**Imagen que contiene Logotipo

Descripción generada automáticamenteLogotipo, nombre de la empresa

Descripción generada automáticamente**

**Anteproyecto**

**Microprocesadores**

**Sistema de Riego**

**Equipo # 4**

**Nombre:** Mario Alejandro Bernal Cárdenas **Matrícula:** 1995007

**Nombre:** Jahir Nicolás López Hernández **Matrícula**: 1967014

**Nombre:** Arturo Alejandro Vázquez Cuesta **Matrícula:** 1959737

**Nombre:** Erick Soriano Santillán **Matrícula:** 2078566

**Nombre:** Melissa Olivares García **Matrícula:** 2023215

**Nombre:** Daniel Treviño Almaguer  **Matricula:** 1845896

**Maestro:** Luis Gerardo Garza Garza

**Materia:** Microprocesadores **Grupo**: 035 **Aula:** 115

**Carrera:** Licenciatura en Ciencias Computacionales

**Anteproyecto. - Sistema de riego**

Este proyecto se centra en el desarrollo de un sistema de riego automatizado que incorporará un sensor de temperatura y uno de aire. Cuando la temperatura y la humedad alcancen ciertos niveles predefinidos, al igual que la calidad del aire, el sistema activará el riego para proporcionar agua a las plantas. El enfoque principal es brindar una solución práctica y cotidiana para el cuidado de plantas en entornos domésticos u otros contextos similares. También para tener un monitoreo, se creará una aplicación web (hecha en Vue.js) que pueda reportar dichos niveles de temperatura y humedad, y a su vez, programar horarios definidos para que se active automáticamente en ciertos casos.

**Objetivos principales**

* Implementar un sistema que utilice un sensor de temperatura y humedad para monitorear las condiciones del entorno.
* Desarrollar una aplicación web que permita a los usuarios monitorear los niveles de temperatura y humedad en tiempo real.
* Configurar la conectividad inalámbrica para permitir la comunicación entre el sistema de riego y la aplicación web.
* Implementar la lógica de riego que asegure que el sistema responda de manera adecuada a los cambios en los niveles de temperatura y humedad.
* Diseñar la interfaz de usuario de la aplicación móvil de manera intuitiva y fácil de usar.
* Considerar prácticas sostenibles, como la reutilización del agua o la implementación de horarios de riego que minimicen el desperdicio.

**Alcances principales**

* Desarrollar y construir un sistema de riego automatizado que integre el sensor de temperatura y humedad, así como los componentes necesarios para el riego.
* Programar la lógica del sistema para que interprete los datos del sensor y active el riego de acuerdo con los niveles predefinidos de temperatura y humedad.
* Crear una aplicación web que se conecte al sistema de riego y permita a los usuarios monitorear los niveles de temperatura y humedad, así como programar horarios para la activación automática del riego.
* Configurar la conectividad inalámbrica entre el sistema de riego y la aplicación móvil, utilizando, por ejemplo, módulos como el ESP8266.
* Definir claramente los umbrales de temperatura y humedad que activarán el sistema de riego, garantizando una respuesta efectiva a las condiciones del entorno.
* Realizar pruebas exhaustivas del sistema para asegurar su funcionamiento correcto y realizar ajustes según sea necesario.
* Implementar el sistema de riego automatizado en un entorno real, ya sea en un entorno doméstico o en otro contexto específico.

