

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS

FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

ESCUELA PROFESIONAL ACADÉMICA DE INGENIERÍA DE SOFTWARE



INTERNET DE LAS COSAS

Trabajo:

Jardín Inteligente: Un Sistema IoT para el Cuidado Preciso de Plantas

Docentes

José Herrera

Yessica Rosas

Integrantes

Ovalle Martinez, Lisett Andrea

Bejar, Manuel

Monterola Astorayme, Max Antony

Lima, Perú

2024-1

ÍNDICE

INTRODUCCIÓN.....	3
PROBLEMÁTICA.....	4
OBJETIVOS.....	5
1. Objetivo General.....	5
2. Objetivos Específicos.....	5
DESCRIPCIÓN DEL SISTEMA.....	6
REQUERIMIENTOS DEL SISTEMA.....	6
REQUERIMIENTOS NO FUNCIONALES.....	6
FUNCIONAMIENTO DEL SISTEMA.....	6
ARQUITECTURA.....	7
RESULTADOS.....	7
BENEFICIOS.....	7
Eficiencia en el Uso del Agua.....	7
Cuidado Personalizado de las Plantas.....	7
Monitoreo y Control Remoto Avanzado.....	7
Ahorro de Tiempo y Simplificación del Mantenimiento.....	7
CONCLUSIONES.....	8
FUTURAS MEJORAS Y RECOMENDACIONES.....	8
Integración con Asistentes de Voz.....	8
Análisis Predictivo y Machine Learning.....	8
Mejora Continua de la Interfaz de Usuario.....	8
Implementación de Funcionalidades Ambientales Adicionales.....	9

INTRODUCCIÓN

La jardinería es una actividad que no solo embellece el entorno, sino que también proporciona beneficios emocionales y psicológicos a quienes la practican. Sin embargo, el cuidado adecuado de las plantas requiere conocimientos específicos y atención constante, lo que puede resultar desafiante para muchas personas, especialmente en un mundo donde el tiempo es un recurso limitado. La incorporación de tecnologías avanzadas en la jardinería puede revolucionar la manera en que cuidamos de nuestras plantas, haciéndola más accesible y eficiente.

El concepto de Internet de las Cosas (IoT) ha ganado una considerable atención en los últimos años debido a su capacidad para conectar dispositivos y sistemas a través de Internet, permitiendo la comunicación y el intercambio de datos en tiempo real. La aplicación de IoT en la jardinería abre nuevas posibilidades para el monitoreo y control precisos de las condiciones ambientales y el estado de las plantas, lo que a su vez facilita el mantenimiento de jardines más saludables y sostenibles.

El presente informe se centra en el desarrollo e implementación de un sistema de Jardín Inteligente que aprovecha las capacidades del IoT para proporcionar un cuidado preciso y automatizado de las plantas. Este sistema está diseñado para recolectar datos relevantes sobre el entorno del jardín mediante una serie de sensores, procesar estos datos y tomar decisiones informadas para optimizar el riego y otros aspectos cruciales del cuidado de las plantas. Además, permite a los usuarios monitorear y controlar su jardín de manera remota a través de una aplicación móvil, haciendo la jardinería más accesible y conveniente.

El Jardín Inteligente no solo aborda las necesidades de los entusiastas de la jardinería, sino que también ofrece soluciones significativas para aquellos que buscan implementar prácticas sostenibles en sus espacios verdes. Al optimizar el uso del agua y proporcionar cuidados personalizados para cada planta, el sistema contribuye a la conservación de recursos naturales y mejora la eficiencia del mantenimiento del jardín.

Este informe detalla los objetivos, la descripción del sistema, sus componentes y funcionamiento, así como los beneficios que ofrece. También se presentan los resultados obtenidos de la implementación del sistema, una discusión sobre sus implicaciones y posibles mejoras futuras para ampliar su funcionalidad y alcance. Con esta tecnología, la jardinería puede transformarse de una actividad laboriosa a una experiencia enriquecedora y sostenible, beneficiando tanto a las plantas como a los cuidadores.

