

# Aplicación para el control automático de un Sistema de Invernadero

José de Jesús Toledo Mendoza  
Facultad de Ingeniería y Tecnología  
Ingeniería En Electrónica Y Telecomunicaciones  
Universidad de Montemorelos

Lo que se busca con este trabajo es desarrollar una aplicación en LabVIEW y Arduino para poder tener el control automático de un invernadero con la ayuda de actuadores y sensores para el cuidado en crecimiento de Almacigos. Al usar la placa Arduino-Ethernet Shield podremos conectarnos con el software LabVIEW por la interfaz web a través de la red.

*Keywords:* Control Automático, sistema de invernadero, LabVIEW y Arduino, Arduino-Ethernet Shield Alamacigos,

## Introducción

### Antecedentes

Los invernaderos son altas construcciones diseñadas para proteger temporalmente plantas. Su objetivo principal es proteger y mantener los cultivos para obtener un óptimo crecimiento antes de diciembre. Muchos cuentan con ventilación y riego para tener un control en su humedad y temperatura, aun que ya existen invernaderos más avanzados con sistemas más costosos y control computarizado o inteligente.<sup>1</sup>

Ya que en un campo de siembra se espera sembrar las plantas de mejor calidad. El uso de los almacigos permite tener un control del crecimiento homogéneo y seleccionar las plantas mejor desarrolladas para ser sembradas. Llevando un control de calidad en su crecimiento.<sup>2</sup>

Desde la antigüedad la agricultura ha estado en constante evolución tecnológica y científicamente. Los grandes cambios que se han hecho están revolucionando de forma espectacular la siembra y cosecha reduciendo el impacto sobre los recursos naturales, reduciendo los costos y aumentando la producción con mejor calidad.<sup>3</sup>

### Definición del problema

La Universidad de Montemorelos cuenta con un área designada para la siembra y un invernadero para el cuidado de los almacigos. El problema radica en la falta de personal para el cuidado del invernadero. Con el control automático del invernadero se solucionaría el problema.<sup>1</sup>

### Justificación

El campo de la agricultura es un área que está siendo transformada o renovada gracias a la tecnología. Grandes empresas, en el área de la ciencia y la tecnología, están trabajando para mejorar la calidad de trabajo y producción en

el campo agrícola. Gracias a esto se ha mejorado ciertos aspectos de la siembra sobre cómo tratar la tierra, las plantas, el agua, etc.<sup>2</sup>

La Universidad de Montemorelos cuenta con una huerta donde cosechan naranjas, además de eso se tiene un área de siembra de hortaliza e invernadero donde se tienen almacigos.

Sin embargo, no hay el personal suficiente para cuidarlo procurando que tenga el estado o ambiente adecuado para realizar su función. Con el sistema inteligente, se espera que se adecúen las condiciones indispensables para el desarrollo de la hortaliza; de esta manera ya que no será tan necesario la intervención de una persona para realizar estas actividades.<sup>4</sup>

### Objetivos

**General:** Desarrollar una aplicación en LabVIEW que pueda conectarse con Arduino para el control automático de un sistema de invernadero.

#### Específico:

- Comunicación de LabVIEW con Arduino.
- Crear un Control automatizado de los actuadores con Arduino.
- Medir la temperatura y humedad, con un sensor desde Arduino.
- Crear una aplicación en la LabVIEW que nos ayude a controlar las funciones del tiempo de riego y el control de la temperatura del invernadero.

### Hipótesis

Es posible realizar una aplicación de LabVIEW-Arduino que automatice el proceso en cuidado de Almacigos, según las características establecidas.

## Fundamentos teóricos

### Marco teórico

- **LabView:** LabVIEW es un software de ingeniería de sistemas que requiere pruebas, medidas y control con acceso rápido a hardware e información de datos. De igual manera permite diseñar interfaces de usuario mediante una consola interactiva basado en software, es compatible con herramientas de desarrollo similares y puede trabajar con programas de otra área de aplicación similar.<sup>5</sup>
- **Arduino:** se trata de una placa open hardware por lo que su diseño es de libre distribución y uso; Arduino es, ante todo, una empresa de software y hardware informático de código abierto.<sup>6</sup>
- **Arduino-Ethernet Shield:** es una placa que puede conectarse a la placa Arduino UNO y le permite conectarse a internet. Tiene un estándar de conexión RJ-45, con un transformador de línea integrada y Power over Ethernet habilitado. Esto se puede lograr gracias a la interfaz web con la que cuenta Arduino.<sup>7</sup>
- **Relés:** son los dispositivos destinados a producir determinadas modificaciones y protección, cuando se cumplen ciertas condiciones en un circuito eléctrico.<sup>8</sup>
- **IP:** Protocolos para la comunicación en internet. Define una red de conmutación de paquetes en la que la información a transmitir es fragmentada en trozos o paquetes.<sup>9</sup>
- **TCP/IP:** El aglutinante que hace posible aunar semejante diversidad es el conjunto de normas y lenguajes comunes de comunicación entre sistemas, conocido como familia de protocolos.<sup>9</sup>
- **Ethernet:** Es un sistema de comunicación de transmisión ramificada para transportar paquetes de datos digitales entre estaciones informáticas distribuidas localmente. se ha utilizado para construir sistemas que pueden verse como redes informáticas locales o multiprocesadores acoplados libremente.<sup>10</sup>
- **Aplicación (app):** El término “app” es la abreviatura de la palabra inglesa Application. La app, es una aplicación de software diseñada para ejecutarse en los smartphones (teléfonos inteligentes), tabletas y otros dispositivos móviles.<sup>11</sup>
- **Invernadero:** es una construcción agrícola con una cubierta traslúcida. En esta es posible reproducir o simular las condiciones climáticas más adecuadas para el crecimiento y desarrollo de los cultivos establecidos en su interior, con relativa independencia del medio exterior.<sup>1</sup>

