



POSGRADO EN
HIDROCIENCIAS
MAESTRÍA - DOCTORADO
CAMPUS MONTECILLO

PSICRÓMETRO

RUBÉN ESPARZA OROZCO
BARTOLOMÉ ESPINOSA ESPINOSA

NOVIEMBRE 2021



Características básicas del sensor

- (1) Costo de \$59 pesos mexicanos, disponible en mercado libre
- (2) Voltaje de alimentación: 3 V a 5.5 V
- (3) Rango de temperatura: -55°C a 125°C
- (4) Error en el termómetro: $\pm 0.5^{\circ}\text{C}$ (-10°C a $+85^{\circ}\text{C}$).
- (5) Resolución: programable para 9, 10, 11, 12 bit de resolución con tiempos de respuesta de 93.75, 187.5, 375, 750 ms para los cambio de temperatura de 0.5°C , 0.25°C , 0.125°C , 0.0626°C . La resolución puede modificarse por programación.
- (6) Memoria de 64 byts (8bit): contiene una dirección única que permite identificar entre los diferentes termómetros conectados al bus de comunicación ONEWIRE. El primer bit identifica el tipo de componente. Por ejemplo en el caso del 18 B 20 el primer byt contiene un valor de 28.

Líneas de conexión



1. Tierra
2. Tensión
3. Datos

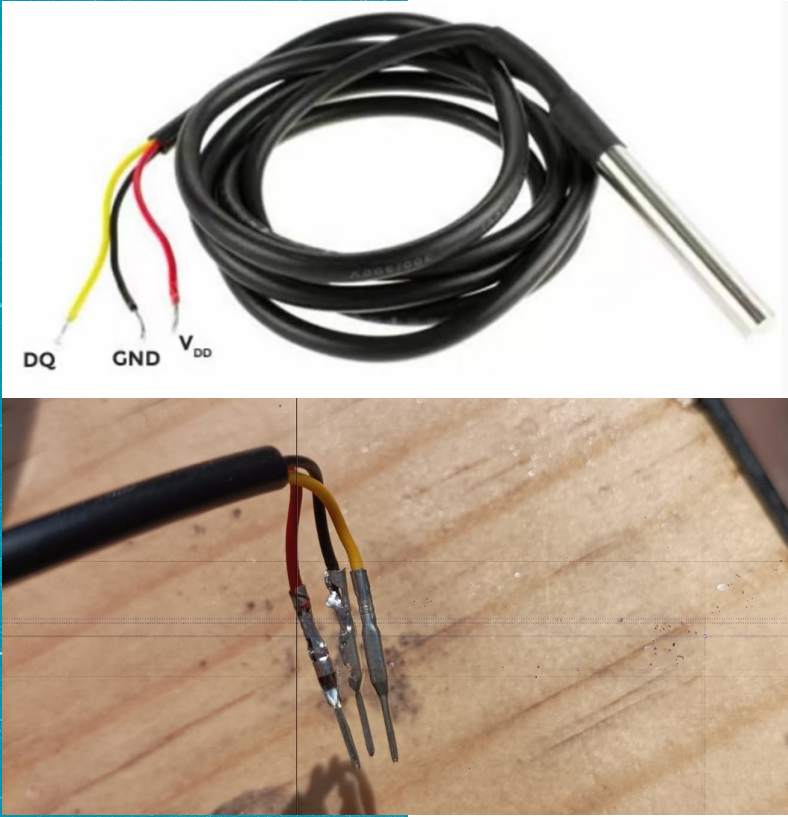


Diagrama de montaje

DS18B29 en serie

El protocolo de comunicación OneWire permite conectar los sensores en serie.

