

```

#include <DFRobot_RainfallSensor.h>
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BME280.h>
#include <SPI.h>
#include <MFRC522.h>

// ===== LCD =====
LiquidCrystal_I2C lcd(0x27, 16, 2);

// ===== BME280 =====
#define SDA_PIN 21
#define SCL_PIN 22
Adafruit_BME280 bme;

// ===== PLUVIÓMETRO =====
DFRobot_RainfallSensor_UART rainSensor(&Serial2);
uint32_t baseConteo = 0;
uint32_t ultimoConteo = 0;
uint32_t ultimoMostrado = 0;

const uint32_t UMBRAL_LLUVIA = 20;
const unsigned long DURACION_ALERTA_LLUVIA = 10000;
bool alertaLluvia = false;
unsigned long tiempoAlertaLluvia = 0;

// ===== BME TIEMPO =====
unsigned long ultimoBME = 0;
const unsigned long INTERVALO_BME = 60000;
float temperatura = 0;
float humedad = 0;

// ===== RFID =====
#define SS_PIN 5
#define RST_PIN 27
#define SCK_PIN 18
#define MISO_PIN 19
#define MOSI_PIN 23

MFRC522 mfr522(SS_PIN, RST_PIN);

struct Persona {

```

```

    byte uid[4];
    const char* nombre;
};

Persona personas[] = {
    {{0xB5, 0x3A, 0xC9, 0xE5}, "Itzurun"}
};

const int totalPersonas = sizeof(personas) / sizeof(personas[0]);
String nombreConductor = "Desconocido";

// ===== INFRARROJOS =====
const int sensor1Pin = 35;
const int sensor2Pin = 34;
const float distancia = 0.235;
const int UMBRAL_DISPARO = 2000;

unsigned long tiempoInicio = 0;
bool sensor1Activo = false;
float ultimaVelocidad = 0;

bool alertaVelocidad = false;
unsigned long tiempoAlertaVelocidad = 0;
const unsigned long DURACION_ALERTA_VEL = 5000;

// ===== FUNCIONES =====
int leerLimpio(int pin) {
    long suma = 0;
    for(int i=0; i<10; i++) suma += analogRead(pin);
    return suma / 10;
}

bool compararUID(byte *uid1, byte *uid2, byte size) {
    for (byte i = 0; i < size; i++)
        if (uid1[i] != uid2[i]) return false;
    return true;
}

void setup() {

    Serial.begin(115200);

    Wire.begin(SDA_PIN, SCL_PIN);
    lcd.init();

```

```

    lcd.backlight();

    // UART lluvia
    Serial2.begin(9600, SERIAL_8N1, 16, 17);
    rainSensor.begin();
    baseConteo = rainSensor.getRawData();

    // BME
    bme.begin(0x77);
    temperatura = bme.readTemperature();
    humedad = bme.readHumidity();
    ultimoBME = millis();

    // RFID
    SPI.begin(SCK_PIN, MISO_PIN, MOSI_PIN, SS_PIN);
    mfrc522.PCD_Init();

    analogSetAttenuation(ADC_11db);

    lcd.print("Sistema listo");
    delay(2000);
    lcd.clear();
}

void loop() {

    unsigned long ahora = millis();

    // ===== LLUVIA =====
    uint32_t rawCount = rainSensor.getRawData();
    uint32_t vuelcos = rawCount - baseConteo;

    if (vuelcos != ultimoMostrado) {
        Serial.print("Vuelcos: ");
        Serial.println(vuelcos);
        ultimoMostrado = vuelcos;
    }

    if (vuelcos >= UMBRAL_LLUVIA && !alertaLluvia) {
        alertaLluvia = true;
        tiempoAlertaLluvia = millis();
        ultimoConteo = rawCount;
    }
}

```

```

if (alertaLluvia) {

    if (rawCount > ultimoConteo) {
        tiempoAlertaLluvia = millis();
        ultimoConteo = rawCount;
    }

    if (millis() - tiempoAlertaLluvia < DURACION_ALERTA_LLUVIA) {
        lcd.setCursor(0,0);
        lcd.print("LLUVIA INTENSA ");
        lcd.setCursor(0,1);
        lcd.print("Reducir vel.  ");
        return;
    } else {
        alertaLluvia = false;
        baseConteo = rawCount;
        ultimoMostrado = 0;
        lcd.clear();
    }
}

// ===== BME =====
if (ahora - ultimoBME >= INTERVALO_BME) {
    temperatura = bme.readTemperature();
    humedad = bme.readHumidity();
    ultimoBME = ahora;
}

// ===== RFID =====
if (mfrc522.PICC_IsNewCardPresent() && mfrc522.PICC_ReadCardSerial())
{

    bool encontrado = false;

    for (int i = 0; i < totalPersonas; i++) {
        if (compararUID(mfrc522.uid.uidByte, personas[i].uid,
mfrc522.uid.size)) {
            nombreConductor = personas[i].nombre;
            encontrado = true;
        }
    }
}

```

```

    if (!encontrado) nombreConductor = "Desconocido";

    Serial.print("Conductor: ");
    Serial.println(nombreConductor);

    mfrc522.PICC_HaltA();
}

// ===== VELOCIDAD =====
int v1 = leerLimpio(sensor1Pin);

if (!sensor1Activo && v1 > UMBRAL_DISPARO) {
    tiempoInicio = millis();
    sensor1Activo = true;
}

if (sensor1Activo) {

    int v2 = leerLimpio(sensor2Pin);

    if (v2 > UMBRAL_DISPARO) {

        unsigned long tiempoMs = millis() - tiempoInicio;

        if (tiempoMs > 20) {
            ultimaVelocidad = (distancia / (tiempoMs / 1000.0)) * 3.6;

            Serial.print("Velocidad: ");
            Serial.println(ultimaVelocidad);

            if (ultimaVelocidad > 10.0) {
                alertaVelocidad = true;
                tiempoAlertaVelocidad = millis();
            }
        }

        sensor1Activo = false;
    }

    if (millis() - tiempoInicio > 3000) {
        sensor1Activo = false;
    }
}

```

```

// ===== ALERTA VELOCIDAD =====
if (alertaVelocidad) {

    if (millis() - tiempoAlertaVelocidad < DURACION_ALERTA_VEL) {
        lcd.setCursor(0,0);
        lcd.print("EXCESO VELOC.  ");
        lcd.setCursor(0,1);
        lcd.print("Reducir velocida ");
        return;
    } else {
        alertaVelocidad = false;
        lcd.clear();
    }
}

// ===== PANTALLA NORMAL =====
lcd.setCursor(0,0);
lcd.print("T:");
lcd.print(temperatura,1);
lcd.print("C H:");
lcd.print(humedad,0);
lcd.print("% ");

lcd.setCursor(0,1);
lcd.print(nombreConductor.substring(0,8));
lcd.print(" ");
lcd.print(ultimaVelocidad,1);
lcd.print("km ");
}

```