

CÓDIGO EN ARDUINO

```
#include <FirebaseESP8266.h>
```

```
#include <FirebaseESP8266HTTPClient.h>
```

```
#include <ESP8266WiFi.h>
```

```
#include <Wire.h> //SCL-> D1 SDA->D2
```

```
#include "RTClib.h"
```

```
#include <LiquidCrystal_I2C.h>
```

```
#include <DNSServer.h>
```

```
#include <ESP8266WebServer.h>
```

```
#include <WiFiManager.h>
```

```
#include "HX711.h"
```

```
#define FIREBASE_HOST "first-cb55e.firebaseio.com"
```

```
#define FIREBASE_AUTH "Y9CJQTignR0vu6FtpzA7hbNSVaecd0QtuPnEocNE"
```

```
RTC_DS1307 RTC;
```

```
HX711 scale1, scale2;
```

```
FirebaseData firebase;
```

```
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

```
int pesa1, pesa2, agua, dispenc, dispena, n_dia, r_dia, n_mes, t_mes, n_anno, d_anno, t_siglo = 6, anio, ms, dy, hr, mn, sg;
```

```
byte gotaf [] = {0x04, 0x0E, 0x0E, 0x1F, 0x1F, 0x1F, 0x0E, 0x00};
```

```
byte gotae [] = {0x04, 0x0A, 0x0A, 0x11, 0x11, 0x11, 0x0E, 0x1F};
```

```
byte tazaf [] = {0x0E, 0x1F, 0x1F, 0x1F, 0x1F, 0x1F, 0x0E, 0x00};
```

```
byte tazae [] = {0x0E, 0x1B, 0x11, 0x11, 0x11, 0x1B, 0x0E, 0x1F};
```

```
byte grams [] = {0x1C, 0x14, 0x1C, 0x04, 0x1C, 0x03, 0x02, 0x02};
```

```
byte wifis [] = {0x1F, 0x00, 0x1F, 0x00, 0x0E, 0x00, 0x04, 0x00};
```

```
String Time, newTime, Hora, STAgua, STComida, STTime, MManual, MtAgua, MtComida, fan, fms, fdy, fho,  
fmn, fsg;
```

```
const int DT1 = D6, SCK1 = D5, DT2 = D7, SCK2 = D8;
```

```
void setup() {
```

```
  pinMode(D0, INPUT);
```

```
  pinMode(A0, INPUT);
```

```
  pinMode(D3, OUTPUT);
```

```
  pinMode(D4, OUTPUT);
```

```
  digitalWrite(D3, LOW);
```

```
  digitalWrite(D4, LOW);
```

```
  WiFi.hostname("Comedor_ESP8266");
```

```
  Serial.begin(9600);
```

```
  Wire.begin();
```

```
  WiFiManager wifiManager;
```

```
  wifiManager.autoConnect("Dispensador");
```

```
  RTC.begin();
```

```
  //RTC.adjust(DateTime(F(__DATE__), F(__TIME__)));
```

```
  lcd.init();
```

```
  lcd.createChar(0, gotaf);
```

```
  lcd.createChar(1, gotae);
```

```
lcd.createChar(2, tazaf);
lcd.createChar(3, tazae);
lcd.createChar(4, grams);
lcd.createChar(5, wifis);
lcd.setBacklight(255);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Esperando Wi-Fi");
lcd.write(byte(5));
lcd.setCursor(0, 1);
while (WiFi.status() != WL_CONNECTED)
{
  lcd.print(".");
  delay(300);
}
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Conectado a:");
lcd.setCursor(0, 1);
//lcd.println(WIFI_SSID);
delay(500);
lcd.clear();
Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
Firebase.reconnectWiFi(true);
Serial.println("inicio");

