

# Proyecto Final



## Pensamiento Computacional y Programación



Colegio La Girouette

---

Profesor:	Darío Creado
Experiencia :	Proyecto Final del Curso - <b>Invernadero Inteligente</b>
Fecha de Trabajo:	22 de Noviembre 2022

---

## 1 Descripción

El presente trabajo aborda la totalidad de los contenidos revisados durante la segunda mitad del año mediante el diseño y la construcción de un invernadero automatizado ó inteligente. Durante los últimas décadas hemos visto como la electrónica digital ha formado parte de nuestra vida diaria a niveles que gran parte de la población desconoce, prácticamente cualquier artefacto electrónico disponible en nuestros hogares, lugares de estudio o trabajo están dotados de cierto nivel de autonomía o incluso inteligencia. Durante esta experiencia se busca que diseñen utilizando las herramientas discutidas en clases (diseño lógico combinacional, secuencial, máquinas de estado finitas, etc) un sistema de control para un invernadero inteligente. Se presentará en el enunciado una serie de exigencias para poder cumplir con los objetivos propuestos.

## 2 Restricciones

- El microcontrolador a utilizar en el proyecto corresponderá al **ATMEGA328P** incluido en la tarjeta de desarrollo **ARDUINO UNO R3** disponible en el laboratorio.
- La alimentación del sistema deberá poseer cierto grado de autonomía, es decir, en la versión final del proyecto el microcontrolador no deberá estar alimentado por un ordenador o alguna fuente de voltaje de laboratorio. Se recomienda el uso de *power bank*.
- No está permitido el uso de tensiones mayores a 12V en régimen continuo, misma restricción para niveles de corriente dados por este voltaje. Cualquier problema de diseño que implique el no cumplimiento de esta restricción deberá ser comunicada al profesor antes de realizar cualquier montaje.
- Se pide el mayor orden en los códigos de programación. Definir variables globales, el uso de funciones, librerías son valorables para poder tener una mejor interpretación del trabajo realizado por parte de quien revisa. **Cualquier código extraído desde alguna fuente deberá ser citado correctamente.**
- La temperatura de referencia para el sistema de control deberá ser de 25 grados celsius con una tolerancia del 1% al 3%.
- En una pantalla LCD u OLED anclada a la estructura del invernadero deberán mostrarse los parámetros relevantes para el control del invernadero. Los parámetros obligatorios son temperatura y humedad.

- g) Se pide que el circuito general del proyecto esté dividido en bloques funcionales. Por otra parte, se deberán realizar entregas parciales de estos bloques simulados en la plataforma <https://www.tinkercad.com/>.
- h) El proyecto deberá contar con un control manual de temperatura, este control se puede implementar para la refrigeración ó la calefacción. Si la temperatura al interior del invernadero es menor

**OBSERVACIÓN : SE PUBLICARÁ UN SIGUIENTE DOCUMENTO CON DETALLE TÉCNICO Y RÚBRICA DE EVALUACIÓN.**

### 3 Diseño Preliminar - Sugerencia

Con el objetivo de acelerar el proceso de diseño, se presenta el siguiente *layout* para el sistema de control del invernadero. Como primer *bloque* (S1) se tiene el control de temperatura y luminosidad para el invernadero. En este bloque destaca la presencia del integrado **Puente H**, fotoresistor, un led simbolizando la fuente lumínica como luz aartificial, sensor de temperatura **TMP36**, display LCD, enfriador y calentador representados por un Motor DC Serie (se pueden reemplazar por ventiladores de computador estándar), etc.