- **Almácigos:** Los recipientes como paso previo a su cultivo en jardines o huertos. Estos recipientes son conocidos como almácigos. Este vocablo proviene del árabe, concretamente de la palabra almástaka que significa campo sembrado.<sup>12</sup>
- **Hortalizas:** El término hortalizas nombra a un conjunto de plantas cultivadas generalmente en huertos o regadíos, que se consumen como alimento, ya sea de forma cruda o preparada culinariamente, y que incluye a las verduras y a las legumbres verdes (las habas y los guisantes). Las hortalizas no incluyen a las frutas y a los cereales.<sup>1</sup>
- **Sistema automatizado:** La automatización es un sistema donde se transfieren tareas de producción, realizadas habitualmente por operadores humanos a un conjunto de elementos tecnológicos.<sup>13</sup>
- **Sistema inteligente:** son aquellos que presentan un comportamiento externo similar en algún aspecto a la inteligencia humana o animal. Se caracterizan por su capacidad para representar, procesar y modificar de forma explícita conocimiento sobre un problema, y para mejorar su desempeño con la experiencia. Esto les permite resolver problemas concretos determinando las acciones a tomar para alcanzar los objetivos propuestos, a través de la interacción con el entorno y adaptándose a las distintas situaciones.<sup>14</sup>

### Estado del arte

En la Universidad Francisco de Paula Santander, en Ocaña Colombia, se realizó una investigación sobre un Sistema Acuapónico para la producción tecnificada de peces y hortalizas en huertos urbanos. El cual funciona con un conjunto de técnicas donde el agua pasa por un proceso de recirculación en la cual los desechos producidos por los peces son absorbidos por los cultivos hidropónico, como se muestra en la figura 1. Estos desechos contienen nutrientes que se encargan de suplir las necesidades de los cultivos, donde el cultivo de plantas se produce en un medio libre de suelo, requiere un mínimo espacio, hay un mínimo consumo de agua, máxima producción y calidad.<sup>7</sup>

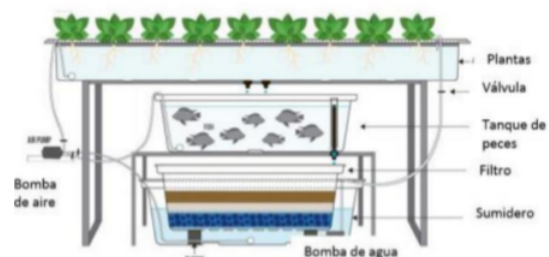


Figura 1: Sistema tradicional acuapónico

Una revista de Ingeniería e Investigación De la Universidad Nacional de Colombia, publico una investigación sobre las viviendas inteligentes (Domótica Fig. 2). Donde publicó que gracias a los avances tecnológicos Las actividades cotidianas En el hogar Puede ser realizadas sin una intervención humana directa, ya sea con un Smartphone, una Tablet, una laptop o computadora personal. Desde el punto de vista de automatización la domótica es un concepto interdisciplinario que se refiere a la integración de distintas tecnologías en el hogar mediante el uso simultáneo de las tecnologías. Además, su fin es mejorar la calidad de vida de los seres humanos. Otro término, para la domótica, es el control remoto que es utilizado para cualquier tipo de proceso, el cual permite resultados en el manejo del dispositivo que se quiere controlar. La flexibilidad de este tipo de control permite a las personas un mejor desempeño en sus actividades cotidianas.<sup>14</sup>

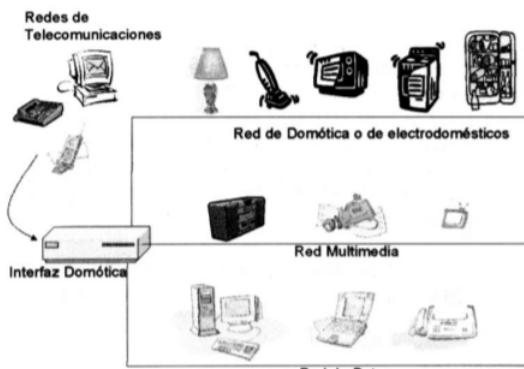


Figura 2: Sistema domótico centralizado para viviendas inteligentes.

**Presentación de resultados**

**Metodología**

**1. Pautar características necesarias en el invernadero:**

Se desarrolló un diagrama de flujo o mapa conceptual, para LabVIEW el cual nos ayudó a poder tener una mejor visión y control sobre la obtención de datos y el sistema. En este diagrama se puede observar las cuatro opciones con las que contar a nuestra aplicación.



Figura 3: Diagrama de flujo en LabVIEW resumido.

Siguiendo como referencia el diagrama de flujo de Lab-

VIEW resumido, (Fig. 3) se va a explicar el funcionamiento de la interfaz o tabla de control de la aplicación. El diagrama completo será puesto en los anexos. Cómo lo ha mencionado el diseño de nuestra aplicación o Software está realizada en LabVIEW, utilizando Arduino como la placa de desarrollo para tener control de los actuadores y sensores utilizados. Se va a observar cómo funciona la aplicación analizando las opciones de nuestra tabla de control y los diagramas de bloques para cada uno:

- **Visualización de datos:** Esta opción muestra los datos de temperatura y humedad, los cuales nos servirán para poder determinar el horario de riego para nuestro invernadero, estableciendo los límites de temperatura y humedad. También se tienen dos botones, que son “VER DATOS” y “GUARDAR” (Fig. 4). Cómo dice su nombre del primer botón es para ver los datos de temperatura y humedad gracias al sensor DHT11 y el segundo botón sirve para almacenar o guardar los datos obtenidos por el sensor (Fig. 5). Esta opción también cuenta con dos graficas que nos ayudan a ver cómo fluyen los datos obtenidos de temperatura y humedad.

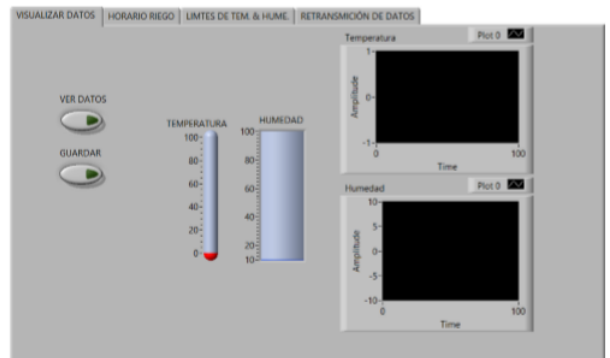


Figura 4: Opción, VISUALIZAR DATOS, en la tabla de control.

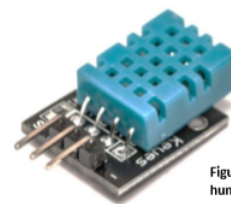


Figura 5: Sensor de humedad y temperatura DHT 11.

- **Horarios de riego:** Esta opción es algo fácil de manipular, lo que se hace es establecer un número de horas y los horarios en los que se van a accionar la bomba de

riego al día, 24horas(Fig. 6). Para este caso se usó una bomba de agua para pecera (Fig. 7). Esta opción cuenta con un botón el cual está enlazado a una pregunta “CREAR HORARIO?”, cuando se acciona este botón nos pedirá que se elija un numero de riegos y las horas en que se regara.



Figura 6: Opción, HORARIO DE RIEGO, en la tabla de control.



Figura 7: Bomba de agua para pecera.

#### ■ Límites de Temperatura y Humedad:

Cuándo se conocen los datos de temperatura y humedad, gracias a la visualización de los datos, se pueden designar límites los cuáles sirven para accionar ya sea el ventilador o el foco (Fig.8). Estos dos componentes están simulando la acción de calentar o enfriar el invernadero, dependiendo la necesidad de las hortalizas o almátigos (Fig. 9 y Fig.10). Estas funciones se accionan al seleccionar los botones para designar el máximo y el mínimo de Humedad y Temperatura. Una vez que lo se ha designado se presionar El botón de “HUMEDAD 2” o “TEMPERATURA 2” dependiendo lo que se quiere configurar en la aplicación.



Figura 8: Opción, LIMITES DE TEMPERATURA Y HUMEDAD, en la tabla de control.



Figura 9: Ventilador de 12V.



Figura 10: Foco de bombilla.

#### ■ Retransmisión de datos:

En esta última opción (Fig.11), de nuestra tabla de controles, la acción que se va a realizar es presionar el botón “ÚLTIMA ACTUALIZACIÓN” para poder visualizar las dos opciones o acciones que se han a realizar en nuestra aplicación. Se van a poder ver los límites de temperatura y humedad que se designaron y que accionarán el ventilador y el foco, así como el horario de riego establecido para el motor.



Figura 11: Opción, RETRANSMISIÓN DE DATOS, en la tabla de control.

#### ■ Diagramas de bloques para cada opción:

Primeramente, se creó una interfaz web capaz de controlar y monitorear sensores y actuadores esto a partir de la comunicación de Arduino y LabVIEW por medio de una red inalámbrica (Fig. 12). Para establecer la conexión se va a buscar la IP que se necesita para conectarnos a un modem mediante WIFI, en la computadora se busca la opción de propiedades de red la IP en la que

está conectado. Y la IP que va a utilizar es la anterior a la que está utilizando la laptop y el puerto 80 como por defecto. Esta conexión se tiene que establecer en cada una de nuestras opciones de la pestaña de controles. Se va a utilizar en el diagrama de bloques una unión de varios componentes o bloques, que trabajan en conjunto para hacer funcionar la aplicación de acuerdo a lo establecido. Y para concluir con la conexión de bloques se utiliza una sección de cierre de conexión la cual se utiliza también en cada una de las opciones haciendo un cierre de conexión (Fig. 13).

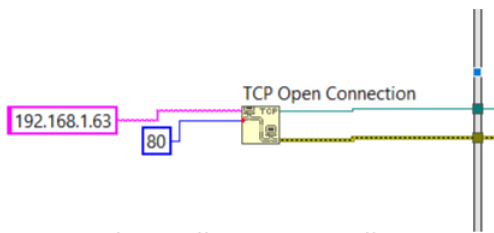


Figura 12: Sección para establecer la conexión por red.

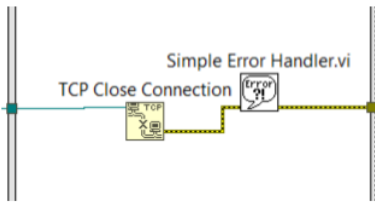


Figura 13: Sección para cerrar la conexión de red.

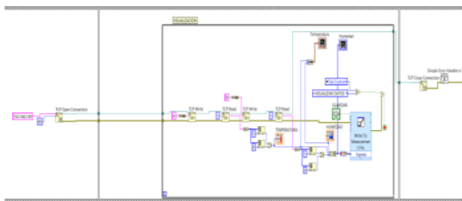


Figura 14: Diagrama de bloques para visualizar datos.

En la figura 14 se puede observar el diagrama de bloques para la opción de visualizar datos, de esta opción se va a escoger los bloques principales que se necesita para ejecutarla. Primero se obtienen los datos del sensor de temperatura mediante la conexión Arduino-LabVIEW, gracias a los componentes TCP Write y TCP Read. Para esta sección se ha dividido la obtención de datos en dos caracteres "A" y "B". Donde A son los datos de temperatura y B son los datos de humedad. Cuando los datos son obtenidos se deben acomodar para lo cual se utiliza los bloques Index Array, este bloque ayuda a acomodar los valores numéricos de una forma lógica (Fig. 15). Para poder visualizar los datos del sensor

se va a utilizar unos indicadores de barra (Fig. 16). De igual manera se conectan dos bloques gráficos para visualizar de forma gráfica el comportamiento de los datos de temperatura y humedad (Fig. 17). Y casi, por último, se usó un bloque Write To Measurement File Express VI, que es un componente para guardar la información en un determinado lugar (Fig. 18).



Figura 15: Bloque Index Array

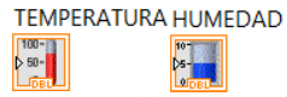


Figura 16: Bloques indicadores.



Figura 17: Bloques indicadores.

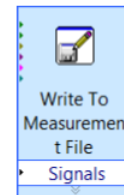


Figura 18: Bloque para guardar datos.

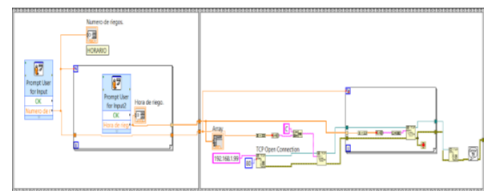


Figura 19: Diagrama de bloques para visualizar datos.

En la figura 19 se puede observar el diagrama de bloques para la opción de horario de riego. En esta opción se escogió los bloques Prompt User for Input Express VI (Fig.20), los cuales nos ayudan para establecer una comunicación con el usuario mediante una solicitud para ingresar una información. En este caso se utilizan dos bloques, en el primer bloque se hace la pregunta para saber el número de horas en que se quiere regalar, el segundo bloque es para preguntar a que hora del día se eran esos riesgos, tomando en cuenta que el día consta de 24 horas. Para poder responder la información que nos piden los bloques Prompt User for Input Express VI,

se va a utilizar los bloques de Control Numérico (Fig. 21) y establecer el valor necesario.