scale1.begin(DT1, SCK1);
scale1.set_scale(359.7845);
```

```
scale1.tare(20);
```

```
scale2.begin(DT2, SCK2);
```

```
scale2.set_scale(256.536);
```

```
scale2.tare(20);
```

```
}
```

```
void loop() {
```

```
  Tiempo();
```

```
  VariablesOut();
```

```
  VariablesIn();
```

```
  Dispensacion();
```

```
  Manual();
```

```
  Alarmas();
```

```
}
```

```
void Manual() {
```

```
  if (MManual == "1")
```

```
  {
```

```
    if (MtAgua == "1")
```

```
    {
```

```
      digitalWrite(D4, HIGH);
```

```
    } else {
```

```
      digitalWrite(D4, LOW);
```

```
    }
```

```
    if (MtComida == "1")
```

```
    {
```

```
      digitalWrite(D3, HIGH);
```

```
} else {  
    digitalWrite(D3, LOW);  
}  
}  
}
```

```
void Dispensacion() {  
    switch (Hora.toInt()) {  
        case 1:  
            if (hr == 12) {  
                if (agua < STAgua.toInt() && dispena == 0)  
                {  
                    digitalWrite(D4, HIGH);  
                } else {  
                    digitalWrite(D4, LOW);  
                }  
                if (pesa1 < STComida.toInt() && dispenc == 0)  
                {  
                    digitalWrite(D3, HIGH);  
                } else {  
                    digitalWrite(D3, LOW);  
                }  
            } else {  
                dispena = 0;  
                dispenc = 0;  
            }  
            if (agua >= STAgua.toInt()) {  
                dispena = 1;  
            }  
        }  
    }  
}
```

```

}
if (pesa1 >= STComida.toInt()) {
    dispenc = 1;
}
break;
case 2:
if (hr == 8 || hr == 16) {
    if (agua < STAgua.toInt() && dispena == 0)
    {
        digitalWrite(D4, HIGH);
    } else {
        digitalWrite(D4, LOW);
    }
    if (pesa1 < STComida.toInt() && dispenc == 0)
    {
        digitalWrite(D3, HIGH);
    } else {
        digitalWrite(D3, LOW);
    }
} else {
    dispena = 0;
    dispenc = 0;
}
if (agua >= STAgua.toInt()) {
    dispena = 1;
}
if (pesa1 >= STComida.toInt()) {
    dispenc = 1;
}

```

```
}
```

```
break;
```

```
case 3:
```

```
if (hr == 6 || hr == 12 || hr == 18) {
```

```
    if (agua < STAgua.toInt() && dispena == 0)
```

```
    {
```

```
        digitalWrite(D4, HIGH);
```

```
    } else {
```

```
        digitalWrite(D4, LOW);
```

```
    }
```

```
if (pesa1 < STComida.toInt() && dispenc == 0)
```

```
{
```

```
    digitalWrite(D3, HIGH);
```

```
} else {
```

```
    digitalWrite(D3, LOW);
```

```
}
```

```
} else {
```

```
    dispena = 0;
```

```
    dispenc = 0;
```

```
}
```

```
if (agua >= STAgua.toInt()) {
```

```
    dispena = 1;
```

```
}
```

```
if (pesa1 >= STComida.toInt()) {
```

```
    dispenc = 1;
```

```
}
```

```
break;
```

```
case 4:
```

```
if (hr == 5 || hr == 10 || hr == 15 || hr == 21) {
    if (agua < STAgua.toInt() && dispena == 0)
    {
        digitalWrite(D4, HIGH);
    } else {
        digitalWrite(D4, LOW);
    }
    if (pesa1 < STComida.toInt() && dispenc == 0)
    {
        digitalWrite(D3, HIGH);
    } else {
        digitalWrite(D3, LOW);
    }
} else {
    dispena = 0;
    dispenc = 0;
}
if (agua >= STAgua.toInt()) {
    dispena = 1;
}
if (pesa1 >= STComida.toInt()) {
    dispenc = 1;
}
break;
default:
    digitalWrite(D3, LOW);
    digitalWrite(D4, LOW);
    dispena = 0;
```