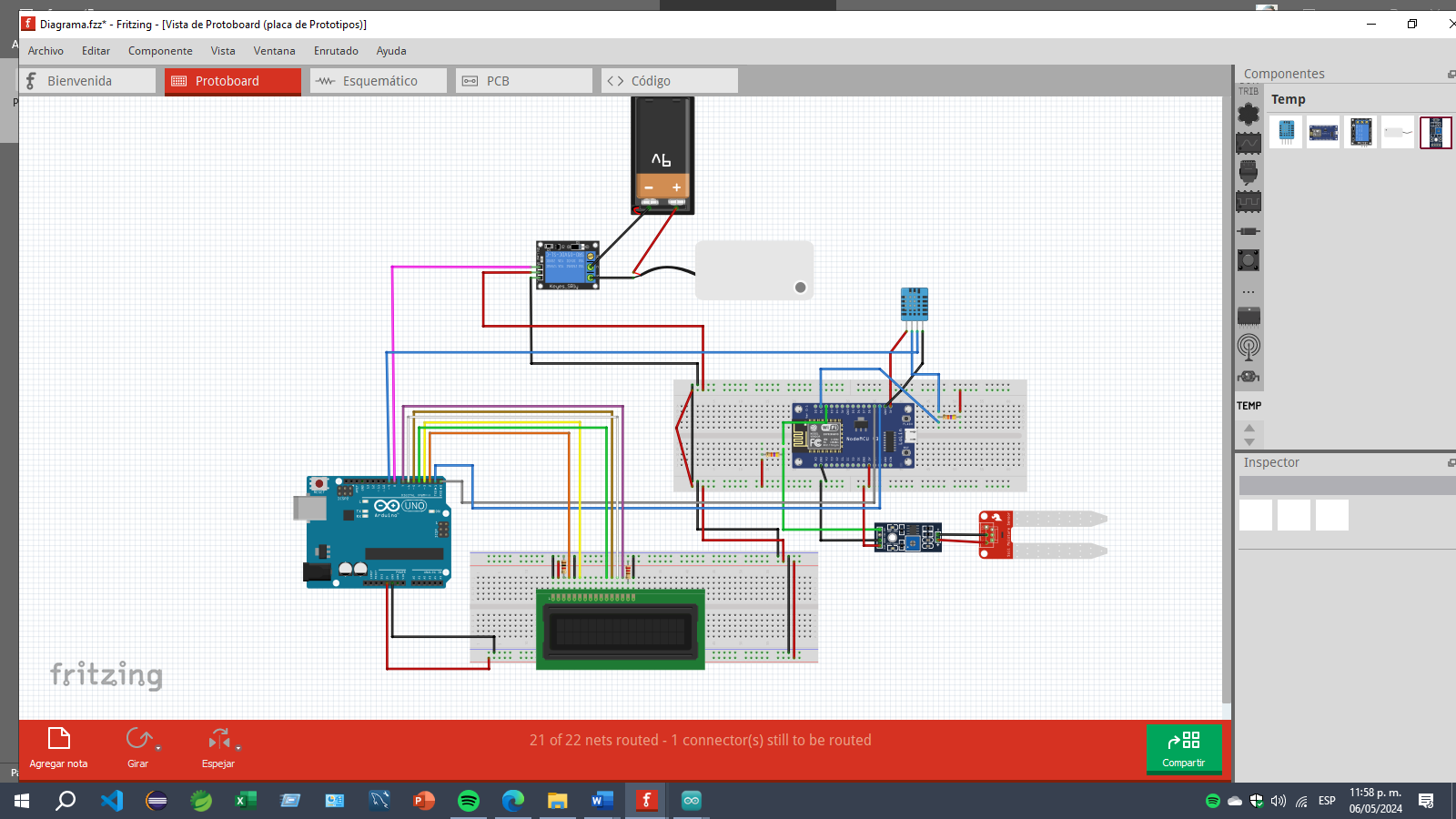