PROBLEMÁTICA

El cuidado de las plantas y jardines presenta varios desafíos que pueden dificultar el mantenimiento y la salud óptima de las plantas. Entre las principales problemáticas se encuentran:

1. **Desconocimiento y Falta de Experiencia:** Muchas personas carecen del conocimiento y la experiencia necesarios para proporcionar el cuidado adecuado a diferentes tipos de plantas. Cada planta tiene requerimientos específicos en términos de agua, luz solar, temperatura y nutrientes, y no conocer estos requisitos puede llevar al deterioro de la salud de las plantas.
2. **Tiempo Limitado para el Cuidado:** En un mundo acelerado, las personas a menudo tienen poco tiempo para dedicarse al mantenimiento regular de sus jardines. Esto puede resultar en riegos insuficientes o excesivos, falta de monitoreo de las condiciones del suelo y del ambiente, y descuido general del jardín.
3. **Uso Ineficiente del Agua:** El riego excesivo o insuficiente es un problema común en la jardinería. El uso ineficiente del agua no solo afecta negativamente a las plantas, sino que también representa un desperdicio de recursos naturales y puede aumentar los costos de mantenimiento.
4. **Monitoreo Inadecuado:** La falta de monitoreo constante de las condiciones ambientales y del suelo puede impedir la identificación oportuna de problemas como la sequía, enfermedades de las plantas o infestaciones de plagas.
5. **Cuidado No Personalizado:** Las soluciones de jardinería generalizadas no siempre se adaptan a las necesidades específicas de cada planta. Un enfoque personalizado es esencial para asegurar que cada planta reciba el cuidado adecuado según sus requerimientos únicos.
6. **Acceso Remoto:** La imposibilidad de monitorear y controlar el jardín de manera remota limita la capacidad de los propietarios para responder a las necesidades del jardín cuando no están en casa.

Estas problemáticas subrayan la necesidad de una solución avanzada que facilite el cuidado preciso y automatizado de las plantas, optimizando el uso de recursos y mejorando la salud general del jardín.

OBJETIVOS

1. Objetivo General

Desarrollar e implementar un sistema de Jardín Inteligente basado en tecnología IoT para optimizar el cuidado de plantas mediante el monitoreo y control precisos de las condiciones ambientales, automatizando el riego y otras tareas de mantenimiento, y permitiendo la supervisión remota a través de una aplicación.

2. Objetivos Específicos

- **Implementar Sensores de Monitoreo:** Desplegar sensores en el jardín para medir la humedad del suelo y detectar la presencia de lluvia en tiempo real.
- **Desarrollar el Sistema de Procesamiento de Datos:** Crear un sistema centralizado que procese los datos recolectados por los sensores y tome decisiones automatizadas sobre el riego.
- **Automatizar el Riego:** Diseñar y configurar un sistema de riego automatizado, utilizando relés y una bomba de agua, que se active basado en los datos de humedad del suelo y la presencia de lluvia.
- **Desarrollar una Aplicación:** Crear una aplicación que permita a los usuarios monitorear las condiciones del jardín, recibir alertas y controlar el sistema de riego.
- **Realizar Pruebas y Validación del Sistema:** Llevar a cabo pruebas exhaustivas del sistema en diferentes condiciones para validar su efectividad y realizar ajustes necesarios antes de su implementación definitiva.
- **Optimizar el Uso del Agua:** Implementar estrategias para maximizar la eficiencia en el uso del agua, reduciendo el desperdicio y asegurando que cada planta reciba la cantidad adecuada de riego.

DESCRIPCIÓN DEL SISTEMA

REQUERIMIENTOS DEL SISTEMA

El sistema de Jardín Inteligente está compuesto por los siguientes elementos:

- **Sensor de Lluvia:** Para detectar la presencia de lluvia y evitar el riego innecesario.
- **Sensor de Humedad del Suelo:** Para medir la cantidad de agua presente en el suelo.
- **Relay:** Para controlar el encendido y apagado de la bomba de agua.
- **Bomba para el Agua:** Para proporcionar riego automatizado según las necesidades detectadas.
- **ESP32:** Como microcontrolador central para procesar los datos de los sensores y controlar los actuadores.
- **Batería con Entrada de 4 Pilas:** Para proporcionar energía al sistema.
- **Aplicación:** El sistema debe permitir el monitoreo remoto de las condiciones del jardín y el control de la bomba de agua a través de una aplicación

