones ambientales

Integración con aplicaciones y
plataformas

Trabajos e investigaciones existentes



Yoolax



PlantNet



Vera



MiFlora



Growing Indoor Plants with Success

- ▶ Análisis de factores ambientales.
- ▶ Recomendaciones.
- ▶ **Sumarización de cuidados.**
- ▶ Ejemplo: temperatura
 1. **Frío** (18°C dia, 10°C noche)
 2. **Templado** (24°C dia, 18°C noche)
 3. **Calor** (30°C dia, 21°C noche)

Síntesis de características



Monitoreo mediante sensores



Base de datos de cuidados



Automatización de recordatorios



Sistema de bitácoras



Interacción amigable

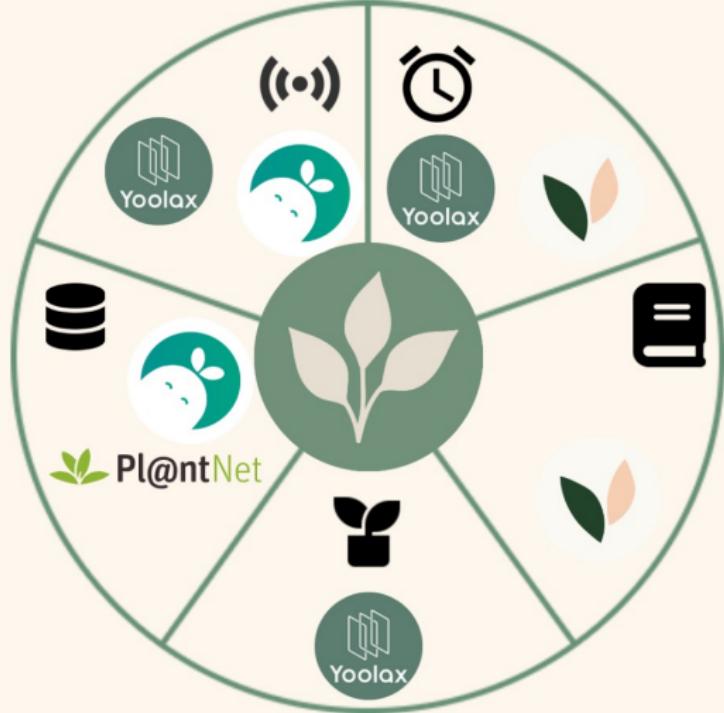
■ Sistema de bitácoras

(⌚) Monitoreo mediante sensores

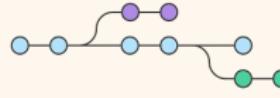
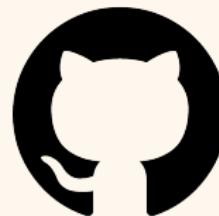
ในฐาน de datos de cuidados

⌚ Automatización de recordatorios

🌿 Interacción amigable



Metodologías



To Do	Doing	Done

Funcionalidades

Monitoreo de plantas domésticas

- ▶ **Seguimiento** de parámetros relevantes para el cuidado
- ▶ **Sensor integrado** que mide temperatura, luz y humedad de la tierra
- ▶ **Aproximaciones** basadas en la ubicación del dispositivo



Funcionalidades

Retroalimentación usuario-planta

- ▶ **Vínculo emocional** entre el usuario y el Hanagotchi
- ▶ El Hanagotchi reacciona a diversos **estímulos** del usuario y el ambiente
- ▶ El usuario puede dar **feedback**



Funcionalidades

Sistema de bitácoras

- ▶ **Historial** personalizado sobre la evolución de cada planta
- ▶ Incluye **texto y fotografías**



Funcionalidades

Red social

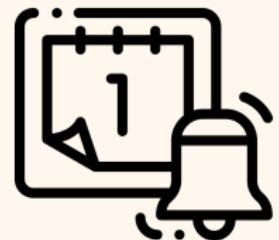
- ▶ **Interacción** entre usuarios
- ▶ **Posteos** que incluyen fotografías, consejos o consultas a la comunidad
- ▶ Búsqueda de temáticas de interés a través de **tags**



Funcionalidades

Sistema de recordatorios

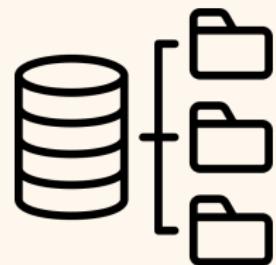
- ▶ **Notificaciones** push personalizables
- ▶ **Complementar** las alertas de la aplicación con tareas que el usuario considere relevantes



