

---

title: "MECATRÓNICA - MANUAL COMPLETO"

subtitle: "Desde Cero hasta Nivel Profesional FP Superior"

author: "Mecatrónica Profesional"

keywords: "mecatrónica, PLC, automatización, robótica, neumática, hidráulica, Arduino, programación"

papersize: a4

toc: true

toc-depth: 2

number-sections: true

---

\newpage

# MECATRÓNICA: MANUAL COMPLETO DE ESTUDIO

---

**Desde Cero Absoluto hasta Nivel Profesional FP Superior**

**Bienvenido al mundo de la mecatrónica.**

Este libro está diseñado para personas que quieren aprender mecatrónica desde cero total. No necesitas conocimientos previos de matemáticas, física ni electricidad. Solo necesitas ganas de aprender y curiosidad por entender cómo funcionan las máquinas que nos rodean.

La mecatrónica es una de las disciplinas más apasionantes de la ingeniería moderna porque combina cuatro áreas fundamentales:

- **Mecánica:** el estudio de las piezas que se mueven, las estructuras que soportan cargas y los mecanismos que transmiten fuerza.
- **Electrónica:** el mundo de los circuitos, los sensores que perciben el entorno y los actuadores que ejecutan acciones.
- **Informática:** la inteligencia que procesa información, toma decisiones y controla los sistemas.
- **Control:** la estrategia que coordina todo lo anterior para que funcione de manera automática y precisa.

Un profesional de la mecatrónica es como un “médico de máquinas”: entiende todos los sistemas que hacen funcionar una fábrica, un robot o un vehículo moderno.

## Estructura del libro

El libro está organizado en 5 niveles progresivos. Cada nivel asume que has comprendido el anterior:

- **NIVEL 0 — Fundamentos absolutos:** Matemáticas básicas, física intuitiva y electricidad fundamental. Todo explicado con ejemplos cotidianos para quien parte de cero.
- **NIVEL 1 — Base de mecatrónica:** Mecánica de mecanismos, componentes electrónicos, sensores y actuadores. Los bloques de construcción de cualquier sistema automatizado.
- **NIVEL 2 — Programación y control:** Lógica de programación, Arduino, automatización ON/OFF, temporizadores y control secuencial.
- **NIVEL 3 — FP Superior:** PLC industrial, lenguaje Ladder, neumática, hidráulica y sistemas mecatrónicos integrados. El nivel exigido en Formación Profesional Superior.
- **NIVEL 4 — Avanzado profesional:** Robótica industrial, cinemática, SCADA, Industria 4.0 e integración total de fábricas.

## Cómo usar este libro

Cada capítulo sigue la misma estructura para facilitar el aprendizaje:

1. **Explicación teórica simple:** los conceptos fundamentales explicados con claridad.
2. **Ejemplo práctico:** un caso concreto que ilustra la teoría.
3. **Aplicación real:** cómo se usa ese conocimiento en la industria o la vida diaria.
4. **Resumen en puntos clave:** lo más importante para recordar.
5. **Ejercicio o práctica:** para que compruebes que has entendido.

Se recomienda leer los capítulos en orden, hacer los ejercicios y no pasar al siguiente nivel hasta dominar el anterior.

\newpage

**Nota importante sobre las matemáticas:** No tengas miedo a las fórmulas. En mecatrónica solo necesitas operaciones básicas: sumar, restar, multiplicar, dividir y usar la regla de tres. Todo lo demás se construye sobre estas bases.

\newpage

---

# NIVEL 0: FUNDAMENTOS ABSOLUTOS

---

*“Nadie nace sabiendo. Todo se puede aprender paso a paso.”*

Este nivel está diseñado para personas que nunca han estudiado tecnología. Aquí construiremos los cimientos sólidos que necesitarás para el resto del libro: matemáticas básicas, física intuitiva y electricidad fundamental.

Si ya tienes conocimientos técnicos previos, puedes saltar directamente al Nivel 1. Pero incluso en ese caso, te recomiendo repasar este nivel para asegurarte de que no hay lagunas en tu base.

---

## CAPÍTULO 0.1: MATEMÁTICAS BÁSICAS I – ARITMÉTICA Y REGLA DE TRES

### 0.1.1 ¿Por qué necesitamos matemáticas en mecatrónica?

Las matemáticas son el idioma de la ingeniería. Igual que necesitas palabras para formar frases y comunicarte, necesitas números y operaciones para describir cómo funcionan las máquinas, calcular valores de componentes y dimensionar sistemas.

No te preocupes: en mecatrónica no usamos matemáticas abstractas ni complicadas. Usamos operaciones muy concretas para resolver problemas reales:

- