**INSTITUCIÓN UNIVERSITARIA CESMAG**

**FACULTAD DE INGENIERÍA - INGENIERÍA ELECTRÓNICA**

**ELECTIVA DE PROGRAMACIÓN 2A**

**PRESENTACIÓN PROYECTO:**

INVERNADERO AUTOMATIZADO

**INTEGRANTES:**

Yesid Nasner

Sebastián Andrade

Daniel Obando

|  |
| --- |
| **Nombre del proyecto:** |
| Resultado de imagen para logos de invernaderos  **INNOVA invernaderos** |
| **Objetivo general:** |
| Diseñar y Desarrollar un sistema que nos permita realizar control de un microclima adecuado para la producción de cultivos de frutas, flores y hortalizas. |
| **Objetivos específicos:** |
| -ventajas del sistema del invernadero automatizados es la mayor productividad por m2.  -tener una producción de calidad, el control eficiente de plagas y enfermedades del cultivo.  - un mayor control de los factores ambientales, para poder producir fuera de época.  -tener más oportunidad de comercializar cultivos de alta calidad en un mercado competitivo. |
| **1. FASE DE ANÁLISIS** |
| 1**.** puerta automática- se abrirá con tan solo tocar un botón en la interfaz de python.  2. sistema de riego automatizado – cuando sea necesario hidratar el cultivo se encenderá una bomba de agua.  3. sistema de enfriamiento inteligente – cuando la temperatura exceda los 30°c debe activarse el ventilador.  4.sensores de temperatura y humedad para tener una medida exacta del clima en el interior de invernadero |
| **2. FASE DE DISEÑO – MAQUETACIÓN Y PROTOTIPADO**    **-Puerta automática**    **Sistema de ventilación**      Sistema de iluminación  C:\Users\joseandres\AppData\Local\Microsoft\Windows\INetCache\Content.Word\IMG_20181205_135922.jpg  Sistema de riego  C:\Users\joseandres\AppData\Local\Microsoft\Windows\INetCache\Content.Word\riego2.jpg |
| **2.1. Maqueta o prototipo del invernadero:** |
|  |
|  |
| **2.2. Mockups Desktop App:** |
|  |
| **2.2. Arquitectura del sistema de control: \*** Debe ser representada de manera gráfica y lógica |
|  |
| **2.4. Componentes electrónicos:**   |  |  |  | | --- | --- | --- | | **Elemento/componente** | **Referencia/característica/Descripción** | **Función** | | **Placa arduino Uno** | Es una placa electrónica basada en el micro controlador ATmega3, tiene 2814 entradas y salidas digitales las cuales 6 pueden ser utilizadas como salidas PWM | Una de sus grandes ventajas, es que tiene infinidad de aplicaciones… pero en este caso es utilizada para el control de un invernadero automatizado | | **protoboard** | Es un tablero con orificios en el cual se puede insertar componentes electrónicos y cables para armar los circuitos | sirve para armar el circuito necesario para el funcionamiento adecuado para el invernadero | | **motor DC** | Es una máquina que convierte energía eléctrica en mecánica, provocando un movimiento rotatorio, gracias a la acción de un campo magnético. | Generar un movimiento rotatorio para una bomba de agua pala adecuada hidratación del cultivo |  |  |  |  | | --- | --- | --- | | **ventilador** | enfriar la máquina y para proteger contra el calor, refrigeración de la CPU | Darle la temperatura adecuada al clima en el interior del invernadero | | **Relé** | Funciona como un interruptor controlado por un circuito eléctrico en el que, por medio de una bobina y un electroimán, se acciona un juego de uno o varios contactos que permiten abrir o cerrar otros circuitos eléctricos independientes. | El funcionamiento en el circuito es encender el motor dc y el ventilador | | **Diodo 1N4004** | Un **diodo** es un componente electrónico de dos terminales que permite la circulación de la corriente eléctrica a través de él en un solo sentido.   El 1N4004 es un rectificador de uso general con baja caída de tensión directa y alta capacidad de corriente de sobretensión. | Hacer una buena circulación de corriente eléctrica en el circuito | | **Resistencia eléctrica** | Resistencia eléctrica es toda oposición que encuentra la corriente a su paso por un circuito eléctrico cerrado, atenuando o frenando el libre flujo de circulación de las cargas eléctricas o electrones. | Reducir el flujo de corriente en algunos dispositivos del circuito armado en la protoboard | |
| **3. FASE DE PROGRAMACIÓN - CÓDIGOS FUENTE**  **\*** El código fuente debe estar debidamente comentado,  explicando la función de las líneas principales |
| **3.1. Código fuente Arduino (.ino):** |
