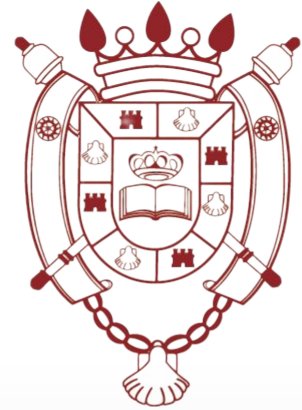


Internet de las Cosas con ESP32

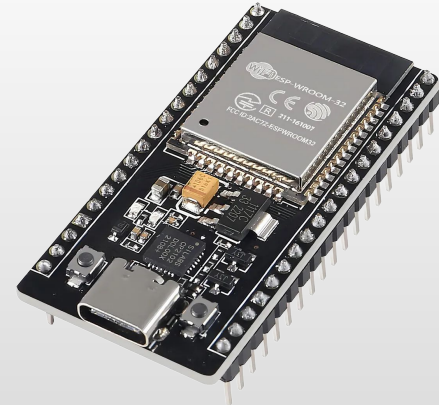
Sensores analógicos, digitales
y Visualización OLED

Ing. Fernando Raúl Vera Suasnívar

Universidad Nacional de Santiago del Estero - 2025



3



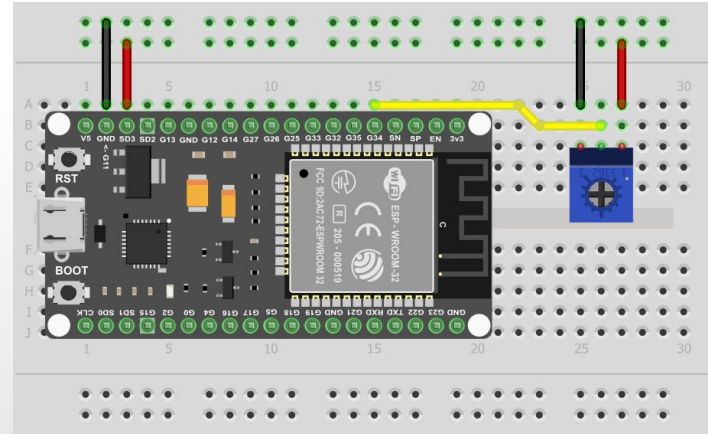
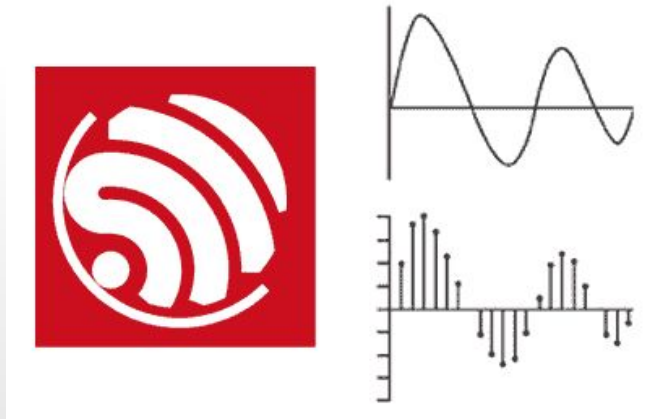
Objetivos

- 1) Usar **ADC** de 12 bits en ESP32
- 2) **Filtros digitales**: móvil y exponencial
- 3) Lectura de **NTC** (analógico) y **DS18B20** (digital)
- 4) Visualizar en **OLED**

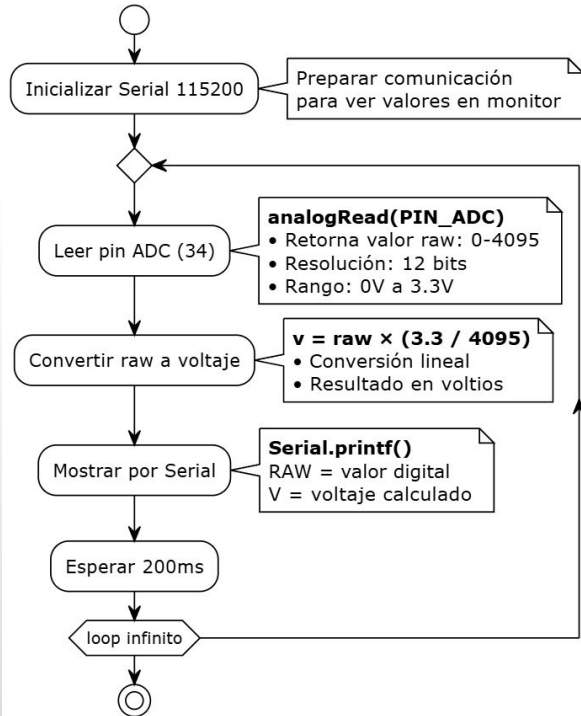
"Medir es conocer."