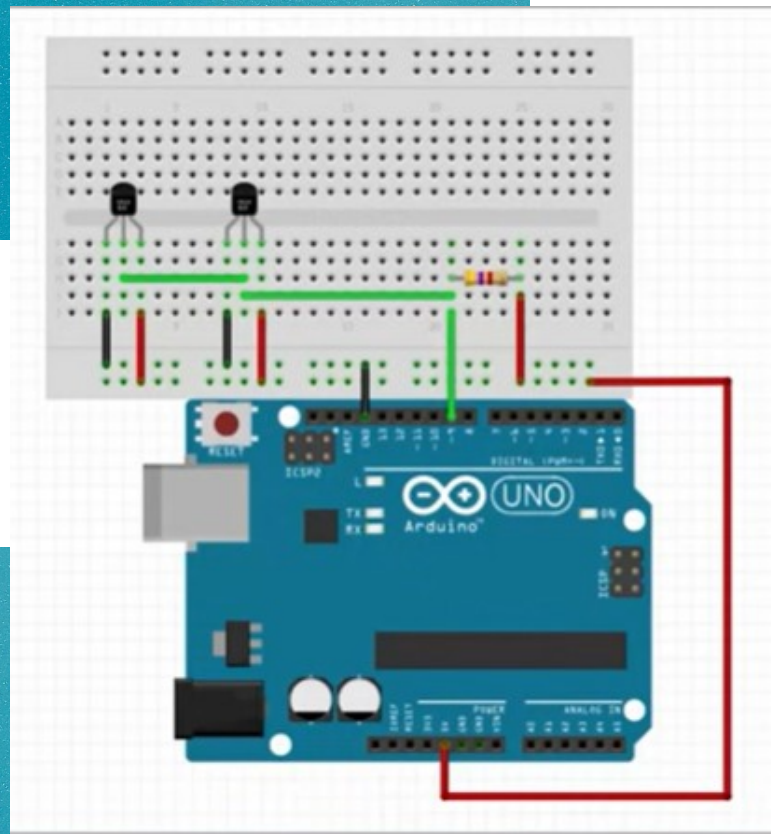
Resistencia en pull up

Recepción de datos

La conexión de datos se entrega a Arduino en un solo pin digital

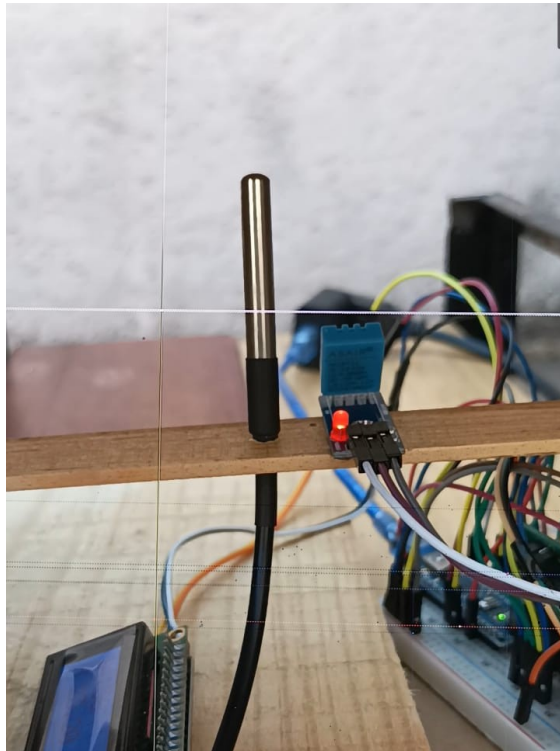
Alimentación

3 V – 5.5 V



Emplazamiento de sensores

Garita meteorológica



Para el emplazamiento de los sensores DS18B20 y DHT11 se utilizó una pequeña viga de madera para su montaje y una garita de cartón para protegerlos de la incidencia directa de los rayos del Sol sobre ellos.



Implementación del sensor DS18B20 con Arduino



Librerías

Librerías OneWire y
DallasTemperature



SetUp()

Inicia el objeto del sensor.



Loop()

Solicitud de datos al termómetro
y salida visual de los mismos.

Ejemplo: Código Arduino

Declaración de constante

`pinDatosDQ = 9;`

Comando de solicitud de dato

`sensorDS18B20.requestTemperatures();`

Identificación termómetro por índice

`sensorDS18B20.getTempCByIndex(0);`

```
//incluir las librerías
#include <OneWire.h>
#include <DallasTemperature.h>

//constante
const byte pinDatosDQ = 9

//Solicitud de los objetos
//Indicar en que pindigital se reciben los datos
OneWire oneWireObjeto(pinDatosDQ);
//Indica la dirección donde esta almacenado el dato
DallasTemperature sensorDS18B20(&oneWireObjeto);

void setup() {
  // put your setup code here, to run once:
  //Inicia comunicación con monitor serie
  Serial.begin(9600);
  //Inicia el bus OneWire
  sensorDS18B20.begin();
}

void loop() {
  // put your main code here, to run repeatedly:
  Serial.println("ENVIANDO COMANDO");
  //Solicitud de dato al sensor
  sensorDS18B20.requestTemperatures();
  Serial.print("Temp");
  //Obtiene la temperatura en el sensor de indice 0
  Serial.println(sensorDS18B20.getTempCByIndex(0));
  delay(1000);
}
```




Algoritmo de toma de mediciones de temperatura y humedad relativa con Arduino

La salida final del algoritmo son dos archivos .txt con el contenido de los promedios de temperatura por minuto y por hora.

Quedan pendientes adjuntar el procedimiento de determinación de valores de máxima y mínima así como su almacenamiento en los archivos correspondientes.

Void loop(){.....}

Referencia tiempo

myRTC.updateTime();

Contadores y ciclo for (c1, c2, p){ }

Contadores suma;
Contadores tiempo;

Ejecución de funciones

PromedioTBHSMinuto();
GrabaTBHSMinuto();
PromedioTBHSHora();
GrabaTBHSHora();

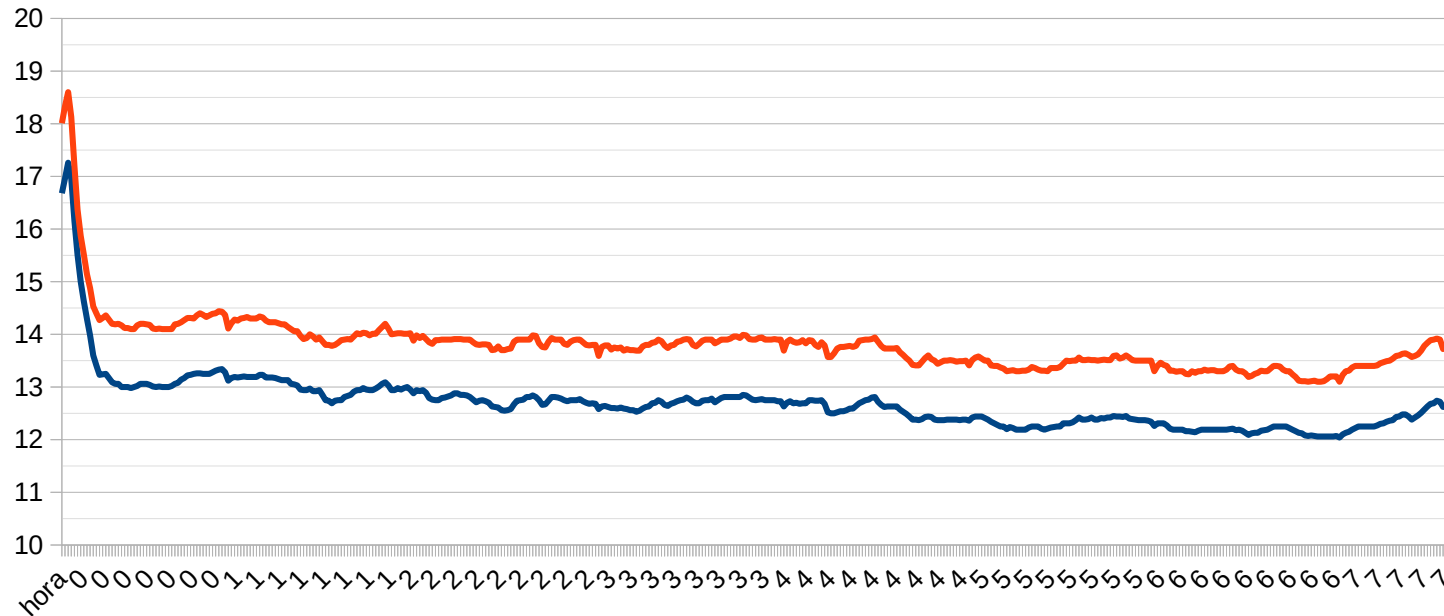
```
//4. LOOP()  
void loop() {  
  myRTC.updateTime();  
  cont_min = 1;  
  sumatbhmin = 0;  
  sumatbsmin = 0;  
  sumahrmin = 0;  
  for(myRTC.minutes <60; myRTC.minutes > 0 ;cont_min++){  
  
    myRTC.updateTime();  
    cont_seg = 1;  
    sumatbhseg = 0;  
    sumatbsseg = 0;  
    sumahrseg = 0;  
    for(myRTC.seconds <60; myRTC.seconds > 0 ;cont_seg++){  
      TBHSIns();  
      PromedioTBHSMinuto();  
      myRTC.updateTime();  
    }  
    //PromedioTBHHora();  
    GrabaTBHSMinuto();  
    delay(1000);  
    PromedioTBHSHora();  
    myRTC.updateTime();  
  }  
  GrabaTBHSHora();  
  delay(60000);  
}
```

Temperatura

Promedios por minuto (7 horas)

Sensores

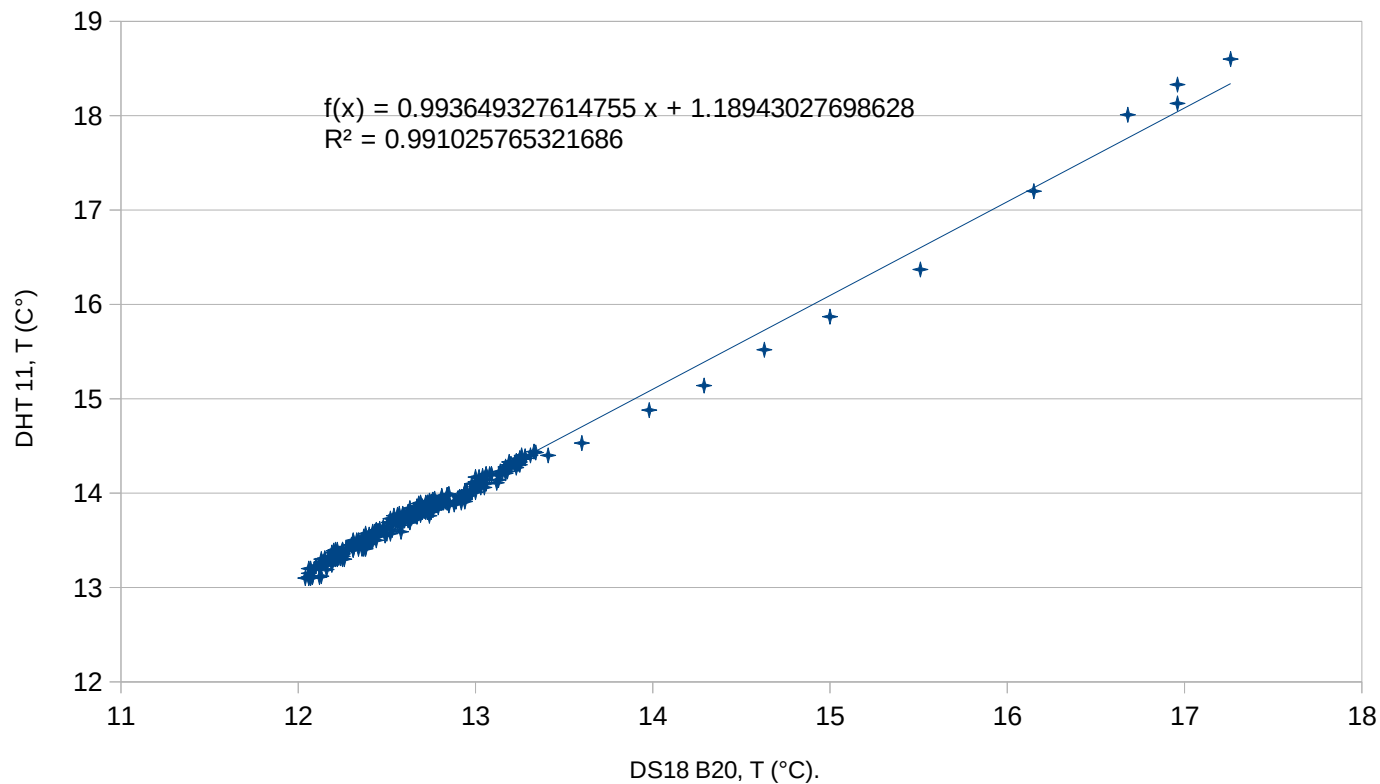
— DS 18 B20 — DHT11



Comparación de mediciones de temperatura

Las curvas están hechas con los promedios por minuto de las mediciones de temperatura de ambos termómetros.

