PECL1. COMPUTACIÓN UBICUA

Pablo Acereda García David Emanuel Craciunescu
Pablo Martínez Gracia Laura Pérez Medeiro

October 2019

1. Introducción

El proyecto consistirá en un sistema inteligente para controlar los factores abioticos que afectan al bienestar de las plantas, con el objetivo de mejorar la salud de las mismas.

Para ello se monitorizarán la humedad y temperatura tanto exterior como de la tierra así como la cantidad de luz que reciben. Con estos valores y una serie de parámetros se generarán las corresponsientes alarmas que indiquen situaciones de riesgo para la salud de la planta.

2. Contexto

2.1. Situación actual del problema a abordar

Actualmente existen bastantes sistemas que se encargan de controlar el riego de las plantas y los cuales ofrecen la posibilidad de ser controlados mediante dispositivos móviles. Algunos de ellos son Blossom, PlantLink o Edyn. Incluso encontramos maceteros inteligentes, como el que nos ofrece Xiaomi, capaz de regar las plantas automáticamente.También encontramos otros proyectos como GRO, capaz de sugerirnos especies de plantas que podríamos cultivar en función del terreno que disponemos.
Sin embargo, ninguno nos ofrece integración con ningún asistente virtual como pueda ser Alexa, ese será la principal diferencia de nuestro proyecto con los anteriores.

2.2. Situación prevista al final del proyecto

Al finalizar la asignatura se pretende tener un dispositivo capaz de mantener con vida un cultivo de manera automática y el cual pueda ser controlado mediante comandos de voz gracias a Alexa. Para ello, nuestro dispositivo contará con la predicción del tiempo, aviso de posibles plagas, estado del terreno y condiciones óptimas necesarias para el tipo de planta que se posea.

2.3. Beneficiarios del proyecto

Con este proyecto se pretende ayudar a los aficionados de la jardinería que no disponen de una gran cantidad de tiempo para el óptimo cuidado de las plantas, así como a agricultores que necesiten una ayuda extra para minimizar el impacto de situaciones climáticas extremas. Por situaciones climáticas extremas se entienden, por ejemplo, heladas producidas en fechas inesperadas.

3. Alcance del proyecto

Como se ha descrito con anterioridad, el objetivo de este proyecto es obtener un producto que presentar a distintos socios inversores para su comercialización. Tal producto, denominado como '*ecø*', es el principal proyecto de la startup *Rainforest*.

'*Ecø*' es un producto que se comercializa tanto como un producto físico, como uno digital. Las funciones que realizará el producto físico serán las de monitorizar todos los factores ambientales que afectan a la planta (humedad, temperatura, luminosidad, calidad del suelo...) como base, aunque en un futuro, se incluirán sensores capaces de detectar la presencia de insectos que puedan resultar perjudiciales para la planta. Por otro lado, '*ecø*' contará con el respaldo de un producto software donde poder visualizar toda la información de manera clara y contar con el apoyo de una comunidad, con el fin de obtener una planta con cuidados óptimos. Además, de la comodidad de poder manejar '*ecø*' mediante un asistente de voz, con el que poder obtener la información sobre el estado de la planta así como regarla. Esas serán actualmentes las funciones que se están desarrollando, aunque en un futuro se esperaría poder ampliar la funcionalidad.

4. Descripción de ideas descartadas

Inicialmente se pensó en un sistema que fuera capaz de controlar las plagas que atacan a los cultivos, sin embargo, tal proyecto no es sencillo ya que la detección de los insectos no es sencilla. Se pensó en utilizar algún tipo de sensor con radiación infrarroja sin embargo, tras buscar información más detalla se ha comprobado que no sería eficaz. Se encontró un tipo de sensor llamado PIR, el cual podría regularse los parámetros del sensor para detectar animales de pequeño tamaño. Temporalmente, esta idea ha sido aplazada.

Otra idea descartada fue la de crear nosotros mismos el microcontrolador a utilizar para el proyecto, esta idea fue descartada por ser demasiado ambiciosa valorando el tiempo disponible para su desarrollo.