– Lord Kelvin

1.1) Usar ADC de 12 bits en ESP32



1.2) Usar ADC de 12 bits en ESP32



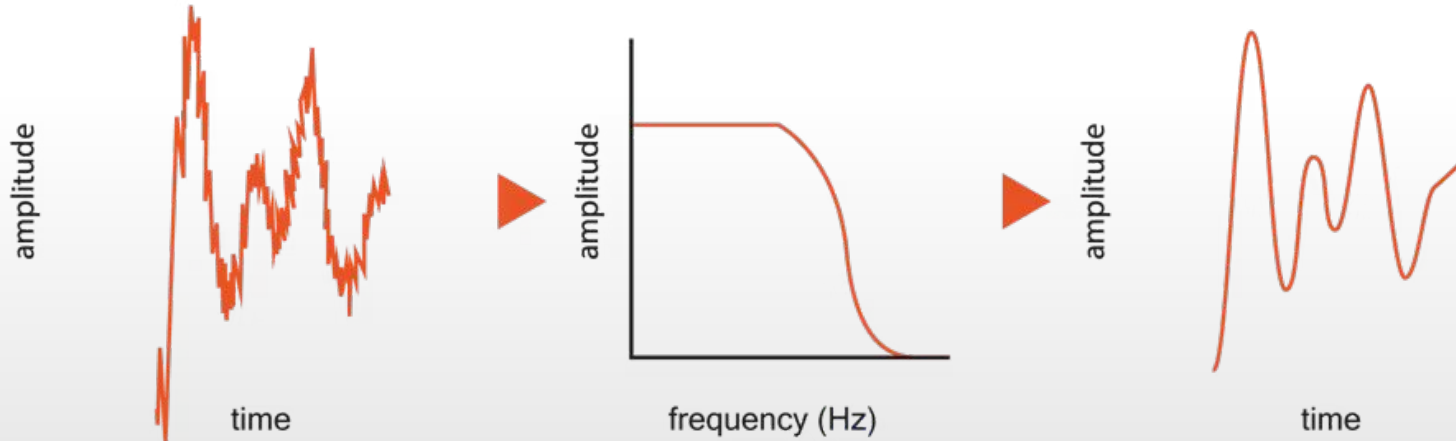
```
#include <Arduino.h>

const int PIN_ADC = 34;

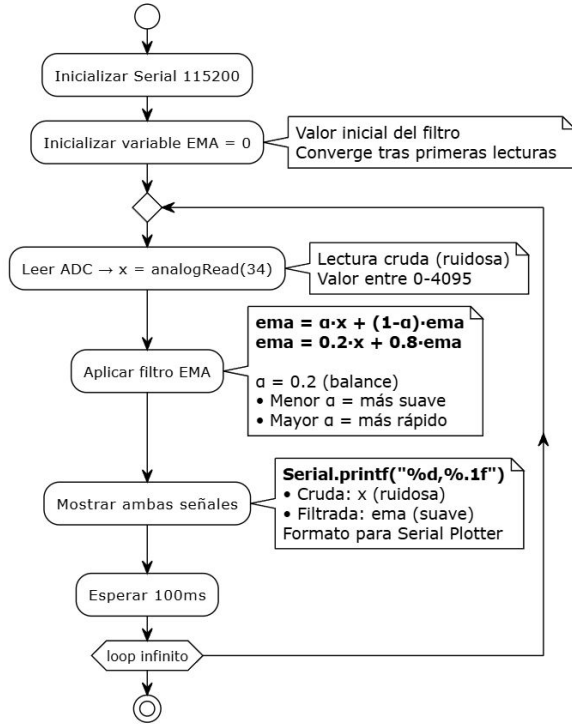
void setup() {
    Serial.begin(115200);
}

void loop() {
    int raw = analogRead(PIN_ADC);
    float v = raw * (3.3f / 4095.0f);
    Serial.printf("RAW=%d  V=%.2f", raw, v);
    Serial.println();
    delay(200);
}
```

2.1) Filtros digitales: móvil y exponencial



2.2) Filtros digitales: móvil y exponencial



```
#include <Arduino.h>
```

```
const int PIN_ADC = 34;
```

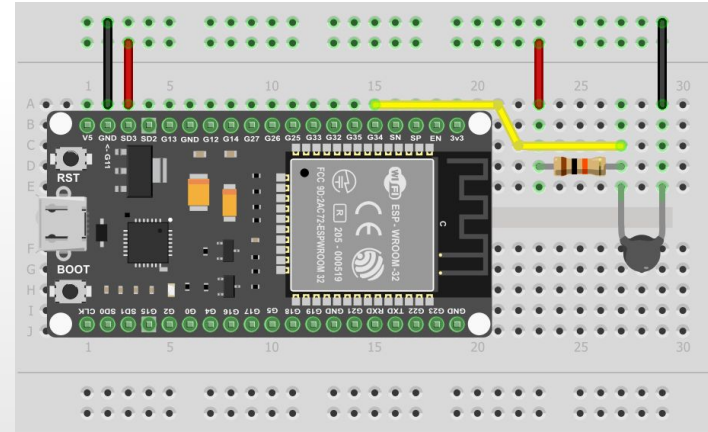
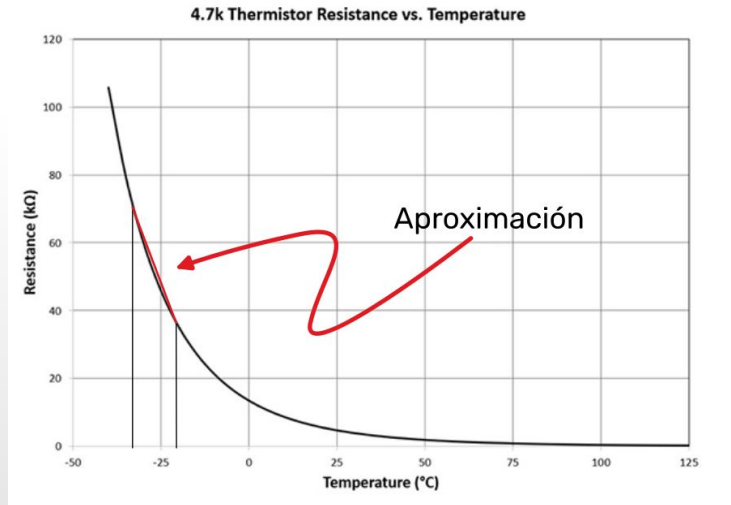
```
float ema = 0;
```

```
void setup() {  
    Serial.begin(115200);  
}
```

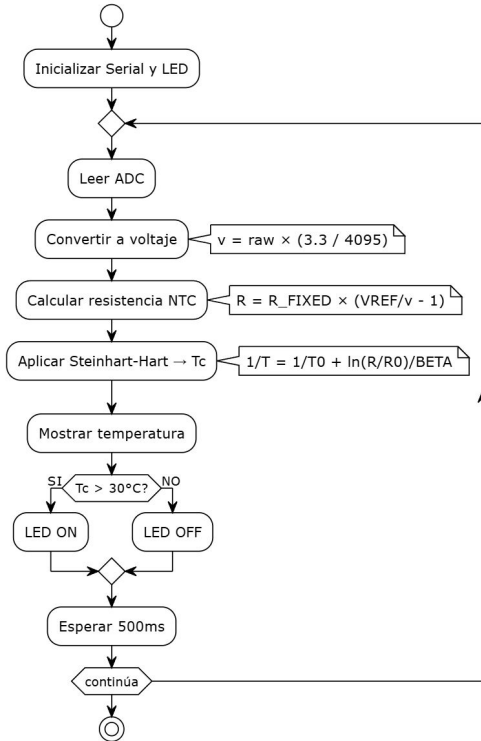
```
void loop() {  
    int x = analogRead(PIN_ADC);  
    ema = 0.2 * x + 0.8 * ema;  
    Serial.printf("%d,%d,%1f", x, ema);  
    Serial.println();  
    delay(100);  
}
```



3.1) Lectura de NTC (analógico)



3.2) Lectura de NTC (analógico)



```
#include <Arduino.h>
#include <math.h>
```