Correlación



Ecuación de línea de tendencia

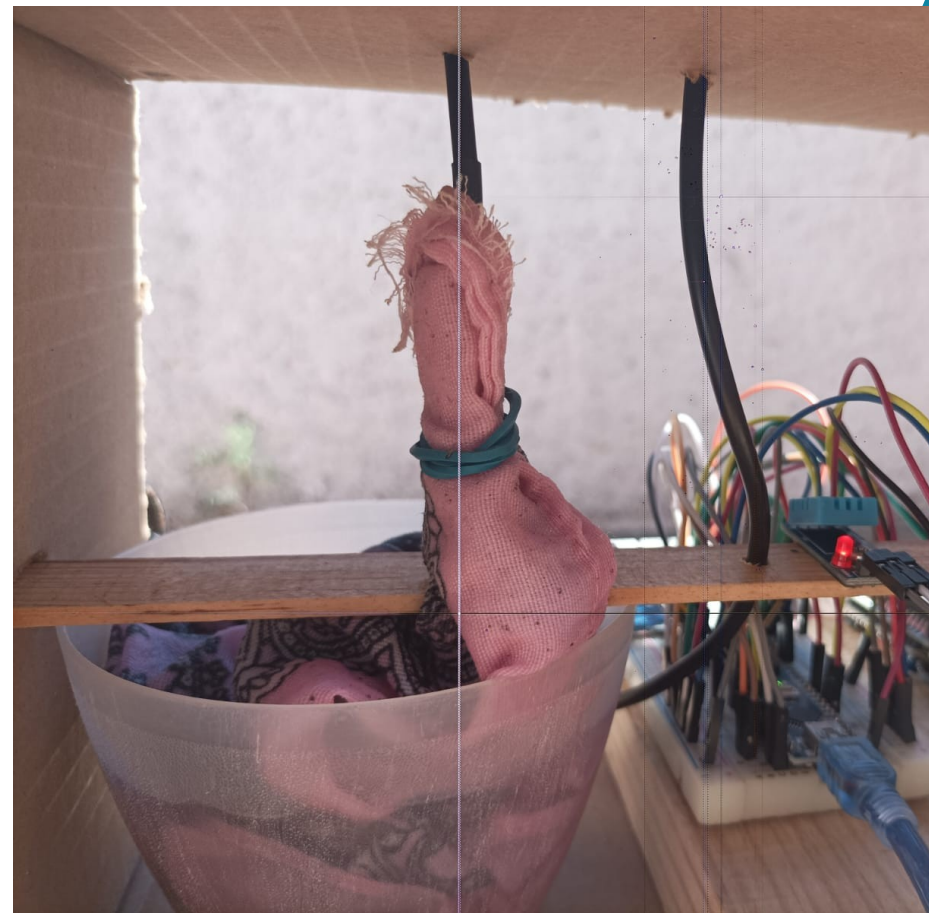
La pendiente de la recta $m=0.99$, revela la tasa de cambio en T (DHT11) por cada grado °C que DS18B20 se modifica.

La constante 1.18 es la medición esperada en DHT11 cuando DS18B20 registre 0°C.