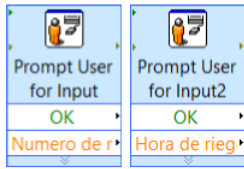


Figura 20: Bloques Prompt User for Input Express

Numero de riegos.



Hora de riego.



Figura 21: Bloques de Control Numérico

El bloque Prompt User for Input Express VI como el Control Numérico para establecer la hora de riego, están dentro del cuadro For Loop (Fig. 22), lo que significa que está configuración se estará repitiendo hasta que se quiera hacer una nueva configuración por el usuario. En cuyo caso se tiene que repetir nuevamente la configuración estableciendo el número de horas y las horas de riego.

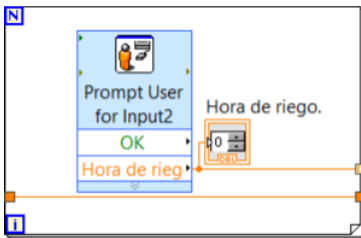


Figura 22: Estructura For Loop.



Figura 23: Bloque de Incremento.

Array output array



Figura 24: Bloques de Array.



Figura 25: Estructura Insert Into Array Function.

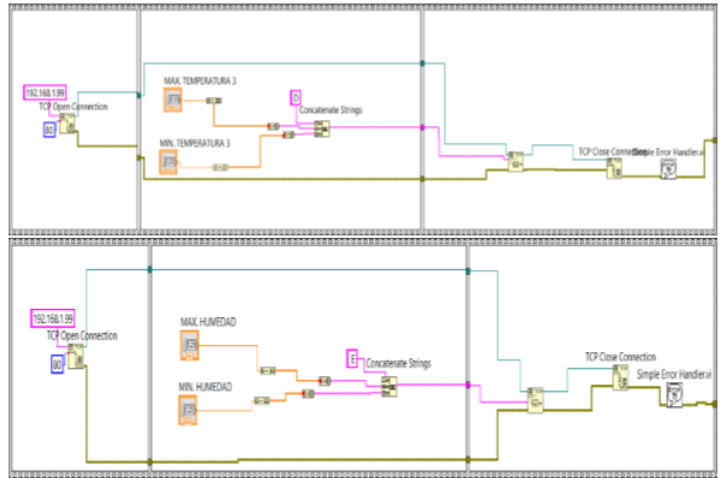


Figura 26: Diagrama de bloques para Limites de Temperatura y humedad.

En la figura 26 se puede observar el diagrama de bloques para la opción de Limites de Temperatura y Humedad, se ha dividido nuestra obtención de datos en dos caracteres “D” y “E”. Esta opción consta de dos casos, establecer un mínimo y un máximo de Temperatura y un mínimo y un máximo de Humedad. Para esto se va a utilizar cuatro bloques de control numérico, como los que se utilizaron en la opción de Horario para establecer el número de horas y las horas de riego (Fig. 21), dos para temperatura y dos para humedad con su nombre correspondiente. Para los valores de los bloques numéricos se utiliza el bloque Concatenate Strings Function (Fig. 27), que funciona para concatenar la salida de los bloques numéricos y ordenarlos de menor a mayor.

### Concatenate Strings



Figura 27: Bloque Concatenate Strings Function.

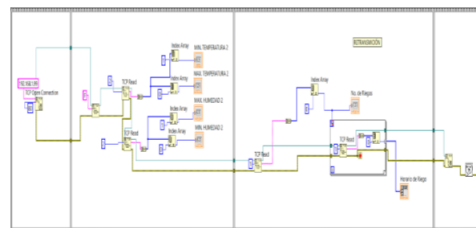


Figura 28: Diagrama de bloques para Retransmisión de datos.

En la figura 28 se puede observar el diagrama de bloques

para la opción de Retransmisión de datos. Estos datos van a ir enlazados a un caracter de entrada llamado "F". Como se puede observar en esta opción se hace la unión de los bloques utilizados en estas dos opciones. Como dice su nombre retransmite los datos para poder observar como está funcionando nuestra aplicación, cómo se puede observar en la figura 11.

El siguiente diagrama de flujo o mapa conceptual es para Arduino, es más extenso ya que lleva la configuración y programación que se necesita para poder obtener los datos del sensor y controlar el modulo para Arduino RTC DS1302 (Fig.30), que es un reloj en tiempo real, para poder establecer el horario en esta aplicación.

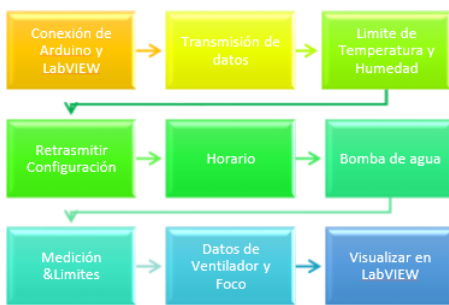


Figura 29: Diagrama de flujo para Arduino, resumido.



