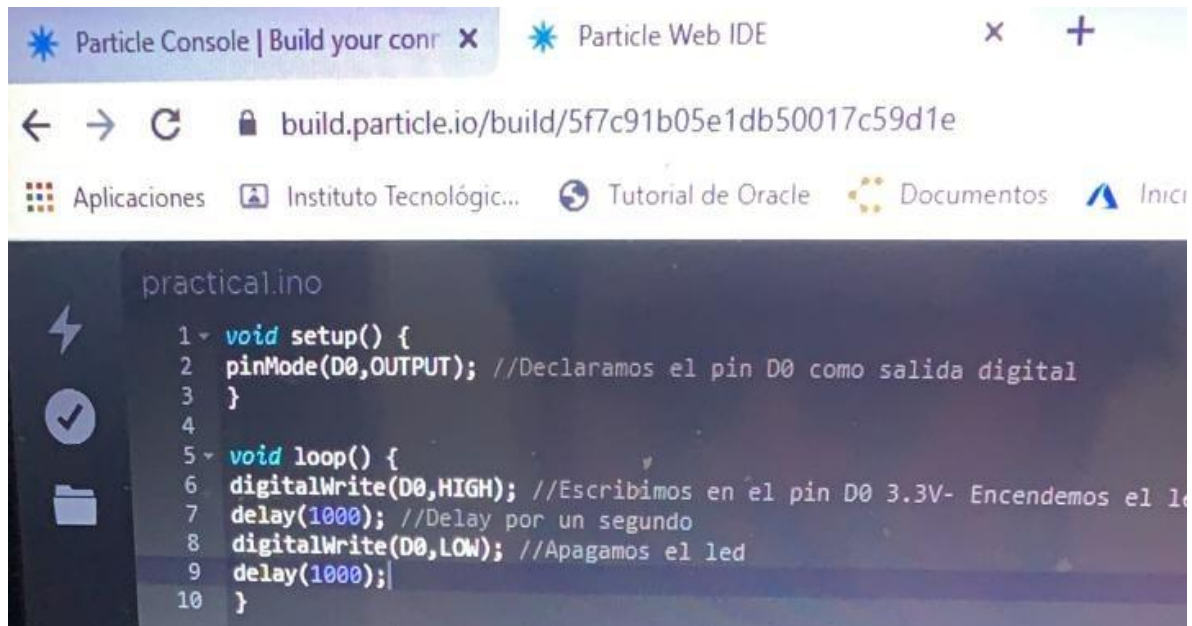


## Practica 1

Comenzando el curso de sistemas embebidos arrancamos con los fundamentos a lo que se le conoce como el hola mundo del dispositivo particle el cual ocupamos del mismo, una tabla popularmente conocida como proto, cables de mínima capacidad de volts y un led. En donde la practica se concentra en dar como resultado un led parpadeante cada segundo (el cual se conoce como delay o retraso). A continuación, se muestra el código para hacer lo anterior mencionado posible



The screenshot shows the Particle Web IDE interface. The browser tab is titled 'Particle Console | Build your contr' and the address bar shows 'build.particle.io/build/5f7c91b05e1db50017c59d1e'. The code editor displays the following C++ code for a file named 'practical.ino':

```
1- void setup() {
2-   pinMode(D0,OUTPUT); //Declaramos el pin D0 como salida digital
3- }
4-
5- void loop() {
6-   digitalWrite(D0,HIGH); //Escribimos en el pin D0 3.3V- Encendemos el led
7-   delay(1000); //Delay por un segundo
8-   digitalWrite(D0,LOW); //Apagamos el led
9-   delay(1000);
10- }
```