```
    dispenc = 0;
    break;
}
}

void galga() {

    if (scale1.is_ready()) {
        //Serial.println(scale1.get_units(20));
        pesa1 = scale1.get_units(20);
        if (pesa1 < 0) {
            pesa1 = 0;
        }
    } else {
        //Serial.println("HX711_1 not found.");
    }

    if (scale2.is_ready()) {
        pesa2 = scale2.get_units(20);
        agua = map(pesa2, 0, 430, 0, 100);
        if (pesa2 < 0) {
            pesa2 = 0, agua = 0;
        }
        if (pesa2 > 100) {
            pesa2 = 430, agua = 100;
        }
    } else {
        //Serial.println("HX711_2 not found.");
    }
}
```

```

}
}
void Alarmas() {
    if (analogRead(A0) < 200) {
        Firebase.setInt(firebase, "/ComedorMascotas/AComida", 1);
    }
    else {
        Firebase.setInt(firebase, "/ComedorMascotas/AComida", 0);
    }
    Firebase.setInt(firebase, "/ComedorMascotas/AAgua", digitalRead(D0));

}

void VariablesIn() {
    Firebase.getString(firebase, "/ComedorMascotas/Hora");
    Hora = firebase.stringData();
    Hora = Hora.substring(Hora.indexOf("") + 1, Hora.lastIndexOf("") - 1);

    Firebase.getString(firebase, "/ComedorMascotas/STComida");
    STComida = firebase.stringData();
    STComida = STComida.substring(STComida.indexOf("") + 1, STComida.lastIndexOf("") - 1);

    Firebase.getString(firebase, "/ComedorMascotas/STAgua");
    STAgua = firebase.stringData();
    STAgua = STAgua.substring(STAgua.indexOf("") + 1, STAgua.lastIndexOf("") - 1);

    Firebase.getString(firebase, "/ComedorMascotas/MManual");
    MManual = firebase.stringData();

```

```
Firestore.getString(firebase, "/ComedorMascotas/MtAgua");
```

```
MtAgua = firebase.stringData();
```

```
//MtAgua = MtAgua.substring(MtAgua.indexOf("") + 1, MtAgua.lastIndexOf("") - 1);
```

```
Firestore.getString(firebase, "/ComedorMascotas/MtComida");
```

```
MtComida = firebase.stringData();
```

```
//MtComida = MtComida.substring(MtComida.indexOf("") + 1, MtComida.lastIndexOf("") - 1);
```

```
Firestore.getString(firebase, "/ComedorMascotas/newTime");
```

```
newTime = firebase.stringData();
```

```
//newTime = newTime.substring(newTime.indexOf("") + 1, newTime.lastIndexOf("") - 1);
```

```
if (newTime == "1") {
```

```
    Firestore.getString(firebase, "/ComedorMascotas/STime");
```

```
    STime = firebase.stringData();
```

```
    fan = STime.substring(STime.indexOf(",YEAR=") + 6, STime.indexOf(",MONTH="));
```

```
    fms = STime.substring(STime.indexOf(",MONTH=") + 7, STime.indexOf(",WEEK_OF_YEAR="));
```

```
    fdy = STime.substring(STime.indexOf(",DAY_OF_MONTH=") + 14, STime.indexOf(",DAY_OF_YEAR="));
```

```
    fho = STime.substring(STime.indexOf(",HOUR_OF_DAY=") + 13, STime.indexOf(",MINUTE="));
```

```
    fmn = STime.substring(STime.indexOf(",MINUTE=") + 8, STime.indexOf(",SECOND="));
```

```
    fsg = STime.substring(STime.indexOf(",SECOND=") + 8, STime.indexOf(",MILLISECOND="));
```