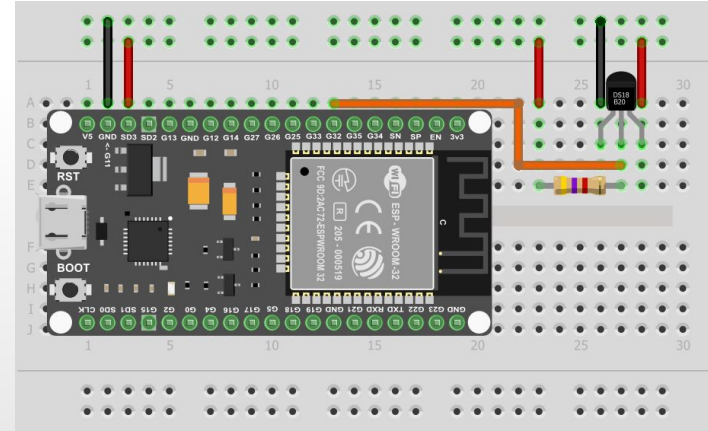
```
const int PIN_ADC = 34; const int PIN_LED = 2;
const float VREF = 3.3, R_FIXED = 10000, R0 = 10000, CT0 = 298.15, BETA = 3950;
```

```
void setup() {
  Serial.begin(115200);
  pinMode(PIN_LED, OUTPUT);
}
```

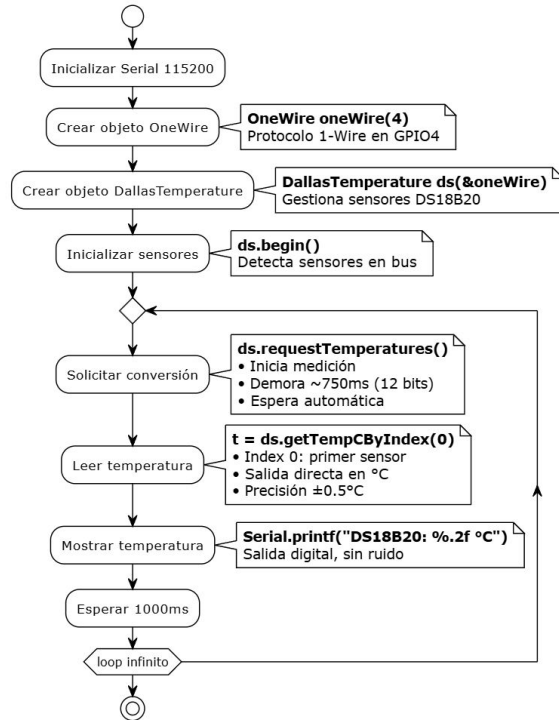
```
void loop() {
  int raw = analogRead(PIN_ADC);
  float v = raw * (VREF / 4095.0);
  float R = R_FIXED * (VREF / v - 1);
  float Tc = (1 / (1 / CT0 + log(R / R0) / BETA)) - 273.15;
  Serial.printf("T=%.2f °C", Tc);
  Serial.println();
  digitalWrite(PIN_LED, Tc > 30 ? HIGH : LOW);
  delay(500);
}
```



3.3) Lectura de DS18B20 (digital)



3.4) Lectura de DS18B20 (digital)



```
#include <Arduino.h>
#include <OneWire.h>
#include <DallasTemperature.h>
```

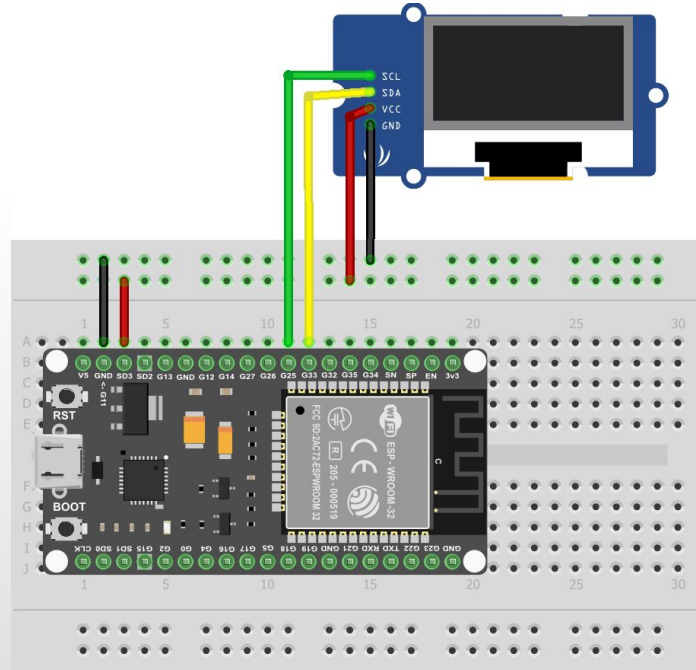
```
OneWire oneWire(4);
DallasTemperature ds(&oneWire);
```

```
void setup(){
    Serial.begin(115200);
    ds.begin();
}
```