## Practica 2

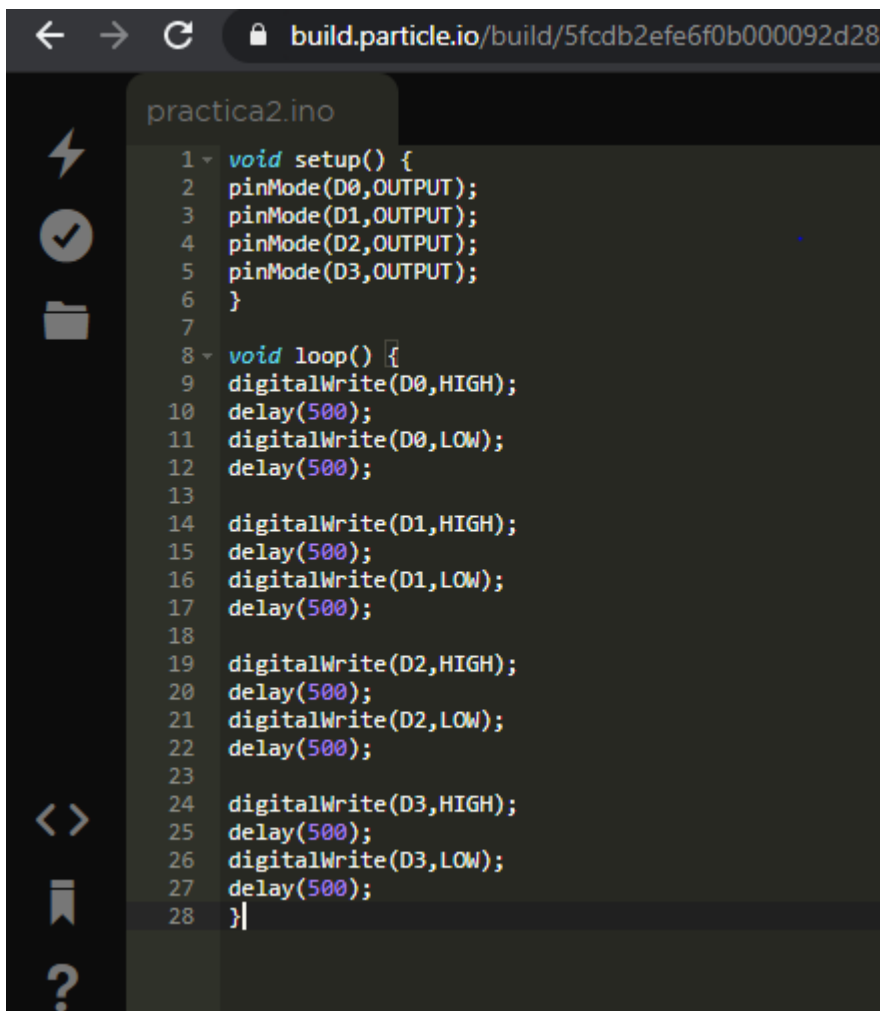
En esta práctica se hizo el llamado a cuatro pines de salda en donde se utilizaron 4 leds para que se cumpliera un loop en donde mientras el led 1 se encontraba prendido el led 2 estaba apagado y cuando este encendiera el led 1 se apagara, explicando de una mejor forma, todos con un retresado de medio segundo:

Led 1,3 = encendidos

Led 2,4 = apagados

Led 1,3 = apagados

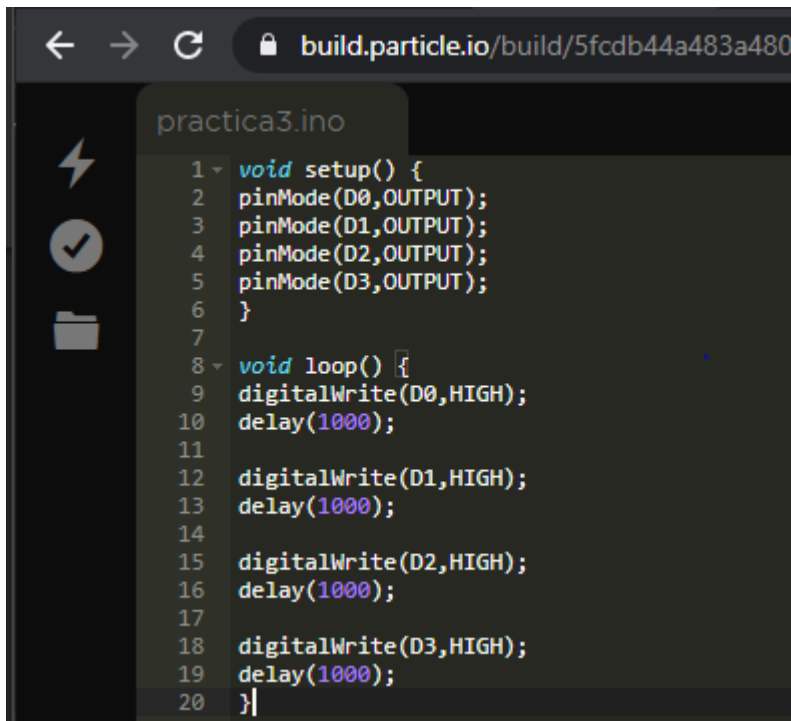
Led 2,4 = encendidos



```
practica2.ino
1 void setup() {
2   pinMode(D0,OUTPUT);
3   pinMode(D1,OUTPUT);
4   pinMode(D2,OUTPUT);
5   pinMode(D3,OUTPUT);
6 }
7
8 void loop() {
9   digitalWrite(D0,HIGH);
10  delay(500);
11  digitalWrite(D0,LOW);
12  delay(500);
13
14  digitalWrite(D1,HIGH);
15  delay(500);
16  digitalWrite(D1,LOW);
17  delay(500);
18
19  digitalWrite(D2,HIGH);
20  delay(500);
21  digitalWrite(D2,LOW);
22  delay(500);
23
24  digitalWrite(D3,HIGH);
25  delay(500);
26  digitalWrite(D3,LOW);
27  delay(500);
28 }
```

### Practica 3

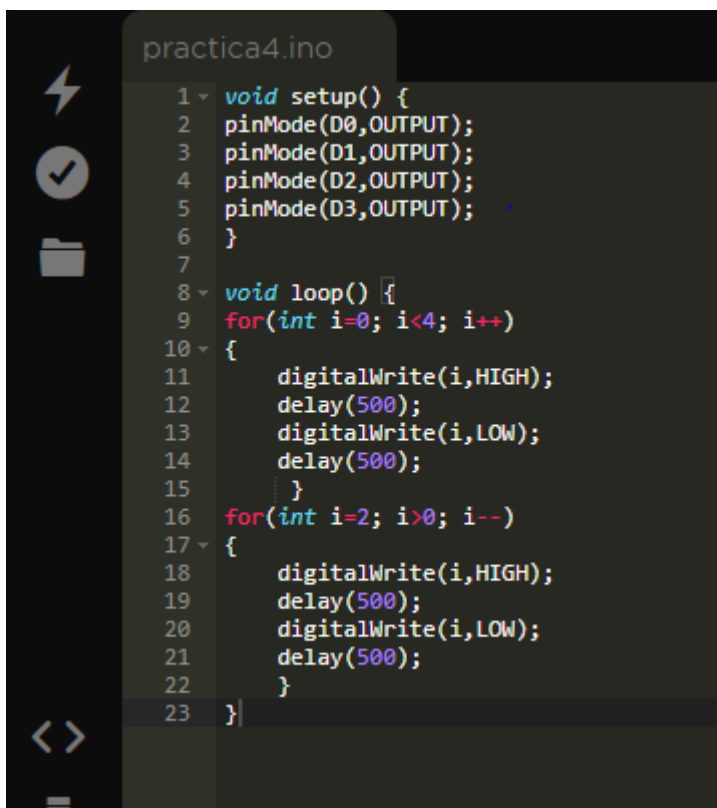
Ahora en esta ocasión el fin a cumplir tratar de hacer uso de 4 leds declarando 4 entradas de salida en los pines (recordemos que tenemos hasta 6 ), haciendo uso nuevamente del loop para que encienda del Led 1 hasta el Led 4 (claramente pasando por Led 2 y 3) en donde al pasar de un segundo gracias al retraso se encenderá un led , seguido del otro y hasta llegar al led final, todos con un segundos de delay(retraso), a continuación se muestra el código para lograr o mencionado.



```
1 void setup() {  
2   pinMode(D0,OUTPUT);  
3   pinMode(D1,OUTPUT);  
4   pinMode(D2,OUTPUT);  
5   pinMode(D3,OUTPUT);  
6 }  
7  
8 void loop() {  
9   digitalWrite(D0,HIGH);  
10  delay(1000);  
11  
12  digitalWrite(D1,HIGH);  
13  delay(1000);  
14  
15  digitalWrite(D2,HIGH);  
16  delay(1000);  
17  
18  digitalWrite(D3,HIGH);  
19  delay(1000);  
20 }
```

## Practica 4

Ahora hemos avanzado considerablemente entendiendo el uso del loop por lo que el siguiente nivel será hacer dentro del loop la implementación de un ciclo for en donde nuevamente se emplearan 4 leds y el objetivo de este es cada vez que el led 1 encienda el siguiente a el se apague con un delay de medio segundo y posterior a eso este lo mantendrá apagado hasta que pase medio segundo para volver a encender, como resultado un led estará encendido, led apagado, led encendido, led apagado (cambiando su estado cada medio segundo), a continuación se muestra el código empleado.



```
practica4.ino
1 void setup() {
2   pinMode(D0,OUTPUT);
3   pinMode(D1,OUTPUT);
4   pinMode(D2,OUTPUT);
5   pinMode(D3,OUTPUT);
6 }
7
8 void loop() {
9   for(int i=0; i<4; i++)
10  {
11    digitalWrite(i,HIGH);
12    delay(500);
13    digitalWrite(i,LOW);
14    delay(500);
15  }
16   for(int i=2; i>0; i--)
17  {
18    digitalWrite(i,HIGH);
19    delay(500);
20    digitalWrite(i,LOW);
21    delay(500);
22  }
23 }
```

## Practica 5.

Como podemos observar se está empleando el uso de una variable de tipo entero la cual su función es el tiempo con un valor de 500 milisegundos en donde nuevamente seguimos con 4 leds pero ahora la diferencia es el uso de dos for dentro del loop, dichos ciclos cumplen con la función de que una vez que del led 1 al led 4 se enciendan(todos estos con un retraso de medio segundo), dichos leds volverán apagarse con un retraso de medio segundo pero ahora comenzando por el led 4, pasando por el 3 ,2 y terminando en el led 1

practica5.ino

```
1  int t = 500;
2
3  void setup() {
4      pinMode(D0, OUTPUT);
5      pinMode(D1, OUTPUT);
6      pinMode(D2, OUTPUT);
7      pinMode(D3, OUTPUT);
8  }
9
10 void loop() {
11     for (int i=0; i<4; i++){
12         digitalWrite(i, HIGH);
13         delay(t);
14     }
15
16     for (int i=3; i>=0; i--){
17         digitalWrite(i, LOW);
18         delay(t);
19     }
20 }
21
22 }
```

## Practica 6

Ahora con lo visto en clase se comprende que los leds pueden ser configurados de manera que muestren diferentes intensidades es por eso por lo que de 0 a 255 hiciste diferente luminosidad entre estos, una muestra clara es usar 4 leds y mostrar la diferente candencia de estos como se muestra en el código haciendo uso del atributo 'analogWrite' (Salida, Intensidad).

```
practica6.ino
1 void setup() {
2   pinMode(D0,OUTPUT);
3   pinMode(D1,OUTPUT);
4   pinMode(D2,OUTPUT);
5   pinMode(D3,OUTPUT);
6 }
7
8 void loop() {
9   analogWrite(D0,255);
10
11   analogWrite(D1,127);
12
13   analogWrite(D2,51);
14
15   analogWrite(D3,12.75);
16 }
```