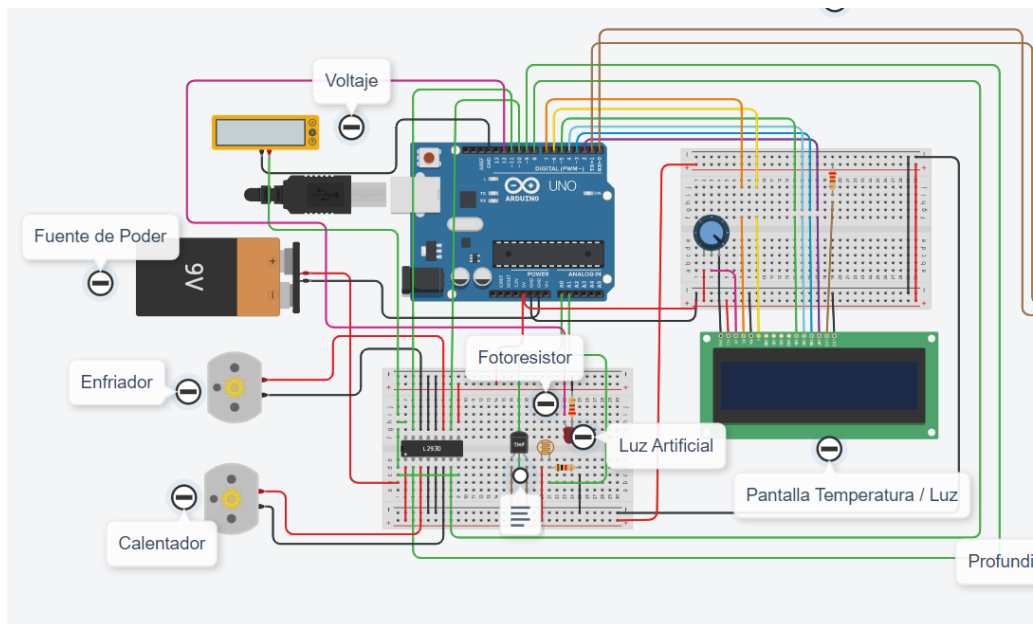


Figure 1: Sección 1 (S1) - Control

Se considera también un segundo bloque (S2) a cargo del control de nivel de agua en un estanque de reserva, acceso por medio de una *matriz de digitación*, medición de humo, control de la motobomba para el riego, etc.

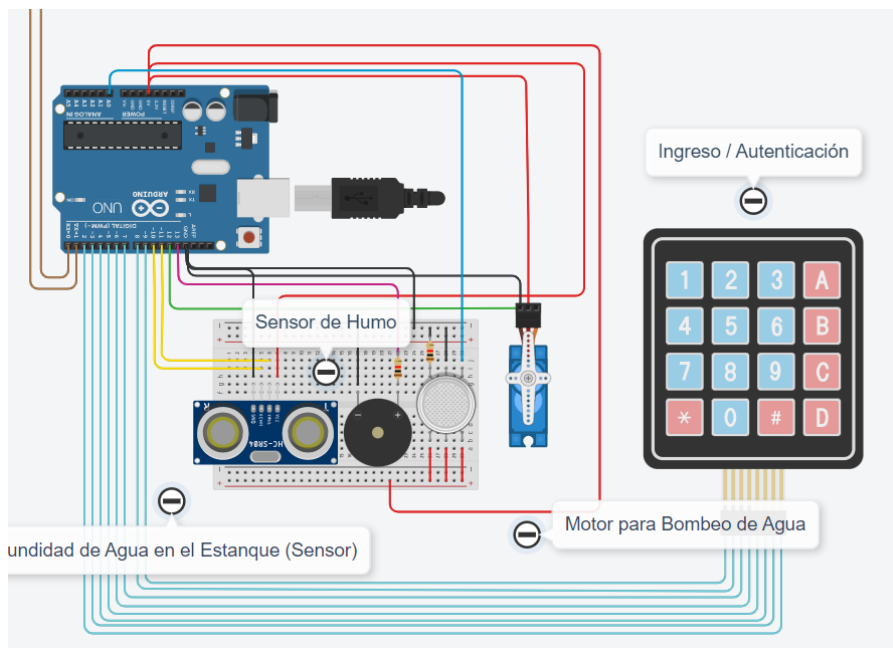


Figure 2: Sección 2 (S2) - Control

los códigos principales para S1 y S2 se presentan a continuación :