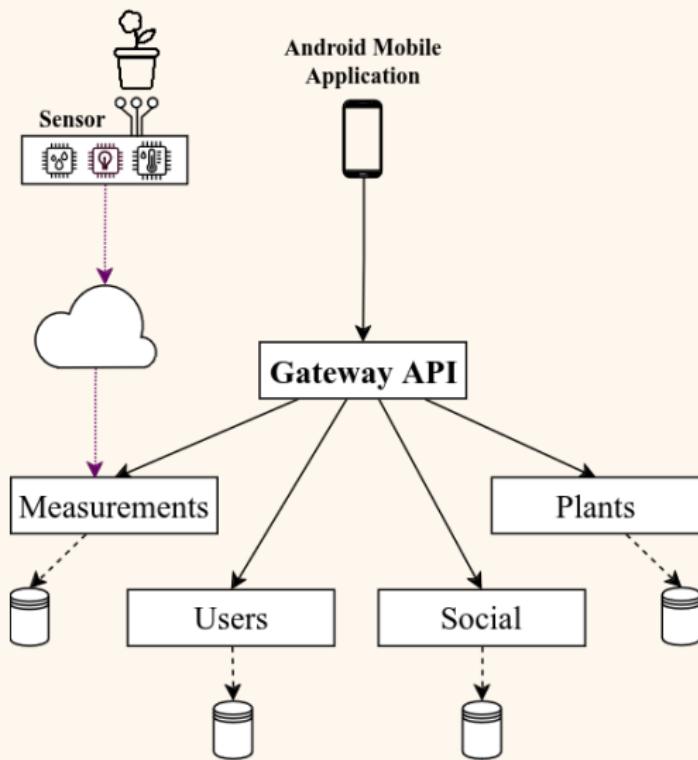
Funcionalidades

Dataset

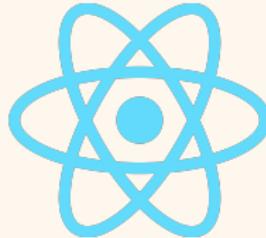
- ▶ **Procesamiento de los datos** de las mediciones y fotografías tomadas por los usuarios
- ▶ Generación de **fuentes de información** que pueden servir para futuras investigaciones realizadas por terceros