```
}
```

```
}
```

```
void VariablesOut() {
```

```
    galga();
```

```
    lcd.setCursor(0, 1);
```

```
    lcd.print("      ");
```

```
    lcd.setCursor(0, 1);
```

```
if (digitalRead(D0) == 0) {
```

```
    lcd.write(byte(1));
```

```
}
```

```
else {
```

```
    lcd.write(byte(0));
```

```
}
```

```
lcd.print(":");
```

```
lcd.print(agua);
```

```
lcd.print("%");
```

```
lcd.setCursor(8, 1);
```

```
if (analogRead(A0) > 200) {
```

```
    lcd.write(byte(3));
```

```
}
```

```
else {
```

```
    lcd.write(byte(2));
```

```
}
```

```
lcd.print(":");
```

```
lcd.print(pesa1);
```

```
lcd.write(byte(4));
```

```
String Actualwt = (String)agua + "%";
```

```
String Actualgr = (String)pesa1 + "gr";
```

```
Firebase.setString(firebase, "/ComedorMascotas/Actualgr", Actualgr);
```

```
Firebase.setString(firebase, "/ComedorMascotas/Actualwt", Actualwt);
```

```
}
```

```
void Tiempo() {
```

```
String aio, mes, dia, hora, minuto, segund;
```

```
char y, me, d, h, m, s;
```

```
//if(Time!="" || newTime!="" || newTime.length()<10){
```

```
if (newTime == "1") {
```

```
    Firebase.setString(firebase, "/ComedorMascotas/newTime", "0");
```

```
    RTC.adjust(DateTime(fan.toInt(), (fms.toInt()) + 1, fdy.toInt(), fho.toInt(), fmn.toInt(), fsg.toInt()));
```

```
}
```

```
lcd.print("      ");
```

```
DateTime now = RTC.now(); // Obtiene la fecha y hora del RTC
```

```
ms = now.month();
```

```
if (ms < 10) {
```

```
    mes = "0" + (String)ms;
```

```
} else {
```

```
    mes = (String)ms;
```

```
}
```

```
me = (char)ms;
```

```
dy = now.day();
```

```
if (dy < 10) {
```

```
    dia = "0" + (String)dy;
```

```
} else {
```

```
    dia = (String)dy;
```

```
}
```

```
hr = now.hour();
```

```
if (hr < 10) {
```

```
    hora = "0" + (String)hr;
```

```
} else {
```

```
    hora = (String)hr;
}
mn = now.minute();
if (mn < 10) {
    minuto = "0" + (String)mn;
} else {
    minuto = (String)mn;
}
lcd.setCursor(0, 0);
lcd.print(hora);
lcd.print(":");
lcd.print(minuto);
lcd.print(" ");
lcd.setCursor(7, 0);
lcd.print(mes);
lcd.print("/");
lcd.print(dia);

n_anno = (now.year() - 2000);
d_anno = n_anno / 4;
n_dia = now.day();
n_mes = now.month();

switch (n_mes) {
    case 1:
        t_mes = 0;
        break;
    case 2:
```

```
t_mes = 3;
```

```
break;
```

```
case 3:
```

```
t_mes = 3;
```

```
break;
```

```
case 4:
```

```
t_mes = 6;
```

```
break;
```

```
case 5:
```

```
t_mes = 1;
```

```
break;
```

```
case 6:
```

```
t_mes = 4;
```

```
break;
```

```
case 7:
```

```
t_mes = 6;
```

```
break;
```

```
case 8:
```

```
t_mes = 2;
```

```
break;
```

```
case 9:
```

```
t_mes = 5;
```

```
break;
```

```
case 10:
```

```
t_mes = 0;
```

```
break;
```

```
case 11:
```

```
t_mes = 3;
```

```
    break;
case 12:
    t_mes = 5;
    break;
default:
    t_mes = t_mes;
    break;
}
```

```
r_dia = n_dia + t_mes + n_anno + d_anno + t_siglo;
```

```
r_dia = r_dia % 7;
```

```
switch (r_dia) {
case 1:
    lcd.print(" Lun");
    break;
case 2:
    lcd.print(" Mar");
    break;
case 3:
    lcd.print(" Mie");
    break;
case 4:
    lcd.print(" Jue");
    break;
case 5:
    lcd.print(" Vie");
    break;
```


case 6:

```
lcd.print(" Sab");
```

```
break;
```

case 0:

```
lcd.print(" Dom");
```

```
break;
```

default:

```
lcd.print("---");
```

```
break;
```

```
}
```

```
}
```