---

```
1 // Monitoreo Invernadero [S1]. Source : TinkerCad Community
2 #include <LiquidCrystal.h>
3
4 // Conexión Pantalla LCD
5 int regSel = 7;
6 int enable = 6;
7 int sig1 = 5;
8 int sig2 = 4;
9 int sig3 = 3;
10 int sig4 = 2;
11
12 // Enfriador y Calefactor
13 int motor1 = 8;
14 int motor2 = 9;
15 int motor3 = 10;
16 int motor4 = 11;
17
18 // Control de luz artificial
19 int light = 12; // led
20
21 // Sensores
22 char temp = A0; // Sensor Temperatura
23 char lumos = A1; // Fotorresistor
24
25 // Variables Auxiliares
26 float temperature;
27 float intensity;
28 int flag = 0;
29
30 // Inicialización en LCD
31 LiquidCrystal lcd(regSel, enable, sig1, sig2, sig3, sig4);
32
33 // Declaración de Funciones
34 float readTemperature(char); // Temperatura
35 float readLuminosity(char); // Intensidad de Luz
36 void authenticate();
37
38 void setup() {
39     // Pins
40     pinMode(motor1, OUTPUT);
41     pinMode(motor2, OUTPUT);
42     pinMode(motor3, OUTPUT);
43     pinMode(motor4, OUTPUT);
44     pinMode(light, OUTPUT);
45
46     // Pins Lectura
47     pinMode(temp, INPUT);
48     pinMode(lumos, INPUT);
49
50     // Serial monitor
```

```

51     Serial.begin(1156200); // baud rate de 1156200 OJO No 9600
52     // LCD
53     Serial.flush();
54     lcd.begin(16, 2); // 16*2 Dimensiones de LCD
55 }
56
57 void loop() {
58     lcd.clear();
59     temperature = readTemperature(temp);
60     lcd.setCursor(3, 0); // Posición LCD
61     lcd.print("Temp: ");
62     lcd.setCursor(9, 0);
63     lcd.print(temperature);
64     if (temperature < 20) {
65         lcd.setCursor(4, 1);
66         lcd.print("BAJA TEMPERATURA");
67         digitalWrite(8, HIGH); // Habilita calefactor
68         digitalWrite(10, LOW); // No enfriador
69     } else if (temperature > 27) { // Temperatura de Control
70         lcd.setCursor(4, 1);
71         lcd.print("ALTA TEMPERATURA");
72         digitalWrite(8, LOW); // No calefactor
73         digitalWrite(10, HIGH); // Habilita enfriador
74     } else {
75         lcd.setCursor(4, 1);
76         lcd.print("TEMPERATURA REGULAR");
77         digitalWrite(8, LOW);
78         digitalWrite(10, LOW);
79     }
80     digitalWrite(9, LOW); // Previene movimiento anticlockwise
81     digitalWrite(11, LOW);
82     delay(150);
83     lcd.clear();
84     intensity = readLuminosity(lumos); // Lectura luz
85     lcd.setCursor(1, 0);
86     lcd.print("Intensidad: ");
87     lcd.setCursor(12, 0);
88     lcd.print(intensity);
89     if (intensity < 100) { // Oscuro / Noche
90         lcd.setCursor(4, 1);
91         lcd.print("MUY OSCURO");
92         digitalWrite(light, HIGH);
93     } else if (intensity > 180) { // Mañana
94         lcd.setCursor(4, 1);
95         lcd.print("MUY CLARO");
96     } else {
97         lcd.setCursor(4, 1);
98         lcd.print("PERFECTO");
99     }
100     delay(150);
101     digitalWrite(light, LOW);
102 }

```

```

103
104 // Lectura de temperatura y conversión a Celsius
105 float readTemperature(char pin) {
106     float volts = analogRead(pin); // Codificación milivolts 8BCD enncoding
107     float value = map(((volts - 20) * 3.04), 0, 1023, -40, 125); // Rango de -40C a 125C
108     return value;
109 }
110
111 // Lectura de luz y conversión a luminosidad
112 float readLuminosity(char pin) {
113     float lum = analogRead(pin); // ( Blanco - Negro) 0 - 1023
114     float value = map(lum, 0, 700, 0, 255); //Rango de lectura intensidad de luz
115     return value;
116 }
117
118
119 // Función Auxiliar
120 void authenticate() {
121     lcd.setCursor(1,0);
122     if (char(Serial.read()) == 'w')
123         lcd.print('y');
124     else
125         lcd.print('n');
126 }
127 }
128