REQUERIMIENTOS NO FUNCIONALES

- **Conectividad:**

- El sistema debe tener conectividad Wi-Fi para permitir la transmisión de datos a la nube y la comunicación con la aplicación.
- **Eficiencia Energética:**
 - El sistema debe ser eficiente en el uso de energía, utilizando una batería con entrada de 4 pilas para alimentar el ESP32 y los sensores.
- **Escalabilidad:**
 - El diseño del sistema debe permitir la fácil integración de más sensores y actuadores en el futuro si se requiere ampliar el sistema.
- **Interfaz de Usuario:**
 - La aplicación móvil debe tener una interfaz intuitiva y fácil de usar, facilitando la interacción del usuario con el sistema.

FUNCIONAMIENTO DEL SISTEMA

- **Recolección de Datos:** Los sensores instalados en el jardín recolectan datos en tiempo real sobre las condiciones del entorno.
- **Procesamiento de Datos:** El ESP32 procesa estos datos y los envía a una plataforma en la nube.
- **Toma de Decisiones:** Basado en los datos recopilados, el sistema toma decisiones automatizadas sobre el riego y otros cuidados necesarios.
- **Interacción del Usuario:** Los usuarios pueden monitorear el estado del jardín y controlar el sistema a través de una aplicación.

ARQUITECTURA

CÓDIGO

ARDUINO

Este es un dispositivo basado en el microcontrolador ESP32 que se conecta a una red WiFi, lee datos de un sensor de humedad del suelo y un sensor de lluvia, y envía estos datos a un servidor a través de HTTP. Además, controla una bomba de agua basada en las lecturas del sensor de humedad. A continuación se explica cada parte del código:

Declaración de Bibliotecas y Variables

```
#include <WiFi.h>
#include <HttpClient.h>
#include <ArduinoJson.h>
```

➤ **#include <WiFi.h>:** Incluye la biblioteca para conectarse a redes WiFi.

- `#include <HTTPClient.h>`: Incluye la biblioteca para realizar solicitudes HTTP.
- `#include <ArduinoJson.h>`: Incluye la biblioteca para manejar datos en formato JSON.

```

5  const char* ssid = "A23"; // Nombre de tu red WiFi
6  const char* password = "dneg1050"; // Contraseña de tu red WiFi
7
8  const char* serverUrl = "https://servidor-iot-seven.vercel.app/api/data"; // URL del
9
10 const int humedadSensorPin = 34; // Pin D34 conectado al sensor de humedad
11 const int lluviaSensorPin = 18; // Pin A0 conectado al sensor de lluvia
12 const int bombaPin = 2; // Pin D2 conectado a la bomba de agua
13
14 const int umbralHumedad = 4000; // Parámetro establecido para la humedad
15

```

- `ssid` y `password`: Credenciales de la red WiFi.
- `serverUrl`: URL del servidor al que se enviarán los datos.
- `humedadSensorPin`, `lluviaSensorPin`, `bombaPin`: Pines del ESP32 conectados al sensor de humedad, sensor de lluvia y bomba de agua, respectivamente.
- `umbralHumedad`: Valor umbral para la humedad del suelo que determina cuándo activar la bomba de agua.

CONFIGURACIÓN INICIAL

```

void setup() {

    Serial.begin(115200);
    delay(1000);

    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        delay(1000);
        Serial.println("Conectando a WiFi...");
    }
    Serial.println("Conectado a WiFi");

    pinMode(humedadSensorPin, INPUT);
    pinMode(lluviaSensorPin, INPUT);
    pinMode(bombaPin, OUTPUT);
    digitalWrite(bombaPin, HIGH); // Asegurarse
}

```

- `Serial.begin(115200)`: Inicializa la comunicación serial a 115200 bps para la depuración.
- `WiFi.begin(ssid, password)`: Conecta el ESP32 a la red WiFi especificada.
- `while (WiFi.status() != WL_CONNECTED)`: Espera hasta que se establezca la conexión WiFi.

