

```

#define BLYNK_TEMPLATE_ID ""

#define BLYNK_TEMPLATE_NAME ""
#include <BlynkSimpleEsp32.h>

//Se incluye librería para el WiFi
#include <WiFi.h>

//Se define el código de Blynk
char auth[] ="";

//Se definen nombre de red y contraseña a la que se va a conectar el
controlador
const char* ssid = "";
const char* pass = "";

//Se inicializa el temporizador
BlynkTimer timer;

//Se conecta el sensor al pin 36
int sensor1 = 36;

BLYNK_CONNECTED() {
    Blynk.syncAll();
}

void setup() {

    //Temporizador para usar la función de envío de datos del sensor cada 1
    segundo
    timer.setInterval(1000L, DatosSensor);

    //Temporizador para verificar el estado de la conexión WiFi cada 1 segundo
    timer.setInterval(5*60*1000, ConexionWifi);

    Serial.begin(115200);

    //Iniciar conexión Wifi
    WiFi.begin(ssid, pass);

    //Mostrar asteriscos mientras se realiza la conexión
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print("*");
    }
}

```

```

//Se anuncia cuando ya se haya conectado a la red WiFi y a la nube
Serial.println("");
Serial.println("Se ha conectado a la red WiFi");

//Se inicia la conexión a la nube
Blynk.begin(auth, ssid, pass);

// Se establece el sensor como entrada
pinMode(sensor1, INPUT);

}

void ConexionWifi(){

//En caso de que el WiFi se haya desconectado volver a conectar
if (WiFi.status() != WL_CONNECTED) {

    ESP.restart();
}

}

//Se crea la función para enviar datos hacia la plataforma
void DatosSensor() {

//Se lee la información del sensor 1
int valor1=analogRead(sensor1);

//Se calibra el sensor siguiendo la ecuación de la curva de calibración
float pH=0.0231*valor1-61.077;

//Se envía el valor obtenido al pin virtual V0
Blynk.virtualWrite(V0,pH);

}

void loop() {

Blynk.run();
timer.run();

}

```