

# Aro de 16 LEDs WS2812 con Arduino

¡Muy buen proyecto! 🔥 Un aro de **16 LEDs WS2812** (también llamado **Neopixel**) es un componente espectacular para proyectos visuales con Arduino.

Aquí te paso unas **recomendaciones importantes** para hacerlo funcionar bien con un **Arduino Uno**:

## ⚡ Recomendaciones para usar WS2812 (Neopixel Ring)

### 1. Librería necesaria

- Usa la librería **Adafruit Neopixel** (muy estable y sencilla de usar).  
👉 En el IDE de Arduino: **Herramientas > Administrar Bibliotecas > Adafruit Neopixel > Instalar**.
- 

### 2. Conexiones básicas

- **VCC (5V)** → al pin **5V** del Arduino (si es solo el aro de 16 LEDs, alcanza).
- **GND** → a **GND** del Arduino.
- **DIN (Data In)** → a un pin digital del Arduino (ejemplo: pin **6**).
- ⚠ Muy importante: **GND del Arduino y GND del aro deben estar conectados juntos**.

### 3. Resistencia y condensador

- Coloca una **resistencia de 330–470Ω** en serie con la línea de datos (entre el pin del Arduino y DIN del aro).
- Coloca un **condensador de 1000 μF / 6.3V o más** entre **+5V y GND** cerca del aro, para evitar picos de corriente.
- 

## 📖 Manual de armado

### ◆ Materiales

- 1 Arduino UNO
- 1 aro de **16 LEDs WS2812**
- 4 botones (push button)
- 1 resistencia de **330Ω** para la línea de datos
- 1 condensador de **1000μF** entre 5V y GND del aro
- Protoboard y cables jumper