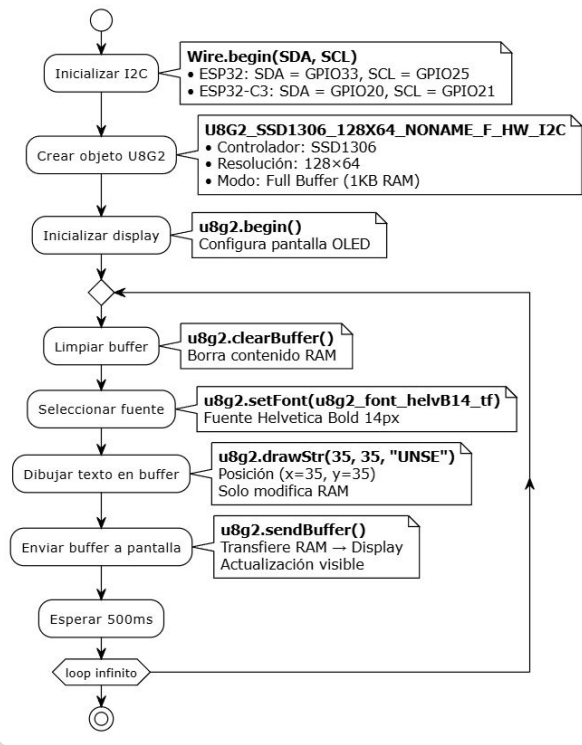
```
void loop(){
    ds.requestTemperatures();
    float t = ds.getTempCByIndex(0);
    Serial.printf("DS18B20: %.2f °C", t);
    Serial.println();
    delay(1000);
}
```



4.1) Visualizar en OLED



4.2) Visualizar en OLED

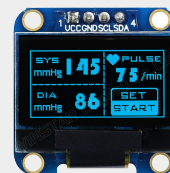


```
#include <Arduino.h>
#include <U8g2lib.h>
#include <Wire.h>
```

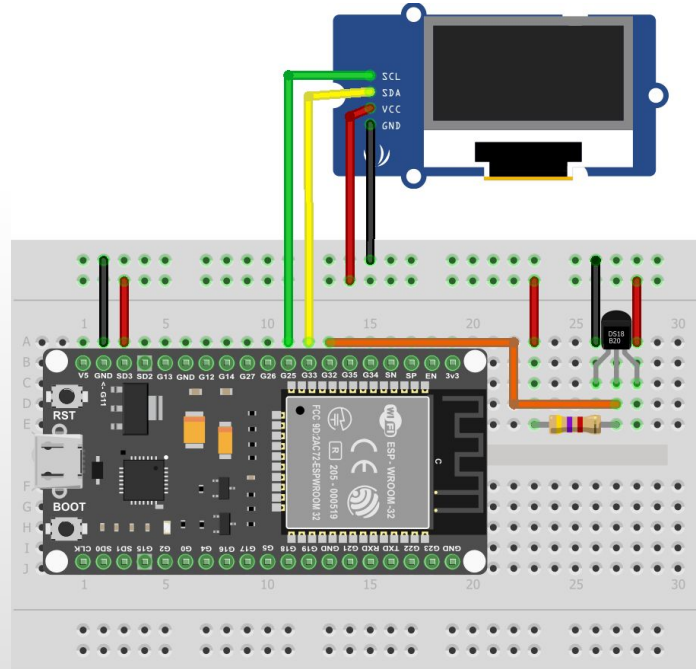
```
U8G2_SSD1306_128X64_NONAME_F_HW_I2C u8g2
    (U8G2_R0, U8X8_PIN_NONE);
```

```
void setup() {
    Wire.begin(33, 25);
    u8g2.begin();
}
```

```
void loop() {
    u8g2.clearBuffer();
    u8g2.setFont(u8g2_font_helvB14_tf);
    u8g2.drawStr(35, 35, "UNSE");
    u8g2.sendBuffer();
    delay(500);
}
```



4.3) Visualizar DS18B20 en OLED



¡Hora de practicar!

Poné a prueba lo aprendido

Lectura de AD con NTC + DS18B20 + OLED

Construí tu propio código paso a paso

Experimentá, probá y no tengas miedo de equivocarte

"Lo que escucho lo olvido, lo que veo lo recuerdo, lo que hago lo aprendo."

— Confucio