Arquitectura



Tecnologías



Implementación de hardware

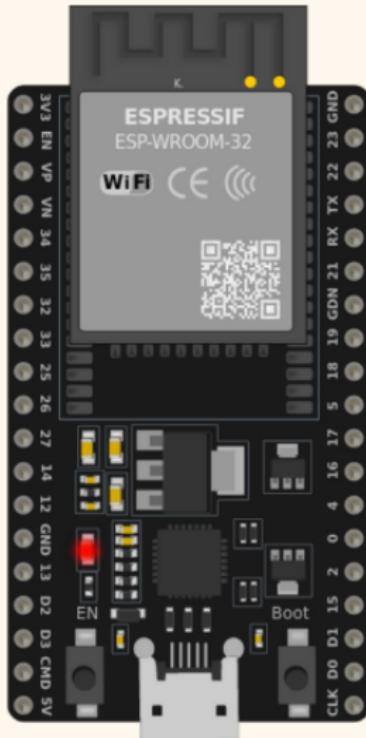
Sensor

Sensor



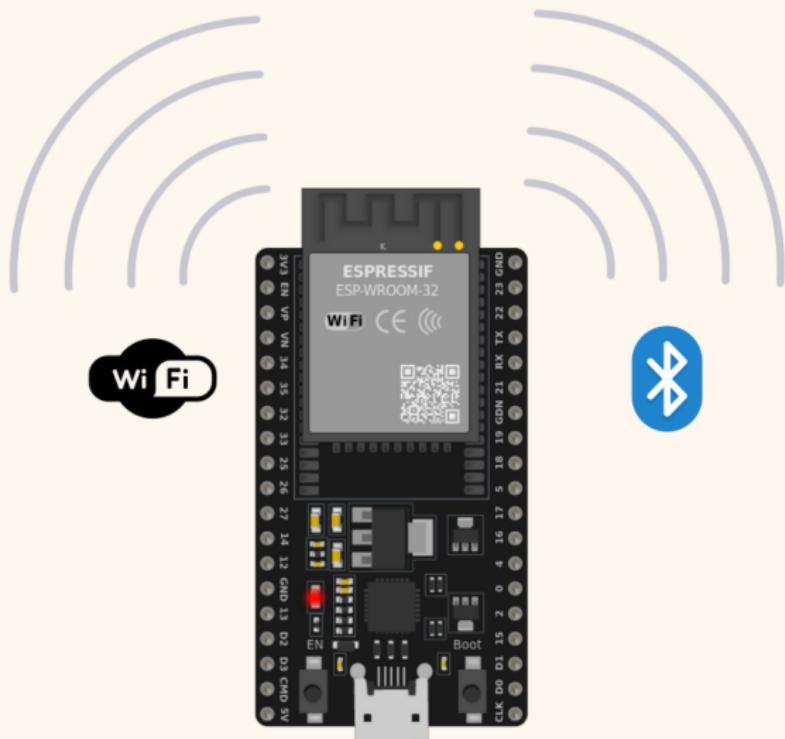
Sensor Xiaomi Flower Care

Sensor



Placa ESP32

Sensor



Placa ESP32 - Canales de comunicación

Sensor



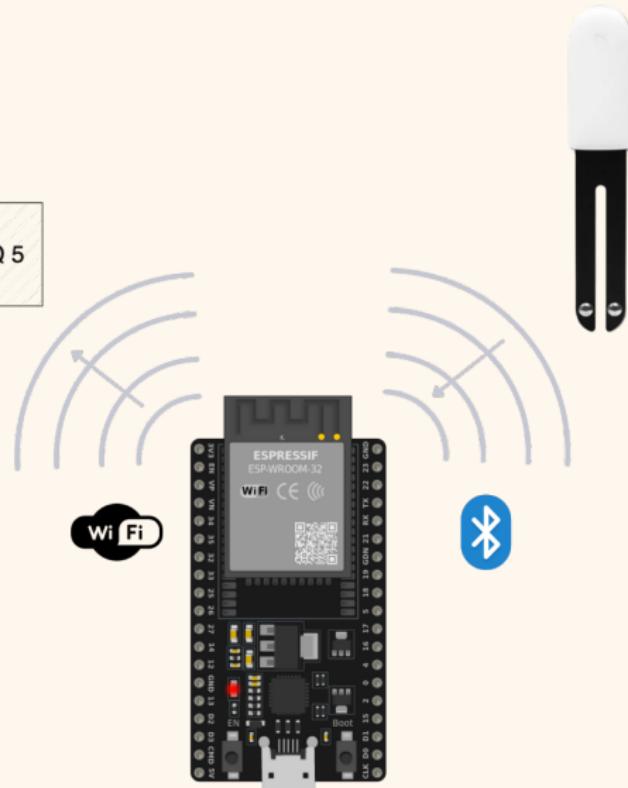
Sensor - conexión Wifi



Sensor

RabbitMQ

PAQ 1	PAQ 2	PAQ 3	PAQ 4	PAQ 5
-------	-------	-------	-------	-------



Sensor

Simulador

- ▶ Herramienta interna desarrollada para prescindir del hardware
- ▶ Imita comportamiento del sensor
- ▶ Dos posibles usos

Experimentación y/o validación



Pruebas basales



¿En qué consistieron?

- ▶ Pruebas E2E (end-to-end)
- ▶ Probar "a mano" todas las funcionalidades
- ▶ Total de 17 pruebas basales grabadas en video

Objetivo

- ▶ Integración completa de los componentes del sistema
- ▶ Escenarios con funcionalidades sean los esperados
- ▶ Correcto funcionamiento de la aplicación

Pruebas basales



Pruebas de campo

¿En qué consistieron?

- ▶ Compartir aplicación y hardware de sensorización
- ▶ Prueba de 2 a 3 días por usuario
- ▶ Recolección de feedback vía encuestas

Objetivo

- ▶ Evaluar y validar funcionamiento completo del sistema
- ▶ Evaluar criterios no funcionales (*usabilidad, simplicidad, robustez*)
- ▶ Obtener feedback de los usuarios



Análisis de resultados



Usabilidad e interfaz
de usuario



Validación de
funcionalidades



Nuevas funcionalidades
propuestas



Identificación de
problemas no detectados



Problemas de
rendimiento

Demo

Riesgos materializados

Riesgos materializados

Recordatorios personalizados

- ▶ **Contexto:** Implementación de sistema de recordatorios
- ▶ **Riesgo:** Esfuerzo muy alto
- ▶ **Materialización:** Implementación de un nuevo microservicio
- ▶ **Plan de respuesta:** Contratación de servicio de terceros
- ▶ **Lecciones aprendidas:** Evitar reinventar la rueda

Riesgos materializados

Hardware

- ▶ **Contexto:** Una placa ESP32 hace de intermediaria entre el sensor y nuestro sistema
- ▶ **Riesgo:** Limitaciones al trabajar en alto nivel
- ▶ **Materialización:** Formato y frecuencia de los paquetes enviados
- ▶ **Plan de respuesta:** Implementación de lógica adicional
- ▶ **Lecciones aprendidas:** Trabajar más a bajo nivel

Riesgos materializados

Plataforma de despliegue

- ▶ **Contexto:** Utilización de Heroku con planes académicos
- ▶ **Riesgo:** Limitaciones en los recursos
- ▶ **Materialización:** Saturación de los servidores
- ▶ **Plan de respuesta:** Upgrade de los planes contratados
- ▶ **Lecciones aprendidas:** Mejor gestión de recursos

Trabajos futuros

Trabajos futuros

Machine Learning

- ▶ Exportación de datos para futuras investigaciones.
- ▶ Estimación de factores ambientales a través de imágenes capturadas de la planta.
- ▶ Detección de enfermedades en plantas a través de imágenes capturadas de la planta.

Trabajos futuros

Hardware

- ▶ Configuración de la red WiFi del ESP32 desde la aplicación móvil.
- ▶ Acomodar la aplicación para que pueda ser utilizada con componentes de hardware que disponga el usuario.

Conclusiones

¿Preguntas?

Muchas gracias