## Practica 7

Continuamos por el mismo camino de la intensidad en los leds, ahora solamente haremos uso de un Led para mostrar como podemos hacer el cambio de la intensidad cada 50 milisegundos (usando el delay) para hasta que en dicho ciclo se cumpla la condición la cual es si empezamos de 0 cada vez aumentaremos la intensidad hasta llegar a su máximo estado. Todo esto es posible con la implementación del código que se muestra a continuación.

practica7.ino

```
1 void setup() {  
2   pinMode(D0,OUTPUT);  
3 }  
4  
5 void loop() {  
6   for(int i = 0; i<256; i++){  
7     analogWrite(D0,i);  
8     delay(50);  
9   }  
10 }
```

## Practica 8

Ahora en la practica se busca tener el mismo resultado que la anterior agregándole que se debe regresar de su punto máximo a su intensidad mas baja como se muestra a continuación, aquí solamente se agrego un ciclo que su finalidad es la contraria al primero en donde una vez que llegue a su punto máximo se ira restando intensidad.

```
practica8.ino
1 void setup() {
2   pinMode(D0,OUTPUT);
3 }
4
5 void loop() {
6   for(int i = 0; i<256; i++){
7     analogWrite(D0,i);
8     delay(50);
9   }
10  for(int i = 255; i>=0; i--){
11    analogWrite(D0,i);
12    delay(50);
13  }
14 }
```



## Practica 9

Ahora en dicha practica mostrada estamos implementado el uso de un ciclo en donde hacemos el uso de 4 leds en donde del 1 al 4 enciendes con retrasos ósea delay según lo configuremos y una vez se llegue al cuatro regresaremos del led numero 4 al numero 1, para que todo esto sea posible se usa el siguiente código mostrado en seguida:

pract9.ino

```
1  int t = 500;
2
3  void setup() {
4    pinMode(D0, OUTPUT);
5    pinMode(D1, OUTPUT);
6    pinMode(D2, OUTPUT);
7    pinMode(D3, OUTPUT);
8  }
9
10 void loop() {
11   for (int i=0; i<=3; i++){
12     digitalWrite(i, HIGH);
13     delay(t);
14   }
15
16   for (int i=3; i>=0; i--){
17     digitalWrite(i, LOW);
18     delay(t);
19   }
20
21
22
23   for(int i = 0; i<256; i++){
24     analogWrite(D0,i);
25     delay(5);
26     analogWrite(D1,i);
27     delay(5);
28     analogWrite(D2,i);
29     delay(5);
30     analogWrite(D3,i);
31     delay(5);
32   }
33   for(int i = 255; i>=0; i--){
34     analogWrite(D0,i);
35     delay(5);
36     analogWrite(D1,i);
37     delay(5);
38     analogWrite(D2,i);
39     delay(5);
40     analogWrite(D3,i);
41     delay(5);
42   }
```

Ready.

pract9.ino

```
38     analogWrite(D2,i);
39     delay(5);
40     analogWrite(D3,i);
41     delay(5);
42 }
43 digitalWrite(D0,HIGH);
44 delay(500);
45 digitalWrite(D0,LOW);
46 delay(500);
47
48 digitalWrite(D1,HIGH);
49 delay(500);
50 digitalWrite(D1,LOW);
51 delay(500);
52
53 digitalWrite(D2,HIGH);
54 delay(500);
55 digitalWrite(D2,LOW);
56 delay(500);
57
58 digitalWrite(D3,HIGH);
59 delay(500);
60 digitalWrite(D3,LOW);
61 delay(500);
62
63 digitalWrite(D2,HIGH);
64 delay(500);
65 digitalWrite(D2,LOW);
66 delay(500);
67
68 digitalWrite(D1,HIGH);
69 delay(500);
70 digitalWrite(D1,LOW);
71 delay(500);
72
73
74 digitalWrite(D0,HIGH);
75 delay(500);
76 digitalWrite(D0,LOW);
77 delay(500);
78 }
```

Ready.