5. Tecnología a utilizar

Las decisiones acerca de la tecnología a utilizar en el proyecto se han tomado teniendo en cuenta la experiencia de los distintos miembros que conforman el área de desarrollo, así como el costo que tienen los distintos componentes.
Las distintas opciones barajadas para la realización del proyecto han sido:

Hardware

Controladoras

- Arduino: Ofrece una amplia gama de placas, con distintas prestaciones, cuenta con un entorno de programación para la creación de aplicaciones y una gran comunidad que apoya el Desarrollo. El principal inconveniente que se encontraba era su precio, el cual ronda los 20 euros.
- Raspberry: Fue una opción que se estuvo barajando utilizar, ya que ofrece una amplia variedad de lenguajes de programación a utilizar. Sin embargo, fue descartada debido a su gran tamaño.
- ESP32: Ha sido la opción finalmente elegida, ya que se trata de un SoC de muy bajo coste (su precio ronda los 5 euros), cuenta con Bluetooth y es utilizado en multitud de proyectos de IoT. Además, ofrece compatibilidad con Arduino, pudiendo utilizar los amplios recursos desarrollados por esa comunidad.

Sensores

- DHT22: Consiste en un sensor analógico para medir valores de humedad y temperatura cuya señal de salida es digital. Se eligió este modelo debido a su tamaño, factor bastante importante para nuestro proyecto, su precisión y el precio del mismo.
  Este sensor será utilizado para medir la temperatura y humedad ambiental.
- LDR: Es un fotoresistor, con el cual mediremos la cantidad de luz recibida por la planta.
- YL-69: Sensor de humedad de la tierra, nos servirá como activador del sistema de riego o lanzador de notificación para que el usuario conozca que su planta necesita ser regada.

Lenguaje de programación

Dado que la microcontroladora elegida ha sido la ESP32, las opciones a la hora de programar disponibles son:

- MicroPython: consiste en un pequeño intérprete de Python, el cual contiene un subconjunto mínimo y optimizado de librerias para que pueda correr en microcontroladores. En un primer momento, fue la idea elegida para utilizar en nuestro proyecto.
  Ventajas:
  - RELP Interactiva, es decir, un programa que permite leer e interpretar los comandos para evaluarlos e imprimir el resultado sin la necesidad de compilar ni cargar el programa en el microcontrolador.
  - Gran cantidad de librerías disponibles
  - Extensibilidad. Ofrece la posibilidad de mezclas códigod que requieran una ejecución más rápida a bajo nivel mediante la extensión de sus funciones.

Desventajas:
  - Código más lento y con necesidad de mayor cantidad de memoria, en comparación con C o C++
  - Proceso de inicialización del microcontrolador más complejo
  - Funcionalidad de ciertas librerías para componentes más limitadas
- RTOS: es un sistema de operación en tiempo realutilizado en sistemas embebidos. Proporciona métodos para múltiples subprocesos o hilos, mutexes, semáforos, temporizadores... Además de soportar las prioridades de los hilos. Por último, cuenta con un modo que reduce el consumo energético (tickless). En FreeRTOS las aplicaciones puede asignarse de manera estática, mientras que los objetos pueden asignarse dinámicamente con distintos esquemas de asignación de memoria. Pese a contar con una enorme cantidad de documentación, la complejidad y la curva de aprendizaje que supondría el uso de este lenguaje hizo que fuera descartado rápidamente.
- Mongoose OS: framework de desarrollo disponible bajo la licencia de Apache (con una versión community y otra enterprise) y amplaimente utilizado en aplicaciones relacionadas con el IoT. Cuenta con compatibilidad para microcontroladores de bajo consumo, como es el caso del ESP32. Cuenta con un servidor web integrado, soporta programación tanto en C como en JavaScript y cuenta con integración de nubes privadas y públicas (como AWS Iot o Mosquitto).
- IDE Arduino: Software de código abierto que cuenta con una amplia comunidad de respaldo. Los lenguajes que admite son C y C++ haciendo uso de reglas especiales de estructuración de código, además de que bajo este IDE se suministran multitud de bibliotecas para procedimientos comunes de entrada y salida. Finalmente ha sido la opción elegida para la programación de la ESP32, ya que la forma de empezar a usar este software con la placa es muy sencilla.

6. Metodología de desarrollo

La metodología escogida para la realización del proyecto ha sido Lean más Kanban con ligeras modificaciones. Los motivos que impulsaron la elección de esta metodología fueron los principios que sigue:

1. Eliminación de desperdicios. Todo aquello que no aporte valor debe eliminarse (códigos basura, requisitos modificados...)
2. Ampliación del aprendizaje. Aunque inicialmente cada miembro del equipo se encargará del desarrollo de una parte del proyecto, todo el equipo está abierto a ofrecer su ayuda a los demás y a aprender nuevas tecnologías o formas de resolver problemas.
3. Realizar las tomas de decisiones a medida que se van teniendo en cuenta los requisitos a cumplir, así como la manera de enfocarlos. El proyecto ha ido construyéndose de manera progresiva, no se ha seguido una idea fija desde el primer momento sino que esta se ha ido desarrollando y madurando con el tiempo.

4. Entregas rápidas. Cada vez que se realiza una entrega esta implica un aumento de la funcionalidad o una corrección de esta.
5. Potenciar el equipo, todos los miembros han sido partícipes en la toma de decisiones importantes para el proyecto.
6. Creación de integridad, haciendo uso de un sistema para el control de versiones (github en nuestro caso).
7. Visualizar todo en conjunto. *"Pensar en grande, actuar en pequeño, equivocarse rápido y aprender con rapidez"* Esa frase resume la mentalidad de todos los integrantes del equipo.

Esta metodología se ha apoyado con Kanban, el cual no es una metodología de trabajo en sí sino una forma de visualizar el trabajo, controlar la asignación de tareas y mejorar la comunicación de los miembros del equipo.

Lean Kanban es una metodología seguida por varías startups, como IMVU startup creada por Eric Ries quien creó una metodología llamada Lean Startup que se basa principalmente en Lean con Kanban pero con ciertos aspectos más centralizados en el funcionamiento de una startup.

Los ciclos del desarrollo comienzan con el lanzamiento de una idea, la cuál es desarrollada y valorada, se obtiene un aprendizaje con los datos arrojados por esta idea y se incorpora al producto. De esta forma se inició el proyecto con la intención de crear un sistema de detección de riesgos potenciales para los cultivos (clima y plagas principalmente) y se pensó en incluir alexa en el control de este sistema.

6.1. Aplicación de la metodología

Para aplicar la metodología de manera eficaz se realizaba semanalmente una breve exposición de los avances realizados (mejoras realizadas, problemas encontrados...), así como un nuevo reparto de tareas y la exposición de alguna nueva idea aplicable al proyecto.

Todo la actividad quedaba registrada en github mediante los commit y el dashboard con las distintas actividades.

7. Arquitectura de la aplicación

DIAGRAMA DE PABLO + EXPLICACIÓN DE ESTE

8. Modelo de negocio

MODELO CANVAS AQUÍ

9. Planificación temporal y de desarrollo

La planificación se ha ido elaborando semanalmente, según se tenía mayor conocimiento de las labores a realizar. Estas tareas y el encargado de realizarlas se refleja en el dashboard que nos ofrece github. En líneas generales, para noviembre se contaría con toda la información relativa a los sensores a utilizar, el esquema de las conexiones a realizar y un servidor donde almacenar los datos de los sensores. Así mismo, se avanzaría con la interfaz de voz de Alexa mediante la realización de un curso para tener un mayor conocimiento acerca de su funcionamiento y la manera de integrarlo en el proyecto.

10. Riesgos del proyecto

Cosos de Pablo de seguridad

11. Plan de contingencias

12. Resumen y conclusiones

La Antorcha Humana fue a bankia y le denegaron una hipoteca.