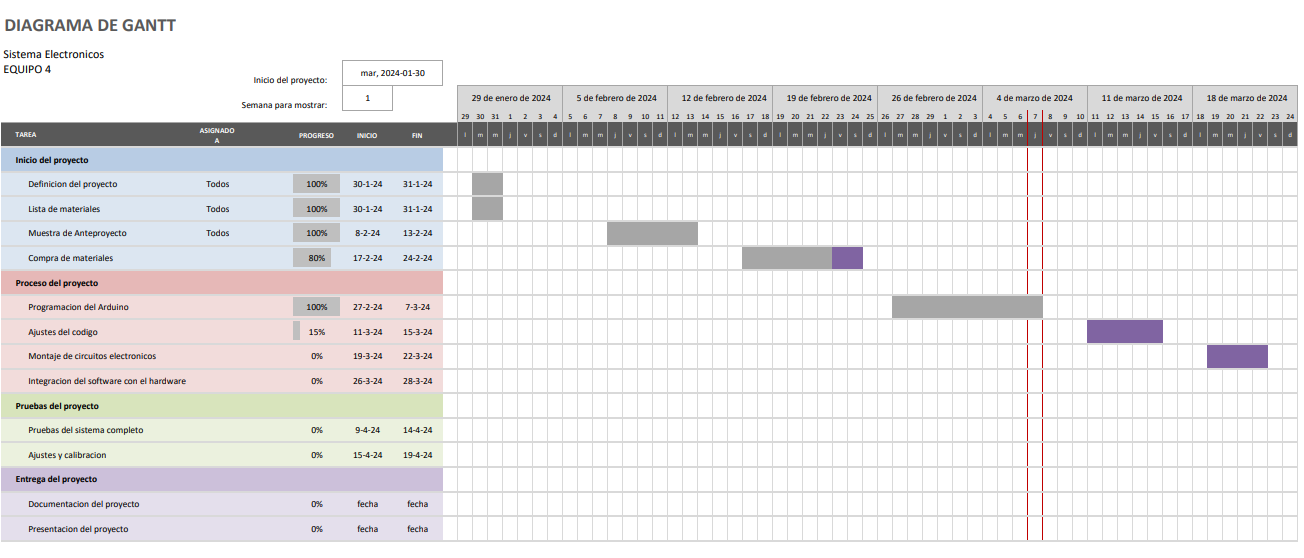
**Beneficios**