## Practica 10

Continuando con las practicas ahora se va a declarar una variable double la cual es igualada a 0 para usar A0 como entrada en donde se hace la lectura analógica y se monitorea la variable de dicha lectura, todo esto para incursionar dando un giro a nuevas funcionalidades, para hacer todo esto posible tenemos que escribir el siguiente código

```
pract10.ino
1  double lectura = 0; //Declaramos la variable como double y la igualamos a cero
2  void setup() {
3    pinMode(A0,INPUT); //Declaramos A0 como entrada
4    Serial.begin(9600); //Inicializamos el monitor serial
5  }
6
7  void loop() {
8    lectura = analogRead(A0); //Leemos la entrada analogica
9    Serial.println(lectura); //Monitoreamos la variable lectura
10   delay(1000);
11 }
```

## Practica 11

Ahora bien, hacemos unos ligeros cambios como se pueden observar en el código siguiente mostrado, basados en la estructura similar de lo anterior:

```
pract11.ino
1  double lectura = 0;
2
3  void setup() {
4    pinMode(A0,INPUT);
5    Serial.begin(9600);
6  }
7
8  void loop() {
9    lectura = analogRead(A0);
10   Serial.println(lectura);
11   delay(1000);
12 }
```

## Practica 12

Con esta práctica se busca la implementación de dos variables, lectura y voltaje en donde debemos inicializar las variables desde 0 y para esto usamos una formula la cual esta explicita en el código como "voltaje = (lectura \* 3.3) / 4095.0; para así poder imprimir el voltaje con un retraso de un segundo como se muestra en el siguiente código para que todo esto pueda ser posible

```
pract12.ino
1  double lectura = 0;
2  float voltaje = 0;
3
4  void setup() {
5    pinMode(A0, INPUT);
6    Serial.begin(9600);
7  }
8
9  void loop() {
10   lectura = analogRead(A0);
11   voltaje = (lectura * 3.3) / 4095.0;
12   Serial.println(lectura);
13   Serial.print(voltaje);
14   Serial.println(" V");
15   delay(1000);
16 }
```

## Practica 13

Haciendo la implementación del sensor de humedad el cual también se puede usar para medir la temperatura actual dependiendo de como se configure, se implementara el siguiente código donde se usaran 4 variables para poder implementarlas posteriormente y ser mostrada la temperatura usando fórmulas que harán posible este resultado como se muestra en la pantalla el trabajo del voltaje (formula ya usada) con los volts y con un retraso de 1 segundos en la captura y actualización de la temperatura actual mostrada en grados centígrados

```
pract13.ino
1  double lectura = 0;
2  double voltaje = 0;
3  double mv = 0.0;
4  double grados = 0.0;
5
6  void setup() {
7    pinMode(A0, INPUT);
8    Serial.begin(9600);
9  }
10
11 void loop() {
12   lectura = analogRead(A0);
13   voltaje = (lectura * 3.3)/4095.0;
14   mv = voltaje * 1000.0;
15   grados = mv / 10.0;
16   Serial.println(lectura);
17   Serial.print(voltaje);
18   Serial.println(" volts");
19   Serial.print(grados);
20   Serial.println(" °C");
21   delay(1000);
22 }
```

## Practica 14

Mejorando la practica anterior ahora vemos como se va a trabajar con un foco el cual encenderá y apagará cada que la condicional lo indique en este caso el foco encenderá haciendo uso del relevador cuando este a mas de 20 grados y se apagará cuando su temperatura sea leída menos de los 20 grados centígrados, haciendo uso de cuatro variables inicializadas en 0 para declarar las 4 entradas, entre ellas entradas y salidas. Dicha practica se puede comprobar con el uso del siguiente código y ayuda de un par de hielos o paletas y un encendedor o fosforo para aumentar y bajar la temperatura de forma rápida:

```
pract14.ino
1  int lectura = 0;
2  double volt = 0.0;
3  double mini = 0;
4  double grad = 0.0;
5
6  void setup() {
7      pinMode(A0,INPUT);
8      pinMode(D0,OUTPUT);
9      pinMode(D1,OUTPUT);
10     pinMode(D2,OUTPUT);
11     Serial.begin(9600);
12 }
13
14 void loop() {
15     lectura = analogRead(A0);
16     volt = (lectura*3.3)/4095.0;
17     mini = volt*1000.0;
18     grad = mini/10;
19     Serial.println("La temperatura es: ");
20     Serial.print(grad);
21     Serial.println("°C");
22     delay(1000);
23
24     if(grad <= 20){
25         digitalWrite(D1,LOW);
26         digitalWrite(D2,LOW);
27         digitalWrite(D0,HIGH);
28         delay(500);
29         digitalWrite(D0,LOW);
30         delay(500);
31     }
32
33     else if(grad > 20 && grad <= 30){
34         digitalWrite(D0,LOW);
35         digitalWrite(D2,LOW);
36         digitalWrite(D1,HIGH);
37         delay(500);
38         digitalWrite(D1,LOW);
39         delay(500);
40     }
41 }
```