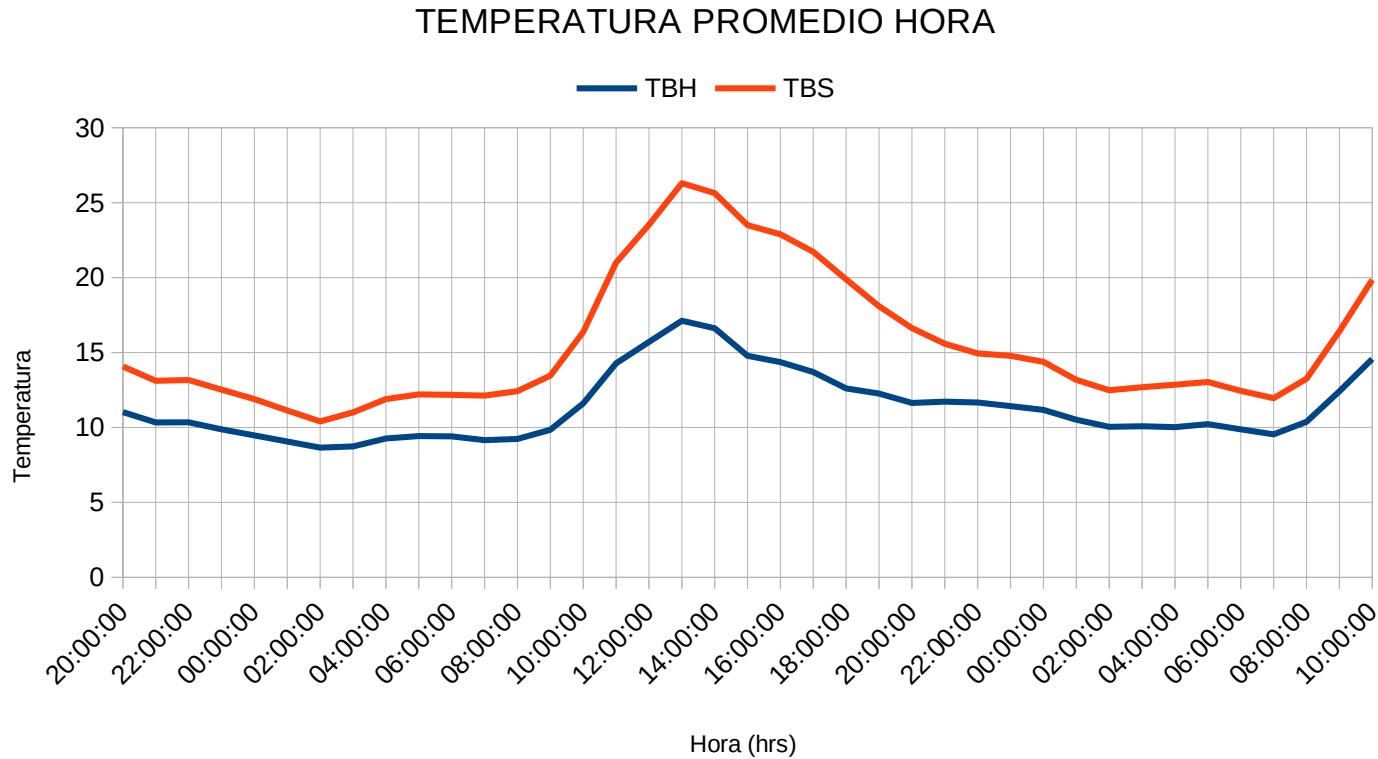
Termómetro bulbo húmedo

Para generar la condición de temperatura en punto de saturación o punto de rocío se arropó el termómetro DS18B20 con un paliacate y se le aseguro con una liga plástica.

Se eligió esta tela porque el agua en el recipiente sube con facilidad por capilaridad a través de él.



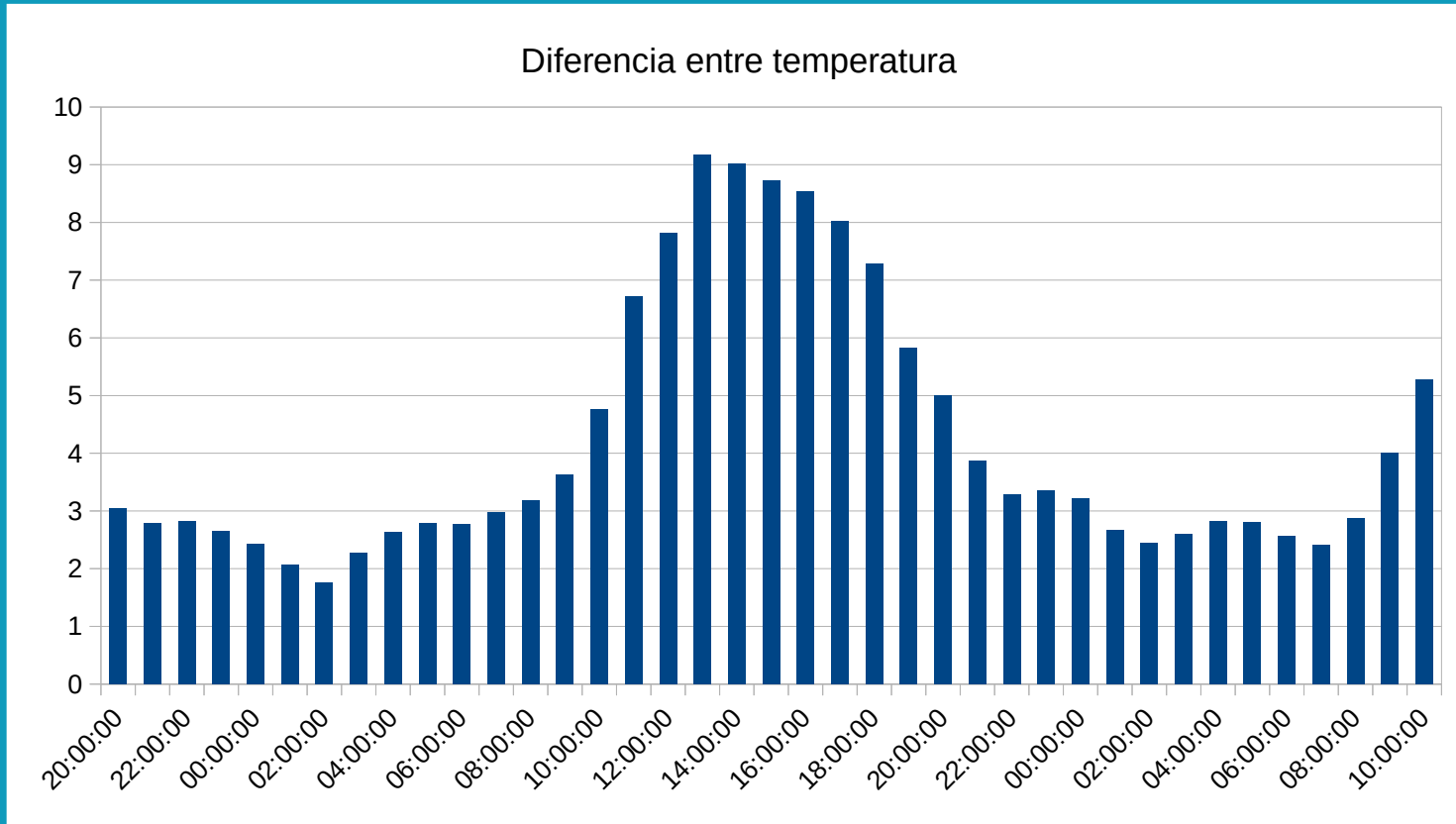
Temperatura



Curvas de temperatura bulbo húmedo bulbo seco

Las curva están hechas con los promedios por hora de las mediciones de temperatura de ambos termómetros.