Figura 30: Reloj RTC DS1302 para Arduino.

Siguiendo como referencia el diagrama de flujo de Arduino resumido, (Fig. 29) se explicará la programación y el funcionamiento diseñado para la aplicación. El diagrama completo será puesto en los anexos.

Se colocará parte de la programación en cada caso y en los anexos estará la programación completa. Lo que está escrito después de las diagonales (//) son las descripciones o comentarios.

Estas son las variables que se utilizaron para la programación y las funciones principales al ejecutarse el programa:

#### VARIABLES DE ENTRADA Y SALIDA.

```

virtuabotixRTC myRTC(6, 7, 8); // pines para conectar el reloj
byte Tiempos[] = ;//variable para las horas de riego
byte DimencionHoras;//variable para la dimensión de los números de horas
int Cont = 1;
  
```

```

int T_anterior;// variable usado para el horario
char caracterEntrada; //variable para llamar el canal de entrada
byte vector1[2], vector2[2]; //Retran-Tem,RetranHum;
char canal; //variable
/int valor_analog;

int Switch = 4;//pin o puerto conectado en la placa
int motor = 11;//pin o puerto conectado en la placa
int foco = 12;//pin o puerto conectado en la placa
int ventilador = 13;// pin o puerto conectado en la placa
///Variables de actualización temperatura y humedad
int t = 2;
int MaxTem = 0; // variable de entrada
int MinTem = 0; // variable de entrada
int MaxHum = 0; // variable de entrada
int MinHum = 0; // variable de entrada
byte RetranTem[2], RetranHum [2];
  
```

#### Funciones y parámetros.

```

void setup()
Serial.begin(9600); // se abre el puerto serial y se configura a los bps por segundo que trabaja Arduino

pinMode(motor,OUTPUT); //se declara como pin de salida
pinMode(foco, OUTPUT);
pinMode(ventilador, OUTPUT);
pinMode(Switch, INPUT; //se declara como pin de entrada

void ControlTemMax() //función para el control de ventilador y foco de cuando temperatura en máximo
digitalWrite(ventilador , HIGH); //variable para encender
digitalWrite(foco , LOW; //variable para encender
void ControlTemMin() //función para el control de ventilador y foco cuando temperatura en mínimo
digitalWrite(foco , HIGH);
digitalWrite(ventilador , LOW);
void ControlTemNeu() //función para el control de ventilador y foco cuando temperatura cuando
digitalWrite(ventilador , LOW);
digitalWrite(foco , LOW);
  
```

### ■ Conexión de Arduino y LabVIEW.

En esta sección, cómo lo dice su nombre, es para establecer una conexión entre Arduino y LabVIEW y así poder comenzar la transmisión de datos obtenidos por los sensores y actuadores.

```
// CONFIGURACION DE LA PLACA
ETHERNET
byte mac[] = 0x00, 0x22, 0x68, 0x2F, 0xA0,
0x9C ; // Dirección MAC
IPAddress ip(192,168,1,63); // Dirección IP
EthernetServer server(80); // puerto de comuni-
cación HTTP
```

### ■ Transmisión de datos.

Una vez que ya se estableció la conexión se una pregunta para saber si se quiere transmitir los datos o no, tomando en cuenta que tal vez la aplicación ya tenía algo transmitido. De otro modo si se quiere transmitir los datos se va a comenzar una comunicación. Sí no, sólo se hace la comunicación, pero sin ninguna transmisión de datos.

Sí se decide transmitir los datos se tiene que hacer un stop en LabVIEW, este stop va a ser tanto para el horario cómo para los límites de temperatura y humedad si es que ya había unos establecidos. Y se va a trabajar con los datos que se van a mandar del sensor de temperatura y humedad, así como tomar en cuenta la hora que se estableció en el programa.

```
// CONFIGURACIÓN DE LA COMUNICA-
CION TCP/IP
Ethernet.begin(mac, ip); // Inicialización de la
pila TCP/IP
server.begin(); // Inicialización del servidor
if (digitalRead(Switch) == 1 )
EthernetClient client = server.available(); // Se
crea el objeto cliente
if (client) // Si se recibe una petición de un clien-
te
while(client.connected()) // Mientras el cliente
esté conectado
if (client.available()) // Si el cliente está dispo-
nible
caracterEntrada = client.read(); // toma un nuevo
byte
// INICIO DE COMANDO 'A' Y 'B' (LECTU-
RA DEL SENSOR DHT11)
if (caracterEntrada == 'A')
int t = analogRead(A0);
```

```
Serial.println (t);
```

```
vector1[0] = lowByte(word(t)); // genera el pri-
mer byte
```

```
client.write(vector1, 1); // envía los dos bytes
convertidos en binario
```

```
if (caracterEntrada == 'B')
```

```
int h = analogRead(A5);
```

```
Serial.println (h);
```

```
vector2[0] = lowByte(word(h)); // genera el pri-
mer byte
```

```
client.write(vector2, 1); // envía los dos bytes
convertidos en binario
```