- `pinMode(humedadSensorPin, INPUT), pinMode(lluviaSensorPin, INPUT), pinMode(bombaPin, OUTPUT)`: Configura los pines para los sensores y la bomba.
- `digitalWrite(bombaPin, HIGH)`: Asegura que la bomba de agua esté apagada inicialmente.

BUCLE

```
void loop() {
    int humedad = analogRead(humedadSensorPin);
    int lluvia = analogRead(lluviaSensorPin);

    Serial.print("Humedad: ");
    Serial.println(humedad);

    Serial.print("Lluvia: ");
    Serial.println(lluvia);

    StaticJsonDocument<200> jsonDoc;
    jsonDoc["humedad"] = humedad;
    jsonDoc["lluvia"] = lluvia;

    String jsonString;
    serializeJson(jsonDoc, jsonString);

    if(WiFi.status() == WL_CONNECTED) {
        HTTPClient http;
        http.begin(serverUrl);
        http.addHeader("Content-Type", "application/json");

        int httpResponseCode = http.POST(jsonString);

        if (httpResponseCode > 0) {
            Serial.print("HTTP Response code: ");
            Serial.println(httpResponseCode);
        } else {
            Serial.print("Error code: ");
            Serial.println(httpResponseCode);
        }
    }
}
```

- `analogRead(humedadSensorPin)` y `analogRead(lluviaSensorPin)`: Lee los valores de los sensores de humedad y lluvia.
- `Serial.print` y `Serial.println`: Imprime los valores de humedad y lluvia para la depuración.
- `StaticJsonDocument<200> jsonDoc`: Crea un documento JSON para almacenar los datos de los sensores.

- `jsonDoc["humedad"] = humedad` y `jsonDoc["lluvia"] = lluvia`: Asigna los valores de los sensores al documento JSON.
- `serializeJson(jsonDoc, jsonString)`: Convierte el documento JSON a una cadena de texto.
- `if (WiFi.status() == WL_CONNECTED)`: Verifica si el dispositivo está conectado a WiFi.
- `HTTPClient http`: Crea un objeto `HTTPClient` para enviar solicitudes HTTP.
- `http.begin(serverUrl)`: Inicializa la solicitud HTTP con la URL del servidor.
- `http.addHeader("Content-Type", "application/json")`: Agrega un encabezado de tipo de contenido JSON a la solicitud.
- `int httpResponseCode = http.POST(jsonString)`: Envía los datos JSON al servidor usando el método POST y obtiene el código de respuesta HTTP.
- `if (httpResponseCode > 0)`: Verifica si la solicitud fue exitosa y imprime el código de respuesta.

```

    http.end();
  } else {
    Serial.println("WiFi Desconectado");
  }

  if (humedad > umbralHumedad) {
    digitalWrite(bombaPin, LOW); // Encender la bomba de agua
    delay(1000); // Mantener la bomba encendida por 1 segundo
    digitalWrite(bombaPin, HIGH); // Apagar la bomba de agua
    Serial.println("Bomba de agua activada");
  }

  delay(10000); // Espera 10 segundos antes de la próxima
}

```

- `http.end()`: Finaliza la conexión HTTP.
- `if (humedad > umbralHumedad)`: Verifica si la humedad del suelo supera el umbral establecido.
- `digitalWrite(bombaPin, LOW)`: Enciende la bomba de agua.
- `delay(1000)`: Mantiene la bomba encendida durante 1 segundo.
- `digitalWrite(bombaPin, HIGH)`: Apaga la bomba de agua.
- `delay(10000)`: Espera 10 segundos antes de realizar la próxima lectura de los sensores.

APLICATIVO

Main

Es un servidor web básico utilizando Flask, un framework de microservicio web en Python. El servidor recibe y gestiona datos de sensores a través de solicitudes HTTP. A continuación se explica cada parte del código:

Importaciones y Configuración

```
from flask import Flask, request, jsonify, render_template

app = Flask(__name__)

sensor_data = {
    'humedad': 0,
    'lluvia': 0
}
```

Importa las bibliotecas necesarias.