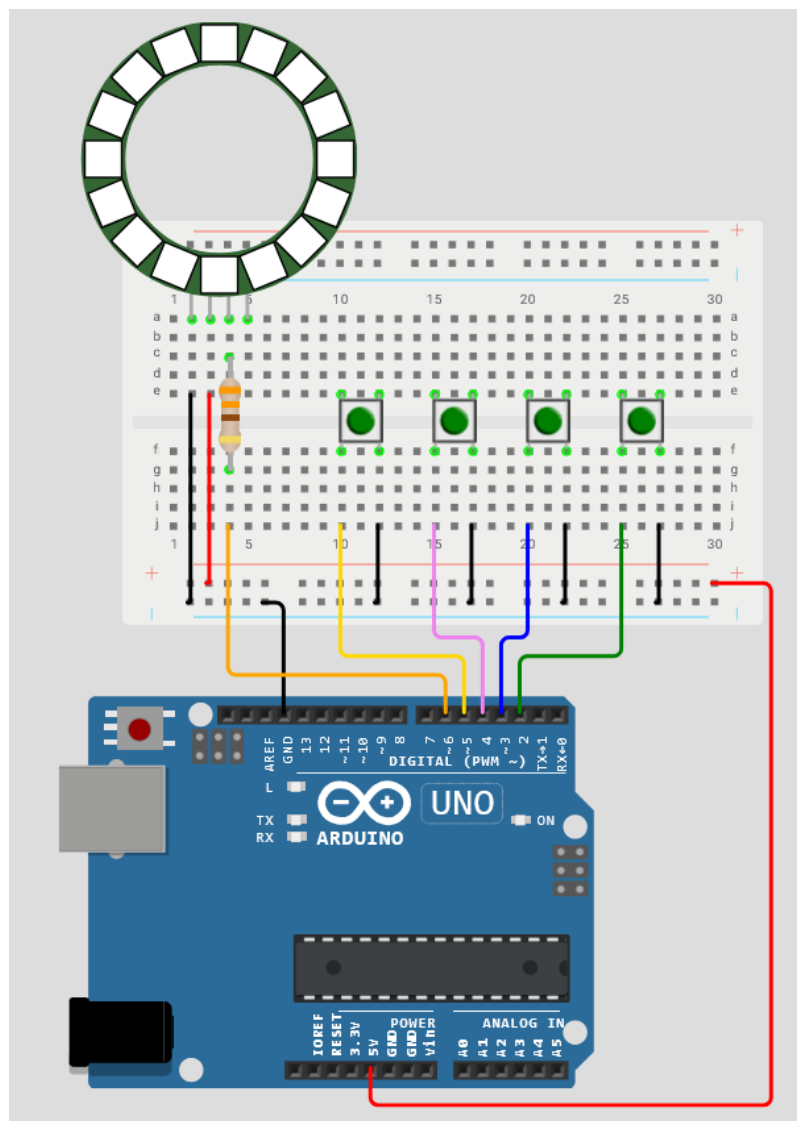


## ◆ Conexiones

- **Aro WS2812**
  - VCC → 5V Arduino
  - GND → GND Arduino
  - DIN → Pin 6 (a través de resistencia de 330Ω)
- **Botones**
  - Botón 1 → Pin 2
  - Botón 2 → Pin 3
  - Botón 3 → Pin 4
  - Botón 4 → Pin 5
  - Cada botón → conectado a **GND**

👉 Con este sistema, los niños podrán aprender:

- Uso de **botones con INPUT\_PULLUP**
- Manejo de **colores RGB**
- **Animaciones básicas** con el aro



## PROGRAMA PARA CARGAR AL ARDUINO UNO

```
#include <Adafruit_NeoPixel.h>

#define PIN_NEOPIXEL 6      // Pin de datos del aro WS2812
#define NUMPIXELS 16        // Cantidad de LEDs del aro

// Pines de los 4 botones
#define BOTON1 2
#define BOTON2 3
#define BOTON3 4
#define BOTON4 5

Adafruit_NeoPixel pixels(NUMPIXELS, PIN_NEOPIXEL, NEO_GRB + NEO_KHZ800);

int brillo = 80; // Brillo de los LEDs (0-255)

void setup() {
  pixels.begin();
  pixels.setBrightness(brillo);
  pixels.show(); // Apagar todos al inicio

  pinMode(BOTON1, INPUT_PULLUP);
  pinMode(BOTON2, INPUT_PULLUP);
  pinMode(BOTON3, INPUT_PULLUP);
  pinMode(BOTON4, INPUT_PULLUP);
}

void loop() {
  bool b1 = !digitalRead(BOTON1);
  bool b2 = !digitalRead(BOTON2);
  bool b3 = !digitalRead(BOTON3);
  bool b4 = !digitalRead(BOTON4);

  // --- Colores individuales ---
  if (b1 && !b2 && !b3 && !b4) {
    setColor(255, 0, 0); // ● Rojo
  }
  else if (b2 && !b1 && !b3 && !b4) {
    setColor(0, 255, 0); // ● Verde
  }
  else if (b3 && !b1 && !b2 && !b4) {
    setColor(0, 0, 255); // ● Azul
  }

  // --- Combinaciones de botones ---
  else if (b1 && b2) {
    setColor(255, 255, 0); // Rojo + Verde = Amarillo
  }
}
```

```

else if (b1 && b3) {
    setColor(255, 0, 255); // Rojo + Azul = Magenta
}
else if (b2 && b3) {
    setColor(0, 255, 255); // Verde + Azul = Cian
}

// --- Secuencias aleatorias ---
else if (b4) {
    randomSequence();
}
}

// Función para encender todo el aro con un color
void setColor(int r, int g, int b) {
    for (int i = 0; i < NUMPIXELS; i++) {
        pixels.setPixelColor(i, pixels.Color(r, g, b));
    }
    pixels.show();
}

// Función para secuencias aleatorias
void randomSequence() {
    int opcion = random(1, 4); // 1 a 3
    if (opcion == 1) {
        // Secuencia 1: giro
        for (int i = 0; i < NUMPIXELS; i++) {
            pixels.clear();
            pixels.setPixelColor(i, pixels.Color(0, 0, 255)); // ● Azul
            pixels.show();
            delay(100);
        }
    } else if (opcion == 2) {
        // Secuencia 2: todos parpadean
        setColor(255, 0, 255); delay(200);
        setColor(0, 0, 0); delay(200);
    } else if (opcion == 3) {
        // Secuencia 3: arcoíris
        rainbow(20);
    }
}

// Efecto arcoíris
void rainbow(int wait) {
    for (long firstPixelHue = 0; firstPixelHue < 5 * 65536; firstPixelHue
+= 256) {
        for (int i = 0; i < pixels.numPixels(); i++) {
            int pixelHue = firstPixelHue + (i * 65536 / pixels.numPixels());
            pixels.setPixelColor(i, pixels.gamma32(pixels.ColorHSV(pixelHue)));
        }
    }
}

```

```
    }  
    pixels.show();  
    delay(wait);  
  }  
}
```