* El sistema automatizado permite un riego más preciso y eficiente. Al activarse solo cuando sea necesario según las condiciones climáticas, se reduce el desperdicio de agua.
* El sistema puede proporcionar un cuidado más preciso y específico para las plantas.
* La automatización del riego basada en condiciones específicas libera a los usuarios de la tarea manual de controlar constantemente el riego. Esto ahorra tiempo y esfuerzo.
* Desarrollar y construir este tipo de proyectos fomenta la innovación y la aplicación práctica de la tecnología en entornos domésticos, promoviendo un enfoque más tecnológico y sostenible para el cuidado de las plantas.

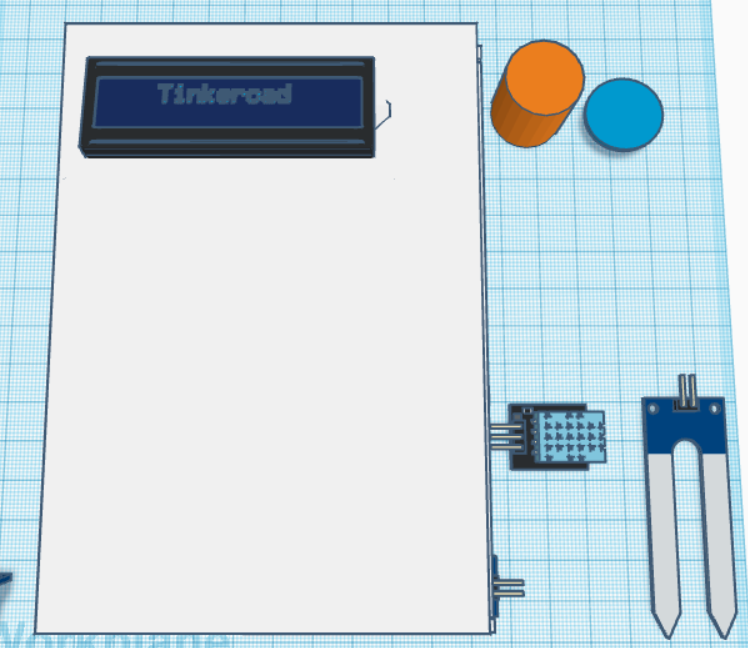
**Materiales**

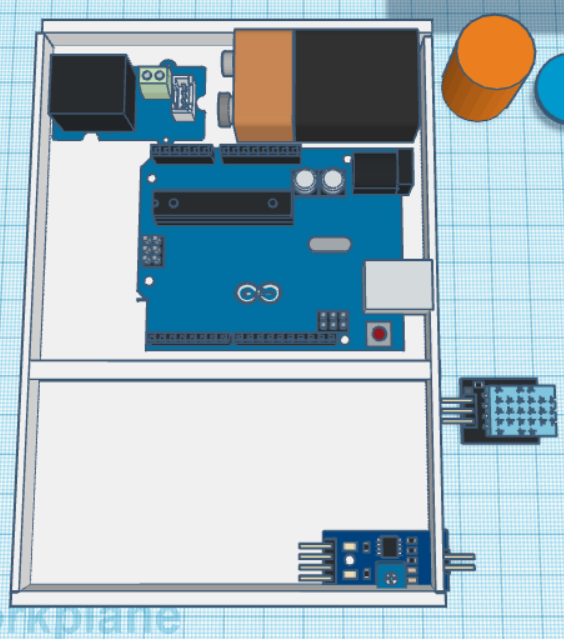
|  |  |  |  |
| --- | --- | --- | --- |
| **Componente** | **Proveedor** | **Precio** | **Tiempo de entrega** |
| **Sensor DHT11 3V - 5V DC** | Electrónica para estudiantes | $40.00 | 01/03/24 |
| **Pantalla LCD 5V DC (16x2)** | Electrónica para estudiantes | $60.00 | 01/03/24 |
| **Nodemcu (ESP8266)** | Electrónica para estudiantes | $145.00 | 08/03/24 |
| **Bomba de agua 2,5V-5V DC** | Electrónica para estudiantes | $60.00 | 08/03/24 |
| **Relé 5V DC**  **K542** | Electrónica para estudiantes | $30.00 | 08/03/24 |
| **Protoboard** | Electrónica para estudiantes | $45.00 | 01/03/24 |
| **Resistencia 1KΩ** | Electrónica para estudiantes | $5.00 | 01/03/24 |
| **Resistencia 220Ω** | Electrónica para estudiantes | $5.00 | 01/03/24 |
| **Resistencia 4.7 KΩ** | Electrónica para estudiantes | $5.00 | 01/03/24 |
| **Arduino UNO** | Electrónica para estudiantes | $158.00 | 01/03/24 |
| **Sensor de Tierra**  **C102** | Electrónica para estudiantes | $27.00 | 08/03/24 |
| TOTAL | | $580.00 |  |

**Esquema**

**Diagrama de Gantt**

**Bosquejo de la maqueta**





**Código fuente**

Para Nodemcu:

#include "DHT.h"

#include <SoftwareSerial.h>

#include <ESP8266WiFi.h>

#include <Arduino.h>

#include <ArduinoJson.h>

#include <Firebase\_ESP\_Client.h>

#include "addons/TokenHelper.h"

#include "addons/RTDBHelper.h"

#define DHTPIN 4

#define DHTTYPE DHT11 // DHT 11

#define WIFI\_SSID "your\_user"

#define WIFI\_PASSWORD "your\_password"

#define API\_KEY "AIzaSyBhGWjM2IEoMMO0l\_OujUsBJPHfjKqCH90"

#define DATABASE\_URL "https://sistema-de-riego-24fc2-default-rtdb.firebaseio.com/"

#define FIREBASE\_PROJECT\_ID "sistema-de-riego-24fc2"

  FirebaseData fbdo;

  FirebaseAuth auth;