Temperatura



Curvas de temperatura bulbo húmedo bulbo seco

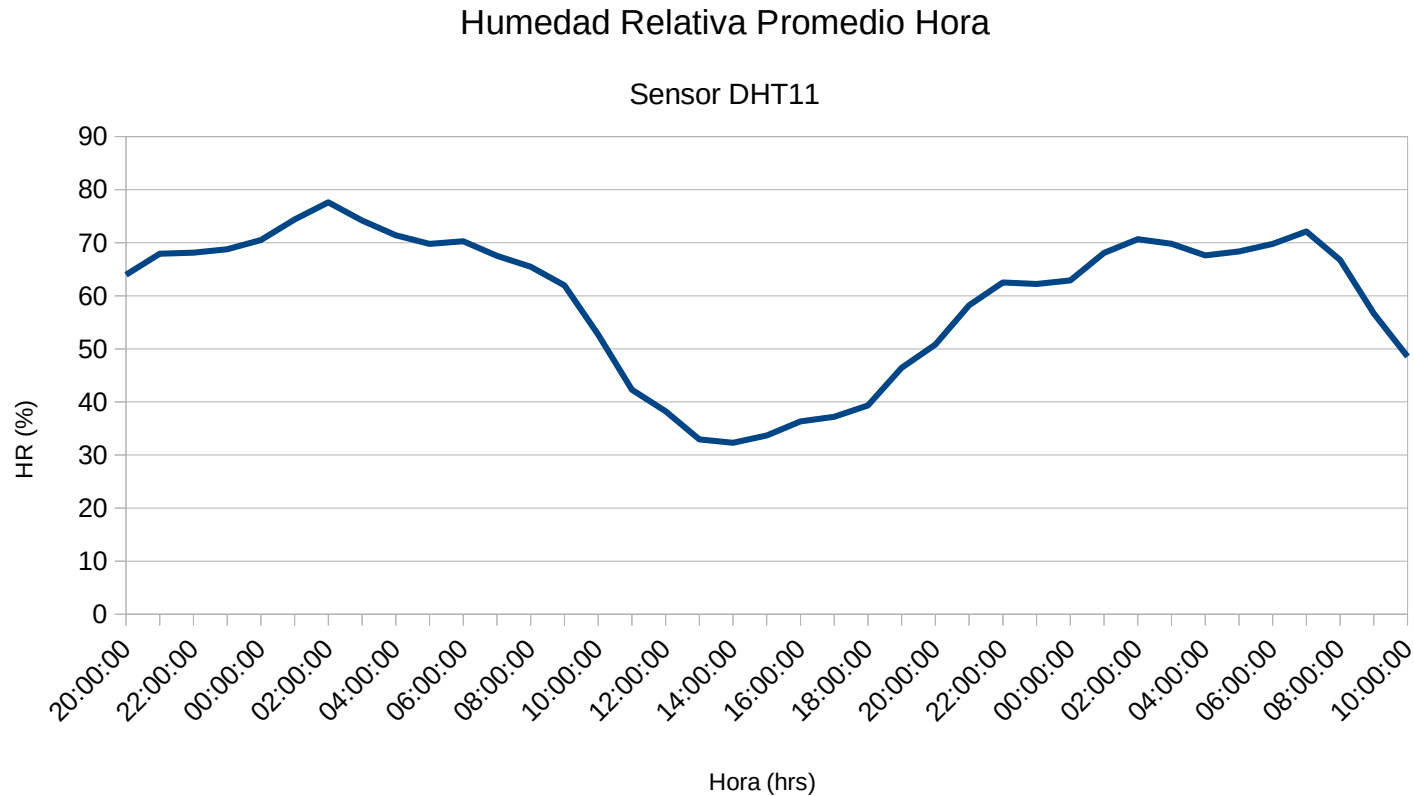
La diferencia mínima entre ambos registros de temperatura es de 1.75 que ocurrió a las dos de la mañana y diferencia máxima fue de 9.18 que ocurrió a las dos de la tarde.

Humedad relativa

Se obtuvieron datos de humedad relativa a través del sensor DHT11.

Los datos obtenidos fueron comparados con los datos de H. R. determinados con una carta psicrométrica en base a los datos de temperatura de bulbo húmedo y bulbo seco.

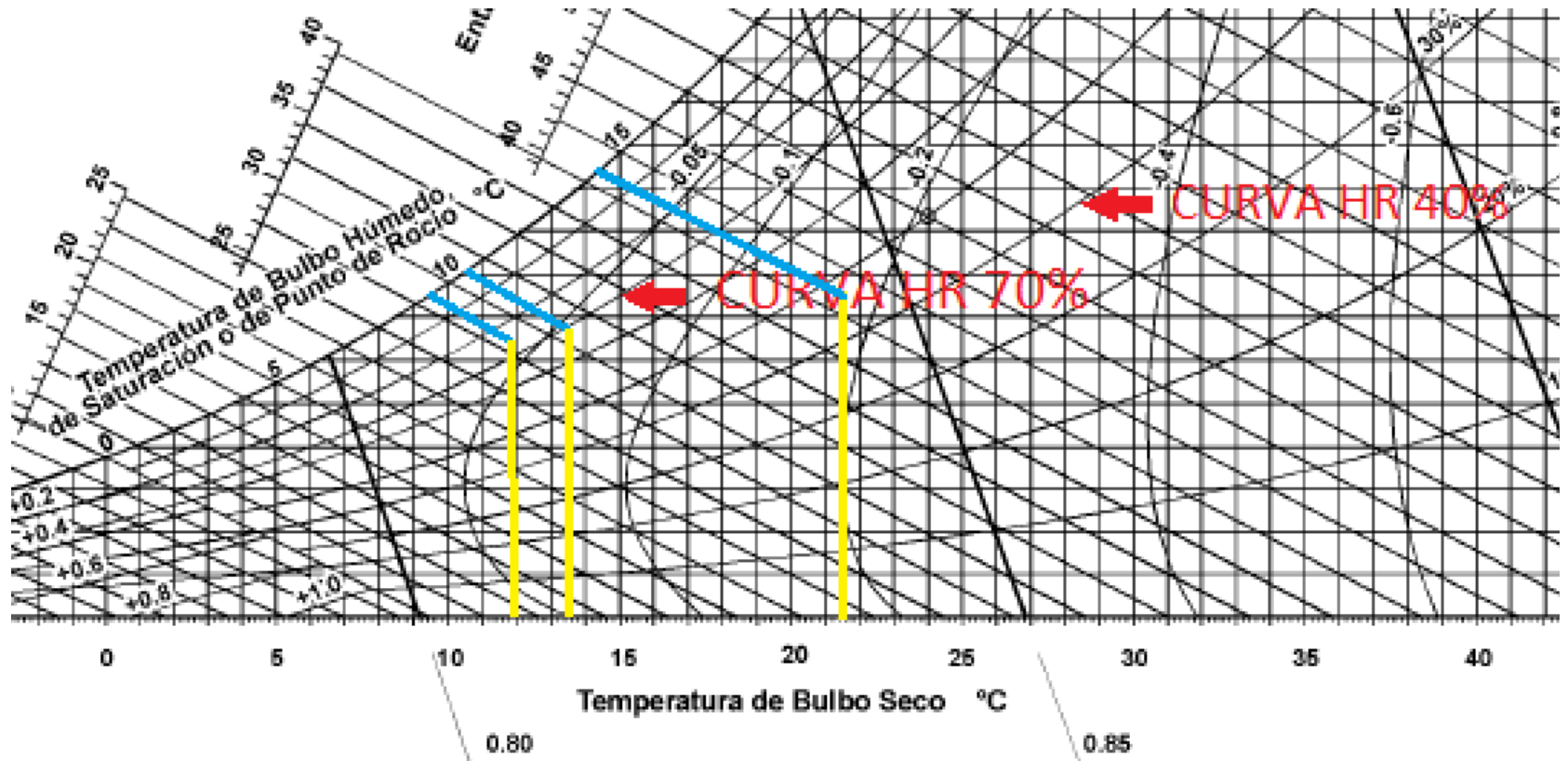
H. R.



Curvas de humedad relativa (DHT11)

Se registró una H. R. máxima de 77.63% a las 02:00:00 hrs y una H. R. mínima de 32.29% a las 14:00:00 hrs.

Carta psicrométrica



Tablas resumen

Temperatura	B. Húmedo	B. Seco	Hora	Diferencia
Tmáx	17.12 °C	26.3 °C	14:00:00	9.18 °C
Tmín	08.65 °C	10.4 °C	2:00:00	1.75 °C

Humedad relativa		Hora
H. R. máx	77.63 %	2:00:00
H. R. mín	32.29 %	14:00:00

Principio de Ockham

Cuando se ofrecen dos o más explicaciones de un fenómeno, es preferible la explicación completa más simple; es decir no deben multiplicarse las entidades sin necesidad.



**Agradecemos su
atención**