```

---

```

1 // Monitoreo Invernadero [S2]. Source : TinkerCad Community
2
3 #include <Servo.h> // Libreria Servo Motor para Bomba
4 #include <Keypad.h>
5
6 // Configuración del KeyPad
7 byte rows = 4;
8 byte cols = 4; // Setup 4*4 keypad
9 char keys[4][4] = { // 2D Array Claves (DP)
10     {'1', '2', '3', 'A'},
11     {'4', '5', '6', 'B'},
12     {'7', '8', '9', 'C'},
13     {'*', '0', '#', 'D'}
14 };
15 byte rowPins[4] = {9, 8, 7, 6}; // Pins para mapear filas
16 byte colPins[4] = {5, 4, 3, 2}; // Pins para mapear columnas
17
18
19 Keypad kp = Keypad(makeKeymap(keys), rowPins, colPins, rows, cols); // Interfaz Byte
20
21
22 Servo svr;
23
24 // Auxiliares
25 char gas = A0;

```

```

26  int echo = 10;
27  int trig = 11;
28  int serv = 12; // Control Angulo ServoMotor
29  int buzz = 13;
30
31  // Declaraciones Estaticas
32
33  String password= "0000"; // Contiene Clave Maestra 0000
34  String pressed=""; // Contiene Clave Ingresada
35
36  float distance; // Sensor Ultrasonido
37  float gaser;
38
39  int flag = 0; // trigger Autenticación
40
41  // Delaración de Funciones (Measure == Medida)
42  float measureDistance(int, int);
43  float measureSmoke(int);
44  void checkKeyPad();
45  void setup() {
46      Serial.begin(1156200); // Serial Monitor para Com.  pinMode(trig, OUTPUT); // Envío Señal
47      Serial.flush();
48
49      pinMode(trig, OUTPUT);
50      pinMode(echo, INPUT);
51      pinMode(gas, INPUT);
52      pinMode(buzz, OUTPUT);
53
54      svr.attach(serv);
55  }
56
57  void loop()
58  {
59      distance = 0.01723 * measureDistance(trig, echo); // 340 * 100 * 0.00001 / 2
60      Serial.print("Distance: ");
61      Serial.print(distance);
62      Serial.println();
63      if (distance > 200) { // Lectura al nivel de agua / profundidad
64          for (int i = 0; i <= 180; i++) {
65              svr.write(i);
66              delay(5);
67          }
68          delay(50);
69          for (int i = 180; i >= 0; i--) {
70              svr.write(i);
71              delay(5);
72          }
73      } else {
74          svr.write(0);
75      }
76
77      gaser = measureSmoke(gas); // Lectura de Gas

```

```

78     Serial.print("Gas: ");
79     Serial.print(gaser);
80     Serial.println();
81     if (gaser > 100) {
82         for (int i = 200; i <= 600; i += 50) {
83             tone(buzz, i, 10); // Tocar buzzing
84             delay(5);
85         }
86     } else {
87         digitalWrite(buzz, 0); // Parar buzz
88     }
89     svr.write(0);
90     delay(20);
91 }
92
93 // Distancia
94 float measureDistance(int triggerPin, int echoPin) {
95     digitalWrite(triggerPin, LOW);
96     delayMicroseconds(2);
97     // Trigger 10 microsegundos
98     digitalWrite(triggerPin, HIGH);
99     delayMicroseconds(10);
100    digitalWrite(triggerPin, LOW);
101    return pulseIn(echoPin, HIGH);
102 }
103
104 // Mide la cantidad de humo en el ambiente.
105 float measureSmoke(int pin) {
106     return analogRead(pin);
107 }
108
109 // Eventos - Matriz
110 void checkKeyPad() {
111     char key = kp.getKey();
112     if (int(key) != 0) {
113         if (key == 'C') { // Borrar ingreso por teclado
114             pressed = "";
115             delay(100); // Limpiar cache
116         } else if (key == '*') { // Ingreso check de pass
117             if (pressed == password) {
118                 flag = 1; // salida del loop de verificación
119             }
120             Serial.write(char('w'));
121         } else {
122             pressed.concat(key);
123             Serial.write(char(key));
124         }
125     }
126 }

```

---



## 4 Estructura Diseño - Sugerencia

A continuación se presenta una propuesta de diseño para la estructura final del invernadero. Se sugiere reducir las medidas presentadas para facilitar la construcción.