### ■ Límite de Temperatura y Humedad

Para poder establecer límites de temperatura y humedad se debe tomar en cuenta los datos que se han obteniendo del sensor y también del tipo de cultivo que hay en nuestro invernadero.

```
// INICIO DE COMANDO 'D' Y 'E' (LECTU-
RA DE LIMITES DE TEMPERATURA Y HU-
MEDAD)
```

```
if (caracterEntrada == 'D') //carácter de tempe-
ratura
```

```
Serial.println("Temperatura M/m");//Función
para indicar la conexión con LabVIEW
```

```
caracterEntrada = client.read(); //sirve interpre-
tar el carácter de entrada
```

```
MaxTem = (caracterEntrada); //operación para
recibir el limite max de temp
```

```
caracterEntrada = client.read();
```

```
MinTem = (caracterEntrada); //operación para
recibir el limite min de temp
```

```
Serial.println(MaxTem); //impresión de valores
en el monitor serial
```

```
Serial.println(MinTem); //impresión de valores
en el monitor serial if (caracterEntrada == 'E')
//carácter de humedad
```

```
Serial.println("Humedad M/m");
```

```
caracterEntrada = client.
```

```
MaxHum = (caracterEntrada); //operación para
recibir el limite max de humedad
```

```
caracterEntrada = client.read(); MinHum = (ca-
racterEntrada); //operación para recibir el limite
min de humedad
```



```
Serial.println(MaxHum); //impresión de valores
en el monitor serial
```

```
Serial.println(MinHum); //impresión de valores
en el monitor serial
```

#### ■ Retransmitir configuración.

Esta parte de retransmisión de datos la se utiliza tanto para el horario cómo para los límites cuando ya se ha decidido que establecer.

```
// INICIO DE COMANDO 'F' (RETRANSMI-
CIÓN DE DATOS)
```

```
if (caracterEntrada == 'F') //carácter
de temperatura RetranTem[0] = lowBy-
te(word(MaxTem)); // genera el primer byte
RetranTem[1] = lowByte(word(MinTem)); //
genera el segundo byte client.write(RetranTem,
2); // envía los dos bytes convertidos en binario
RetranHum[0] = lowByte(word(MaxHum));
// genera el primer byte RetranHum[1] =
lowByte(word(MinHum)); // genera el se-
gundo byte client.write(RetranHum, 2); //
envía los dos bytes convertidos en binario
client.write(DimencionHoras + 1); //envia
o convierte los byts for (byte i = 0; i <=
DimencionHoras; i++) //se un bucle para el
tamaño de horas client.write(Tiempos[i]);//se
mandan los datos
```

#### ■ Horario

Para establecer un horario de riego primeramente se debe tener una hora o un tiempo en el programa y por eso se agrego el reloj. Este reloj ayudara para monitorear el horario de riego que se establezca en el invernadero. En otras palabras, se establece un horario para saber cuándo se debe prender y apagar la bomba de agua.

```
// INICIO DE COMANDO 'C' (LECTURA DE
UNA ENTRADA ANALÓGICA DEL RELOJ)
```

```
if (caracterEntrada == 'C') //carácter de tempe-
ratura
canal = client.read(); // toma un nuevo byte
DimencionHoras = lowByte(word(int(canal)));
//se declara o llama a la variable para ver su ta-
maño
int x = DimencionHoras;
Serial.println(x);
for (int i = 0; i <x; i++)
```

```
char canalHoras = client.read();//LabVIEW da
un carácter y se tiene que convertir
```

```
Tiempos[i] = (int)canalHoras;
```

```
DimencionHoras = lowByte(i); //se declara o
llama a la variable para ver su tamaño
```

```
Serial.print(i=");
```

```
Serial.println (i);
```

```
Serial.println(Tiempos[i]);
```

#### ■ Bomba de agua

La siguiente programación es para encender la bomba dependiendo el horario que se haya establecido

```
myRTC.updateTime();
```

```
int xz = myRTC.seconds; // variable para co-
rroborar que el tiempo coincida en una acción
a realizar
```

```
//se declara a la variable para ver su tamaño //
condición para saber la hora en que se va a regar
o encender la bomba de agua
```

```
if (xz == Tiempos[Cont] + 1)
```

```
Cont++;
```

```
if (Cont >= DimencionHoras)
```

```
Cont = 1;
```

```
Tanterior = Tiempos[1];
```

```
if(xz == Tiempos[Cont])
```

```
digitalWrite(motor, HIGH); else
```

```
digitalWrite(motor, LOW);
```

```
delay(500);
```

#### ■ Medición y Límites.

La medición de límites es una visualización nuevamente de los datos que se están obteniendo por medio del sensor y los límites que se han establecido para el funcionamiento del ventilador y el foco.