Crea una instancia de la aplicación Flask.

Define un diccionario global para almacenar datos del sensor.

Ruta API para Datos del Sensor

```
@app.route('/api/data', methods=['POST', 'GET'])
def receive_data():
    global sensor_data

    if request.method == 'POST':
        data = request.json
        if 'humedad' in data:
            sensor_data['humedad'] = data['humedad']
        if 'lluvia' in data:
            sensor_data['lluvia'] = data['lluvia']
        return jsonify(sensor_data), 200

    elif request.method == 'GET':
        return jsonify(sensor_data), 200
```

Define la ruta `/api/data` para manejar solicitudes **POST** y **GET**.

POST: Actualiza `sensor_data` con los datos recibidos y devuelve los datos actualizados.

GET: Devuelve los datos actuales del sensor en formato JSON.

Ruta para Página Principal

```
@app.route('/')
def index():
    return render_template('index.html')
```

Define la ruta principal `/` para renderizar la plantilla `index.html`.

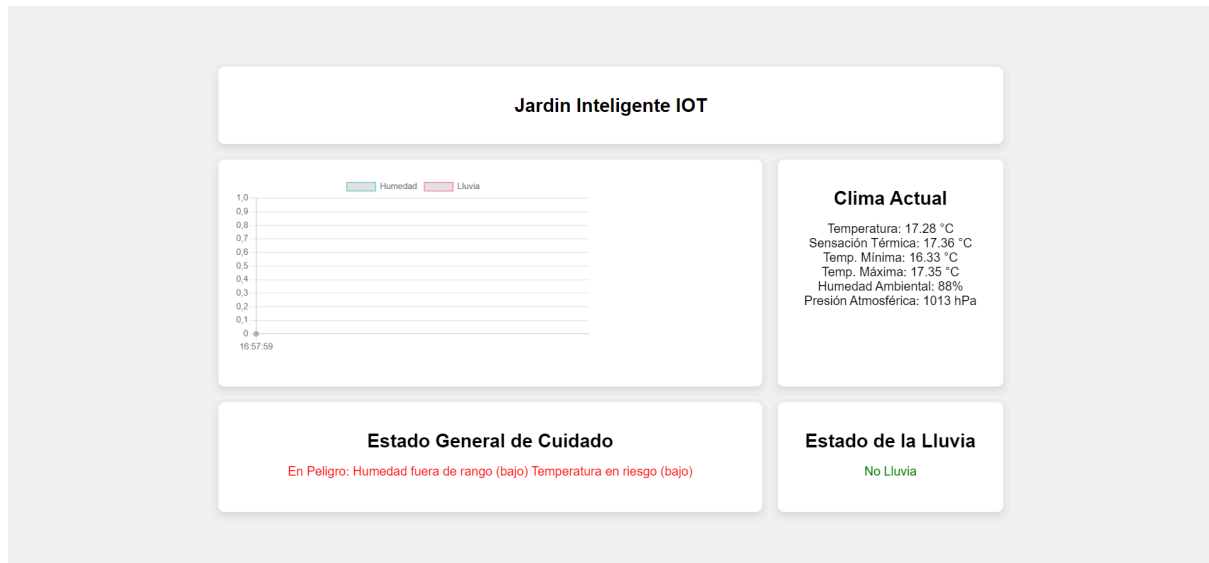
Ejecución del Servidor

```
if __name__ == '__main__':  
    app.run(debug=True)
```

Inicia el servidor Flask en modo debug para facilitar el desarrollo.

RESULTADOS

JARDÍN INTELIGENTE



BENEFICIOS

Eficiencia en el Uso del Agua

El sistema de Jardín Inteligente no solo automatiza el riego basado en la humedad del suelo, sino que también incorpora algoritmos para optimizar aún más el uso del agua. Utilizando datos precisos recogidos por sensores, el sistema ajusta dinámicamente los ciclos de riego en respuesta a cambios ambientales y patrones de consumo de agua de las plantas. Esto no solo previene el desperdicio de agua, sino que también promueve una irrigación más eficiente y precisa, adaptada específicamente a las necesidades hídricas de cada tipo de planta.