| **const int pinLED = 13;**  **char serialData;**  **int pinLed = 12;**  **const int sensor = A1; //Pin en el que está conectado el sensor**  **const int Rc = 1000; //Resistencia de calibración**  **const int relay = 2; //Pin en en que está conectado el Relay**  **int V;**  **int contador = 0;**  **long Rsensor;**  **long Resk;**  **unsigned long tr = 1; //Tiempo de regado en minutos //AJUSTAR**  **unsigned long tespera = 10; //Tiempo espera en minutos //AJUSTAR**  **const int rr = 0.5 ; //Resistencia (en kohmios) a partir de la cual empieza a regar //AJUSTAR**  **int dato1;**  **int dato2;**  **float voltios;**  **float temperaturaC;**  **int alarma=7;**  **int techo=8;**  **int ventilador=9;**  **void ControlAlarma()**  **{**  **if(temperaturaC<24 && dato2>850)**  **{**  **digitalWrite(alarma,LOW);**  **}**    **if(temperaturaC>=24)**  **{**  **digitalWrite(alarma,HIGH);**  **}**  **}**  **void ControlTecho()**  **{**  **if(temperaturaC>=24)**  **{**  **digitalWrite(techo,HIGH);**  **}**  **else**  **{**  **digitalWrite(techo,LOW);**  **}**  **}**  **void ControlVentilador()**  **{**  **if(temperaturaC>=24)**  **{**  **digitalWrite(ventilador,HIGH);**  **}**  **else**  **{**  **digitalWrite(ventilador,LOW);**  **}**  **}**  **void setup() {**    **// put your setup code here, to run once:**  **pinMode(pinLed,OUTPUT);**  **Serial.begin(9600);**    **//puerta**  **Serial.begin(9600);**  **pinMode(pinLED, OUTPUT);**  **pinMode(10,OUTPUT);**  **pinMode(6,INPUT);**  **pinMode(5,INPUT);**    **Serial.begin(9600); //inicia comunicación serial**  **pinMode (relay,OUTPUT); //Configurar relay como salida**  **pinMode(alarma,OUTPUT);**  **pinMode(techo,OUTPUT);**  **pinMode(ventilador,OUTPUT);**  **}**  **void loop() {**  **// put your main code here, to run repeatedly:**    **V = analogRead(sensor); //leer sensor**  **Rsensor = 1024L \* Rc / V - Rc; //calcular resistencia del sensor**  **Resk = Rsensor /1000; //pasar a kiloohmios**  **Serial.print(contador); Serial.println(" riegos."); //contador para ver en serial cuantas veces se ha regado**  **Serial.print("Valor resistencia: "); Serial.print(Resk); Serial.println(" mil ohmios "); //escribir en serial el valor de la resistencia**  **delay(1000);**  **if (Resk>rr){**  **digitalWrite(relay,HIGH); //activar relay**  **delay(tr\*60\*1000); // Espera Tiempo de regado con relay activado**  **digitalWrite(relay,LOW); //desactivar relay**  **delay(tespera\*60\*1000); // Espera Tiempo de espera entre regados con relay desactivado para que el agua se filtre a la tierra y llegue al sensor**  **contador = contador + 1 ;**  **}**    **//puerta**  **// luz**  **if(Serial.available()>0){**  **serialData = Serial.read();**  **Serial.print(serialData);**    **if(serialData== '1'){**  **digitalWrite(pinLed, HIGH);**  **} else{**  **if(serialData== '0'){**  **digitalWrite(pinLed, LOW);**  **}**  **}**  **}**  **if (Serial.available()>0)**  **{**  **char option = Serial.read();**  **if (option >= '1' && option <= '9')**  **{**  **option -= '0';**  **for (int i = 0;i<option;i++)**  **{**  **digitalWrite(pinLED, HIGH);**  **delay(200);**    **digitalWrite(pinLED, LOW);**      **}**  **}**  **}**      **dato1 = analogRead(A0);**  **voltios = (dato1 / 1023.0) \* 5;**  **temperaturaC = voltios / 10e-3;**  **Serial.print("TEMPERATURA : ");**  **Serial.print(temperaturaC);**  **Serial.println(" C");**  **delay(300);**    **Serial.println("");**    **dato2 = analogRead(A1);**  **// Serial.print("HUMEDAD : ");**  **// Serial.println(dato2);**  **delay(1000);**  **int lecturaporcentaje = map(dato2, 1023, 0, 0, 100);**  **Serial.print("la humedad es del: ");**  **Serial.print(lecturaporcentaje);**  **Serial.println("%");**  **ControlAlarma();**  **ControlVentilador();**  **ControlTecho();**    **}** |
| **3.2. Código fuente Python (.py):** |
| **import serial**  **from tkinter import \***  **import tkinter**  **arduinoData = serial.Serial('COM3',9600)**  **#led 1**  **def led\_on():**  **arduinoData.write(b'1')**  **print("encender luces")**  **def led\_off():**  **arduinoData.write(b'0')**  **print("apagar luces")**  **#ventana 1**  **led\_control\_window = Tk()**  **btn1 = Button (led\_control\_window, text="ON", command=led\_on)**  **btn2 = Button (led\_control\_window, text="OFF", command=led\_off)**  **btn1.grid(row=0,column=1)**  **btn2.grid(row=1,column=1)**  **led\_control\_window.mainloop()** |
| **4. GUI (INTERFAZ GRÁFICA DE USUARIO)**  **\*** Se deben relacionar las interfaces gráficas de usuario que ya han sido desarrolladas.  Tenga en cuenta que cada interfaz debe coincidir con los Mockups creados en la fase de diseño |
|  |
| **OBSERVACIONES GENERALES:** |
| En la fase de análisis debe relacionar en detalle las funciones y operaciones que serán controladas de manera automática y a través de interfaz gráfica:   * **Luces:**   + Sensores de luminosidad * **Ventanas y puertas:**   + Sensores y seguridad   + Uso de servomotores   + Sensores de proximidad * **Ventiladores:**   + Climatización   + Sensores de temperatura * Aportes libres de nuevo conocimiento y/o autoprendizaje del estudiante. |

****