#### ■ Datos de Ventilador y Foco

Una vez que ella se han establecido los límites que se desean se va ver la programación para el control del ventilador y el foco. En dónde se establece que, si la temperatura supera el máximo establecido se prendera el ventilador, pero si la temperatura es menor que es mínimo establecido se encender el foco. Como ya se mencionó anteriormente el ventilador y el foco están simulando lo que es el enfriamiento y el calentamiento del invernadero según lo deseado por el usuario.

```

if (!(MaxTem == MinTem || MaxTem == 0 ||
MinTem == 0 || MaxTem < MinTem))
if (t >= MaxTem) // en esta seccion si la tempe-
ratura es mayor o igual al maximo se enciende
el ventilador y el foco se apaga
ControlTemMax();
else if (t <= MinTem) //si la temperatura es me-
nor o igual a la mínima se enciende el foco y el
ventilador se apaga
ControlTemMin(); else
ControlTemNeu(); //si la temperatura es neutral
o está dentro de los limites el foco y el ventilador
se apagan
Serial.println(4);

```

#### ■ Visualización en LabVIEW

Y por último la programación que se ha establecido se puede visualizar en la aplicación una vez que la halla utilizado.

#### Resultados

Los resultados fueron los deseados ya que se logró el objetivo establecido para esta aplicación. Se pudo establecer comunicación entre Arduino y LabVIEW mediante una interfaz web, se creó un control automático para los actuadores con Arduino, se pudo medir la temperatura y la humedad con el sensor desde Arduino y se creó una aplicación en LabVIEW que ayuda a controlar las funciones del tiempo de riego y el control de la temperatura en el invernadero.

#### Discusión

Este proyecto llego a cumplir el propósito de controlar el riego de agua con un determinado tiempo y la temperatura que el invernadero necesita las cuales se consideraron cómo las variables más importantes para el buen desarrollo de los almácigos. Se consideró que este proyecto se encuentren una etapa básica ya que, por cuestiones de dinero, tiempo y otras limitantes no lo se pudo llevar a la aplicación en un invernadero real, pero eso no impide su funcionamiento.

#### Conclusiones

La aplicación que se desarrolló está en excelente funcionamiento sí se acepta que puede ser la base para una aplicación más amplia, considerando incluir en un futuro un aire acondicionado el cuál sustituya el funcionamiento del ventilador el foco para mantener a una temperatura más estable en el cuidado de los almácigos.

Y el principal beneficio y trabajo a futuro es poder instalarlo en el invernadero de la Universidad de Montemorelos.

#### Referencias

1. B. Santos, H. Obregón-Olivas, and T. Salamé-Donoso, "Producción de hortalizas en ambientes protegidos: estructuras para la agricultura protegida," Publicación HS1182. IFAS Extension, UF Dep. Hortic. Sci. Univ. Florida, Gainesville, FL, 2010.
2. G. Q. Roldán and C. M. Soto, "Evaluación de sustratos para almácigos de hortalizas," *Agron. Mesoam.*, vol. 16, no. 2, pp. 171–183, 2005.
3. A. Turrent-Fernández and J. I. Cortés-Flores, "Ciencia y tecnología en la agricultura mexicana: I. Producción y sostenibilidad," *Terra Latinoam.*, vol. 23, no. 2, pp. 265–272, 2005.
4. A. Turrent-Fernández and J. I. Cortés-Flores, "Ciencia y tecnología en la agricultura mexicana: II. Producción de alimentos," *Terra Latinoam.*, vol. 23, no. 2, pp. 273–281, 2005.
5. H.-P. Halvorsen, "Introduction to LabVIEW," Univ. Coll. Southeast Norw., 2014.
6. S. A. Arduino, "Arduino," Arduino LLC, 2015.
7. J. M. R. Gutiérrez, "Arduino+ Ethernet Shield," línea].
8. A. P. París, "Relés electromagnéticos y electrónicos. Parte I: relés y contactores," *Vivat Acad.*, no. 50, pp. 1–24, 2003.
9. M. Á. Sanz, "A, B, C de Internet," *RedIRIS-Boletín la red Nac. I+ D*, vol. 28, pp. 15–30, 1994.
10. P. A. Haya, "Arquitecturas para el desarrollo de entornos inteligentes." 2001.
11. C. Linea Verde, "¿Qué es una APP?"
12. J. Navarro, "Definición de Almacigo," Definición ABC, 2017.
13. "Sistema Automatizado."
14. L. F. H. Quintero, "Viviendas inteligentes (domótica)," *Ing. e Investig.*, vol. 25, no. 2, pp. 47–53, 2005.