Cuidado Personalizado de las Plantas

Cada planta en el Jardín Inteligente recibe un cuidado verdaderamente personalizado gracias a la capacidad del sistema para monitorear múltiples variables ambientales de forma simultánea. Además de la humedad del suelo, hay un sensor de lluvia, y datos actualizados de la temperatura ambiental y demás, que ayudan con el crecimiento y la salud de las plantas. Con esta información detallada, el sistema ajusta automáticamente los parámetros de riego y

otras necesidades específicas de cuidado para cada planta, asegurando condiciones ideales para su desarrollo óptimo y saludable.

Monitoreo y Control Remoto Avanzado

Los usuarios podrán disfrutar de un control sin precedentes sobre su entorno verde gracias a una aplicación intuitiva y funcional. Este aplicativo proporciona datos en tiempo real sobre las condiciones del jardín y pueden alertar a los usuarios sobre cambios significativos en las condiciones del jardín o la necesidad de intervención manual, asegurando una supervisión constante y efectiva desde cualquier lugar.

Ahorro de Tiempo y Simplificación del Mantenimiento

La automatización completa de tareas rutinarias como el riego y el monitoreo ambiental no solo libera tiempo para los usuarios, sino que también elimina la carga de preocupaciones constantes sobre el estado de sus plantas. Al reducir drásticamente la intervención manual requerida para mantener un jardín saludable, el sistema permite a los usuarios disfrutar plenamente de su espacio verde sin comprometer la calidad del cuidado ofrecido. Esta eficiencia operativa no solo es conveniente, sino que también fomenta un estilo de vida más equilibrado y sostenible para los entusiastas de la jardinería y los profesionales del paisajismo por igual.

CONCLUSIONES

El Jardín Inteligente basado en IoT representa una innovación significativa en el cuidado de las plantas, ofreciendo una solución eficiente y tecnológicamente avanzada. Al aprovechar la tecnología IoT, este sistema no solo simplifica el monitoreo y control del jardín, sino que también promueve prácticas sostenibles y optimiza el uso de recursos naturales. Los resultados obtenidos hasta ahora reflejan el potencial transformador de este enfoque, elevando la jardinería tradicional a nuevos estándares de precisión y automatización.

FUTURAS MEJORAS Y RECOMENDACIONES

Para continuar mejorando el Jardín Inteligente y maximizar su eficacia, se recomienda considerar las siguientes áreas de desarrollo:

Integración con Asistentes de Voz

La integración con plataformas de asistentes de voz como Amazon Alexa o Google Assistant permitiría a los usuarios controlar el sistema de forma intuitiva y manos libres. Esta mejora

no solo aumentaría la accesibilidad del sistema, sino que también mejoraría la experiencia del usuario al ofrecer opciones de control adicionales y prácticas.

Análisis Predictivo y Machine Learning

Implementar capacidades de análisis predictivo utilizando técnicas de machine learning permitiría al sistema anticipar las necesidades futuras de las plantas. Mediante el análisis de datos históricos y las condiciones ambientales actuales, el sistema podría recomendar acciones preventivas como ajustes anticipados en el riego o cambios en la iluminación, mejorando así la salud y el crecimiento de las plantas de manera proactiva.

Mejora Continua de la Interfaz de Usuario

Continuar refinando la interfaz de usuario de la aplicación para hacerla aún más intuitiva y fácil de usar sería beneficioso. Esto incluiría la incorporación de notificaciones Personalizadas y guías paso a paso para la configuración inicial y el mantenimiento continuo del sistema.

Implementación de Funcionalidades Ambientales Adicionales

Explorar la integración de sensores adicionales para medir parámetros ambientales como la calidad del aire, la temperatura ambiente y la humedad relativa podría proporcionar una visión más completa del entorno del jardín. Estos datos adicionales no solo mejorarían la precisión del sistema en la toma de decisiones, sino que también ofrecerían información valiosa sobre el bienestar general del entorno vegetal.