  FirebaseConfig config;

  unsigned long sendDataPrevMillis = 0;

  int count = 0;

  bool signupOK = false;

  const int sensorPin = A0;

  DHT dht(DHTPIN, DHTTYPE);

  float t;

  float h;

  String ruta = "ambientquality";

  void setup(){

    Serial.begin(115200);

    Serial1.begin(115200);

    Serial.println(F("Dtxx Text!!!"));

    dht.begin();

     if (isnan(h) || isnan(t)){

      Serial.println(F("Failed to read from DHT sensor!"));

      return;

    }

    WiFi.begin(WIFI\_SSID, WIFI\_PASSWORD);

    Serial.print("Connecting to Wi-Fi");

    while (WiFi.status() != WL\_CONNECTED){

      Serial.print(".");

      delay(300);

    }

    Serial.println();

    Serial.print("Connected with IP: ");

    Serial.println(WiFi.localIP());

    Serial.println();

    config.api\_key = API\_KEY;

    config.database\_url = DATABASE\_URL;

    if (Firebase.signUp(&config, &auth, "", "")){

      Serial.println("ok");

      signupOK = true;

    }else{

      Serial.printf("%s\n", config.signer.signupError.message.c\_str());

    }

    config.token\_status\_callback = tokenStatusCallback;

    Firebase.begin(&config, &auth);

    Firebase.reconnectWiFi(true);

  }

  void loop(){

    String documentPath = "EspData/ambientquality";

    String documentPathT = "EspData/intervals";

    FirebaseJson content;

    h = dht.readHumidity();

    t = dht.readTemperature();

    String estado = "off";

    int sensorHumedad = analogRead(sensorPin);

    float valorHumedad = ( 100.00 - ( (sensorHumedad/1023.00) \* 100.00 ) );

    if(valorHumedad<0){

      valorHumedad = 0;

    }

    if(t>=25 && h <=20){

      estado = "on";

    }

    if(valorHumedad<20){

      estado = "on";

    }

    if (Serial.available() > 0) {

      String dataReceived = Serial.readStringUntil('\n');

      if(dataReceived.length() == 6){

        content.set("fields/waterPlants/booleanValue", false);

        Firebase.Firestore.patchDocument(&fbdo, FIREBASE\_PROJECT\_ID, "", documentPathT.c\_str(), content.raw(), "waterPlants");

      }

    }

    if (Firebase.ready() && signupOK && (millis() - sendDataPrevMillis > 1000 || sendDataPrevMillis == 0)){

      sendDataPrevMillis = millis();

    }

    if(!isnan(t) && !isnan(h)){

      content.set("fields/Temperature/stringValue", String(t, 2));

      content.set("fields/Humidity/stringValue", String(h,2));

      Firebase.Firestore.patchDocument(&fbdo, FIREBASE\_PROJECT\_ID, "", documentPath.c\_str(), content.raw(), "Temperature");

      Firebase.Firestore.patchDocument(&fbdo, FIREBASE\_PROJECT\_ID, "", documentPath.c\_str(), content.raw(), "Humidity");

    }else{

      Serial.println("Failed to read DHT data");

    }

    if(!isnan(sensorHumedad)){

      content.set("fields/SoilMoisture/stringValue", String(valorHumedad, 2));

      Firebase.Firestore.patchDocument(&fbdo, FIREBASE\_PROJECT\_ID, "", documentPath.c\_str(), content.raw(), "SoilMoisture");

    }else{

      Serial.println("Failed to read Soil Moisture Sensor data");

    }

    String path = "EspData/intervals";

    if(Firebase.Firestore.getDocument(&fbdo, FIREBASE\_PROJECT\_ID, "", path.c\_str(), "")){

      StaticJsonDocument<384> doc;

      DeserializationError error = deserializeJson(doc, fbdo.payload().c\_str());

      if (!error) {

          const char\* name = doc["name"];

          const int temp = doc["fields"]["interval"]["integerValue"];

          const bool interruptor = doc["fields"]["waterPlants"]["booleanValue"];

          mostrarInfo((int) t, h, valorHumedad, estado, temp, interruptor);

      }

    }

    if (!Firebase.RTDB.setFloat(&fbdo, ruta + "/Temperature", (int) t)){

        Serial.println("Failed to Read from the Sensor");

        Serial.println("REASON: " + fbdo.errorReason());

    }

    if (!Firebase.RTDB.setFloat(&fbdo, ruta + "/Humidity", h)){

        Serial.println("Failed to Read from the Sensor");

        Serial.println("REASON: " + fbdo.errorReason());

    }

    if (!Firebase.RTDB.setFloat(&fbdo, ruta + "/SoilMoisture", valorHumedad)){

        Serial.println("Failed to Read from the Sensor");

        Serial.println("REASON: " + fbdo.errorReason());

    }

    delay(3000);

  }

  void mostrarInfo(int temp, float hum, float valorHumedad, String estado, int t, bool inter){

      String strdht = String("H:")+String(hum)+String("%")+String(" T:")+String(temp)+String(" C");

      String strvH = String(" SoilMoisture:")+String(valorHumedad)+String("%") ;

      String strEst = String(" State:")+estado;

      String strtemp = String(" Temp:") + String(t);

      String strinter = String(" B:") + String(inter);

      String arduino = String(strdht)+String(strvH)+strEst+String(strtemp)+String(strinter);

      Serial.println(arduino);

  }

Para Arduino:

#include <LiquidCrystal.h>

LiquidCrystal lcd(2,3,4,5,6,7);

int bombaAgua = 8;

void setup() {

  lcd.begin(16,2);

  Serial.begin(115200);

  while (!Serial) {

    ;

    }

  lcd.print("    Equipo 4");

  pinMode(bombaAgua, OUTPUT);

}

  bool turno = true;

void loop() {

  if (Serial.available() > 0) {

    String dataReceived = Serial.readStringUntil('\n');

    dataReceived.trim();

    int hIndex = dataReceived.indexOf("H:");

    int tIndex = dataReceived.indexOf(" T:");

    int vHIndex = dataReceived.indexOf(" SoilMoisture:");

    int estIndex = dataReceived.indexOf(" State:");

    int tempIndex = dataReceived.indexOf(" Temp:");

    int inteIndex = dataReceived.indexOf(" B:");

    String humedad = dataReceived.substring(hIndex + 2, tIndex);

    String temperatura = dataReceived.substring(tIndex + 3, vHIndex);

    String sm = dataReceived.substring(vHIndex + 14, estIndex);

    String est = dataReceived.substring(estIndex + 7, tempIndex);

    String temp = dataReceived.substring(tempIndex + 6, inteIndex);

    String inter = dataReceived.substring(inteIndex+3);

    lcd.clear();

    lcd.setCursor(0, 0);

    if (turno) {

      lcd.print("   Hum: ");

      lcd.print(humedad);

      lcd.setCursor(0, 1);

      lcd.print("   Temp: ");

      lcd.print(temperatura);

    } else {

      lcd.print(" Soil Moisture:");

      lcd.setCursor(0, 1);

      lcd.print("    " + sm);

    }

    turno = !turno;

    if(inter == "1"){

      if(temp.toInt() != 0){

        lcd.clear();

        lcd.print("   Regando... ");

        digitalWrite(bombaAgua, LOW);

        delay(temp.toInt());

        digitalWrite(bombaAgua, HIGH);

      }

    }

    digitalWrite(bombaAgua, HIGH);

    if(est.length() == 2){

      lcd.clear();

      lcd.print("   Regando... ");

      digitalWrite(bombaAgua, LOW);

      delay(5000);

    }else{

      digitalWrite(bombaAgua, HIGH);

    }

    String off = "false";

    Serial.println(off);

  }

  delay(1000);

}

**Software del proyecto**

