

# SEMANA 2

---



## Temario Completo: Sensores en Arduino

### ● ¿Qué es un Sensor?

Un **sensor** es un dispositivo capaz de detectar cambios físicos, químicos o ambientales y convertirlos en señales eléctricas que pueden ser procesadas por un microcontrolador como Arduino. En términos más simples, los sensores actúan como los "órganos sensoriales" de tu proyecto electrónico: ven, oyen, sienten el calor, detectan movimiento, miden luz, humedad, presión, entre otros.

Los sensores se utilizan en una gran variedad de aplicaciones:

- Sistemas de automatización del hogar
- Monitoreo ambiental
- Control industrial
- Proyectos educativos y experimentales

En Arduino, los sensores son fundamentales porque permiten interactuar con el entorno físico real.

---

### ● Tipos de Sensores Comunes en Arduino

Existen muchos tipos de sensores compatibles con Arduino. Aquí algunos de los más comunes:

Nombre	Tipo	Descripción
LM35	Analógico	Mide temperatura en grados Celsius
DHT11 / DHT22	Digital	Mide temperatura y humedad
LDR (Light Dependent Resistor)	Analógico	Detecta nivel de luz
PIR (Sensor de Movimiento Pasivo)	Digital	Detecta movimiento humano/infrarrojo

Nombre	Tipo	Descripción
HC-SR04	Digital	Mide distancia ultrasónica
Sensor de humedad de suelo	Analógico/Digital	Mide nivel de humedad en tierra
MPU6050	I2C	Acelerómetro y giroscopio para medir movimiento 3D

Estos sensores pueden usarse solos o combinados en proyectos complejos como sistemas de seguridad, robots autónomos, estaciones meteorológicas caseras, etc.

---

## ● Sensores Analógicos vs. Digitales

### ◆ Sensores Analógicos

Los sensores analógicos entregan valores **continuos dentro de un rango**, típicamente de voltaje entre 0V y 5V. Arduino traduce estos valores a un rango numérico usando su **Convertidor Analógico-Digital (ADC)**, mapeando de 0V a 1023 (resolución de 10 bits).

#### Ejemplo:

Cuando conectas un sensor LDR (fotoresistencia), este cambia su resistencia según la cantidad de luz. Al colocarlo en un divisor de tensión, Arduino puede leer ese cambio con `analogRead(pin)`.

```
int valor = analogRead(A0); // Lee valor entre 0 y 1023
```

### ◆ Sensores Digitales

Los sensores digitales solo entregan dos estados: **alto (HIGH)** o **bajo (LOW)**. Esto equivale a un comando "ON/OFF".

#### Ejemplo:

El sensor PIR (de movimiento) entrega un HIGH cuando detecta movimiento. Arduino lo lee así:

```
int estado = digitalRead(7); // Devuelve HIGH o LOW
```

La elección entre usar un sensor analógico o digital dependerá de la precisión necesaria y la naturaleza del fenómeno que deseas medir.

---

## Conexión de Sensores al Arduino

Para conectar cualquier sensor a Arduino, sigue estos pasos generales:

1. **Identifica los pines del sensor:** VCC (alimentación), GND (tierra) y OUT (señal).
2. **Alimentación:** Conecta VCC a 5V o 3.3V dependiendo del sensor.
3. **Tierra:** Conecta GND a GND de Arduino.
4. **Señal:** Conecta OUT al pin correspondiente:
  - Si es analógico → a un pin A0, A1, etc.
  - Si es digital → a un pin 2, 3, ..., 13
5. **Resistencias Pull-up/Pull-down (si aplica):** Algunos sensores requieren esto para evitar flotantes.

**Consejo:** Revisa siempre el datasheet del sensor para conocer las especificaciones de conexión y voltaje.

---

## Lectura de Datos desde Sensores

Una vez conectado el sensor, debes leer sus datos desde el código Arduino.

### ◆ Para sensores analógicos:

Usa la función `analogRead(pin)` donde `pin` corresponde a uno de los pines analógicos (A0, A1...).

```
void loop() {  
    int valorLuz = analogRead(A0);  
    Serial.println(valorLuz);  
    delay(1000);  
}
```

Este código imprime el valor leído del sensor cada segundo por el monitor serial.

### ◆ Para sensores digitales:

Usa `digitalRead(pin)`:

```
int estadoMovimiento = digitalRead(7);  
if (estadoMovimiento == HIGH) {
```

```
Serial.println("¡Movimiento detectado!");  
}
```

Con estas herramientas puedes construir lógica de control basada en las condiciones del entorno.

---

## Uso del Convertidor ADC

Arduino dispone de un **Convertidor Analógico a Digital (ADC)** integrado en su microcontrolador. Este permite tomar las señales analógicas provenientes de sensores y convertirlas en valores digitales que puedan ser procesados por el programa.

### Características del ADC en Arduino UNO:

- Tiene una resolución de **10 bits**
- Esto significa que puede devolver valores entre **0 y 1023**
- El rango de voltaje de entrada es de **0V a 5V**

Por ejemplo:

- 0V → 0
- 2.5V → 512
- 5V → 1023

El ADC permite hacer mediciones precisas de magnitudes analógicas sin necesidad de componentes externos adicionales.

---

## Manejo de Bibliotecas para Sensores

Muchos sensores vienen con bibliotecas especializadas que facilitan su uso. Estas bibliotecas contienen funciones predefinidas para inicializar, configurar y leer los valores del sensor.

### Ejemplo: Usando el sensor DHT11

1. Instala la biblioteca `DHT.h` desde el **Manejador de Bibliotecas** del IDE de Arduino.
2. Incluye la biblioteca en tu sketch:

```
#include <DHT.h>
```

3. Define el tipo de sensor y el pin usado:

```
#define PIN_SENSOR 2
#define TIPO_SENSOR DHT11
DHT dht(PIN_SENSOR, TIPO_SENSOR);
```

4. Lee los datos:

```
float humedad = dht.readHumidity();
float temperatura = dht.readTemperature();

Serial.print("Humedad: ");
Serial.print(humedad);
Serial.println(" %");

Serial.print("Temperatura: ");
Serial.print(temperatura);
Serial.println(" °C");
```

Las bibliotecas simplifican enormemente la programación y permiten acceder a características avanzadas sin tener que programar desde cero.



## Conclusión

Entender cómo funcionan los sensores y cómo se integran con Arduino es clave para desarrollar proyectos electrónicos interactivos. Ya sea que quieras construir un sistema de riego automático, una alarma de seguridad o un robot que evite obstáculos, los sensores son tus herramientas principales.

Esta semana aprendiste:

- Qué es un sensor y su importancia
- Tipos de sensores comunes y sus aplicaciones
- Cómo diferenciar entre sensores analógicos y digitales
- Cómo conectar y leer sensores desde Arduino
- El papel del ADC en la conversión de señales
- Cómo usar bibliotecas para facilitar la programación

# Referencias

Aquí tienes algunas fuentes confiables para profundizar en estos temas:

- [Arduino Official Website](#)
  - [Arduino Playground](#)
  - [Adafruit Learning System](#)
  - Libro: *Beginning Arduino* – Michael McRoberts
  - Documentación oficial de sensores (Datasheets):
    - [LM35 Datasheet](#)
    - [DHT11/22 Datasheet](#)
    - [MPU6050 Datasheet](#)
-