

```

// Variables para calculos
int pin = 8;
unsigned long duration;
unsigned long starttime;
unsigned long samptime_ms = 30000;//sampe 30s ;
unsigned long lowpulseoccupancy = 0;
float ratio = 0;
float concentration = 0;

long previousMillis = 0;
long interval = 1000;

// Cliente WiFly
#include <WiFlyHQ.h>

#include <SoftwareSerial.h>
SoftwareSerial wifiSerial(2,3);

WiFly wifly;
String data;

/* Datos de red */
const char mySSID[] = "mired";
const char myPassword[] = "miclave";

const char site[] = "miservidor.com"; // o el URL de la aplicacion de Parse.com

void terminal();

void setup() {
  // Grove Shield
  Serial.begin(9600);
  pinMode(8,INPUT);
  starttime = millis();// Instante actual

  // WiFly Setup
  char buf[32];
  data = "";
  Serial.begin(9600);
  Serial.println("Starting");
  Serial.print("Free memory: ");
  Serial.println(wifly.getFreeMemory(),DEC);

  wifiSerial.begin(9600);
  if (!wifly.begin(&wifiSerial, &Serial)) {
    Serial.println("Failed to start wifly");
  }
}

```

```

    terminal();
}
/* Unirse a Red WiFi */
if (!wifly.isAssociated()) {
    /* Configurar WiFly para unirse */
    Serial.println("Joining network");
    wifly.setSSID(mySSID);
    wifly.setPassphrase(myPassword);
    wifly.enableDHCP();

    if (wifly.join()) {
        Serial.println("Joined wifi network");
    } else {
        Serial.println("Failed to join wifi network");
        terminal();
    }
} else {
    Serial.println("Already joined network");
}
Serial.print("MAC: ");
Serial.println(wifly.getMAC(buf, sizeof(buf)));
Serial.print("IP: ");
Serial.println(wifly.getIP(buf, sizeof(buf)));
Serial.print("Netmask: ");
Serial.println(wifly.getNetmask(buf, sizeof(buf)));
Serial.print("Gateway: ");
Serial.println(wifly.getGateway(buf, sizeof(buf)));

wifly.setDeviceID("Wifly-WebClient");
Serial.print("DeviceID: ");
Serial.println(wifly.getDeviceID(buf, sizeof(buf)));

if (wifly.isConnected()) {
    Serial.println("Old connection active. Closing");
    wifly.close();
}
}

void loop() {
    // Call for measurements & Check interval first
    unsigned long currentMillis = millis();
    if(currentMillis - previousMillis > interval) {
        // save the last time you sampled
        previousMillis = currentMillis;

        // Carry on with sample
        measureDust(); // Data se envia de measureDust()...
    }
}

```

```
}  
}
```

```
void measureDust() {  
  duration = pulseIn(pin, LOW);  
  lowpulseoccupancy = lowpulseoccupancy+duration;  
  
  if ((millis()-starttime) > sampletime_ms)//if the sampel time == 30s  
  {  
    ratio = lowpulseoccupancy/(sampletime_ms*10.0); // Integer percentage 0=>100  
    concentration = 1.1*pow(ratio,3)-3.8*pow(ratio,2)+520*ratio+0.62; // using spec  
sheet curve  
    Serial.print(lowpulseoccupancy);  
    Serial.print(",");  
    Serial.print(ratio);  
    Serial.print(",");  
    Serial.println(concentration);  
    lowpulseoccupancy = 0;  
    starttime = millis();  
    // Reportar a la nube  
    reportToCloud();  
  }  
}
```

```
void reportToCloud() {  
  if (wifly.available() > 0) {  
    char ch = wifly.read();  
    Serial.write(ch);  
    if (ch == '\n') {  
      /* add a carriage return */  
      Serial.write('\r');  
    }  
  }  
}
```

```
//..
```

```
if (wifly.open(site, 80)) {  
  Serial.print("Connected to ");  
  Serial.println(site);
```

```
// Set data to send  
static char outstr[15];
```

```
String dataString = dtostrf(concentration, 8, 2, outstr);  
data = String("name=Arduino&age=" + dataString);  
Serial.print(data);
```

```
wifly.println("POST /arduino/archivoDeProcesoDeDatos.php HTTP/1.0");
```

```
wifly.println("Host: www.miservidor.com"); // SERVER ADDRESS HERE TOO
wifly.println("Content-Type: application/x-www-form-urlencoded" );
wifly.print("Content-Length: ");
wifly.println(data.length());
    wifly.println();
wifly.print(data);

} else {
    Serial.println("Failed to connect");
}

if (Serial.available() > 0) {
    wifly.write(Serial.read());
}
}

void terminal()
{
    while (1) {
        if (wifly.available() > 0) {
            Serial.write(wifly.read());
        }
        if (Serial.available() > 0) {
            wifly.write(Serial.read());
        }
    }
}
}
```