Ready.

```
41
42 ▾ else {
43     digitalWrite(D0,LOW);
44     digitalWrite(D1,LOW);
45     digitalWrite(D2,HIGH);
46     delay(500);
47     digitalWrite(D2,LOW);
48     delay(500);
49 }
50 }
```

Ready.

## Practica 15

Con ayuda del archivo html previamente facilitado por el ingeniero hacemos la implementación del código en donde cambiamos de estado el encendido y apagado del foco con un retraso de 1 segundo, todo esto posible con el siguiente código:

pract15.ino

```
1 ▾ void setup() {
2     pinMode(D2,OUTPUT);
3 }
4
5 ▾ void loop() {
6     digitalWrite(D2,HIGH);
7     delay(1000);
8     digitalWrite(D2,LOW);
9     delay(1000);
10 }
```

## Practica 16

Con esta práctica se busca encender y apagar el foco mostrando con el apoyo del html dicha temperatura al igual que el apoyo de algunos leds en donde con el siguiente código se podrá ver lo siguiente:

```
pract16.ino
1  int lectura = 0;
2  double volt = 0.0;
3  double mini = 0;
4  double grad = 0.0;
5
6  void setup() {
7    pinMode(A0,OUTPUT);
8    pinMode(D2,OUTPUT);
9  }
10
11 void loop() {
12   lectura = analogRead(A0);
13   volt = (lectura*3.3)/4095.0;
14   mini = volt*1000.0;
15   grad = mini/10;
16   Serial.println("La temperatura es: ");
17   Serial.print(grad);
18   Serial.println("°C");
19   delay(1000);
20
21   if(grad > 30){
22     digitalWrite(D2,HIGH);
23     delay(10000);
24     digitalWrite(D2,LOW);
25     delay(3000);
26   }
27 }
```



## Practica 17

Ahora continuamos con la practica en donde hacemos la implementación de cuatro variables las cuales son lectura, voltaje, mvolts, gradosC , dichas variables comienzan con una inicialización de 0 para que posteriormente el dispositivo particle pueda trabajar con ellas usando las formulas que se muestran en el void loop

```
pract17.ino
1  double lectura = 0;
2  double voltaje = 0;
3  double mvolts = 0;
4  double gradosC = 0;
5
6  void setup() {
7    pinMode(A0, INPUT);
8    Particle.variable("lectura", &lectura, DOUBLE);
9    Particle.variable("voltaje", &voltaje, DOUBLE);
10   Particle.variable("mvolts", &mvolts, DOUBLE);
11   Particle.variable("gradosC", &gradosC, DOUBLE);
12 }
13
14 void loop() {
15   lectura = analogRead(A0);
16   voltaje = (lectura * 3.3)/4095.0;
17   mvolts = voltaje * 1000.0;
18   gradosC = (mvolts * 1.0)/10.0;
19 }
```

## Practica 19

Con los conocimientos ya adquiridos buscamos mostrar de forma grafica con apoyo de Html y javascript, archivos facilitados por el ingeniero los cuales tuvieron unas ligeras modificaciones para hacer la implementación favorable a esta practica y con eso obtener los gratos en formato de centígrados y Fahrenheit , como se muestra a continuación, así mismo tuvimos que tener en cuenta en caso de que no tuviera registro de ninguna de las temperaturas, nos deberá de enviar un mensaje como error, el cual siempre tendrá un retraso de un segundo para obtener los datos ya mencionados:

```
pract19.ino
1 // This #include statement was automatically added by the Particle IDE.
2 #include <Adafruit_DHT_Particle.h>
3
4
5 float gradosC = 0;
6 float gradosF = 0;
7 float humedad = 0;
8
9 #define DHTPIN D6
10 #define DHTTYPE DHT11
11
12 DHT dht(DHTPIN,DHTTYPE);
13
14
15
16 void setup() {
17   pinMode(D6,INPUT);
18   dht.begin();
19   Serial.begin(9600);
20 }
21
22 void loop() {
23   gradosC = dht.getTempCelcius();
24   gradosF = dht.getTempFahrenheit();
25   humedad = dht.getHumidity();
26
27   if(isnan(gradosC) || isnan(gradosF) || isnan(humedad)){
28     Serial.println("Error de lectura");
29     return;
30   }
31   Serial.print(gradosC);
32   Serial.println("");
33
34   Serial.print(gradosF);
35   Serial.println("F");
36
37   Serial.print("El porcentaje de humedad es:");
38   Serial.println(humedad);
39   delay(1000);
40 }
```

Ready.

## Practica 20

Ahora mejorando la practica anterior se busca usar el uso de los siguientes componentes como el relevador, el sensor de humedad, foco, roset, respectivas conexiones todos controlados por el Arduino en donde el resultado final será mostrado un html donde podremos observar la temperatura en grados centígrados y farenheit para que cuando dicha temperatura pase de los 20 grados o lo equivalente a los grados farenheit pueda encender el foco y de manera contraria baje la temperatura el foco se apagara

pract20.ino

```
1 // This #include statement was automatically added by the Particle IDE.
2 #include <Adafruit_DHT.h>
3
4 float gradosC = 0;
5 float gradosF = 0;
6 float humedad = 0;
7
8 #define DHTPIN D6
9 #define DHTTYPE DHT11
10
11 DHT dht(DHTPIN,DHTTYPE);
12
13
14
15 void setup() {
16   pinMode(D6,INPUT);
17   pinMode(D2,OUTPUT);
18   dht.begin();
19   Serial.begin(9600);
20 }
21
22 void loop() {
23   gradosC = dht.getTempCelcius();
24   gradosF = dht.getTempFarenheit();
25   humedad = dht.getHumidity();
26
27   if(isnan(gradosC) || isnan(gradosF) || isnan(humedad)){
28     Serial.println("Error de lectura");
29     return;
30   }
31   Serial.print(gradosC);
32   Serial.println("");
33
34   Serial.print(gradosF);
35   Serial.println("F");
36
37   Serial.print("El porcentaje de humedad es:");
38   Serial.println(humedad);
39   delay(1000);
40
41   if(grad > 20){
```

Ready.

```
40
41 ▾ if(grad > 20){
42     digitalWrite(D2,HIGH);
43     delay(1000);
44     else
45     digitalWrite(D2,LOW);
46     delay(1000);
47 }
48 }|
```

Ready.

## Practica 21

Con dicha práctica se hará el uso de un led el cual encenderá cuando se presione por medio de la terminal a cambiar su estado de on a off, la acción se puede ejecutar gracias al código mostrado

```
pract21.ino
1 ▾ void setup() {
2     pinMode(D6, OUTPUT);
3     Particle.function("led", ledfunction);
4
5
6
7 }
8
9 ▾ void loop() {
10
11 }
12
13 ▾ int ledfunction(String parametro){
14 ▾     if(parametro == "on"){
15         digitalWrite(D6, HIGH);
16         return 1;|
17     }
18 ▾     else if(parametro == "off"){
19         digitalWrite(D6, LOW);
20         return 0;
21     }
22 ▾     else{
23         return -1;
24     }
25
26 }
```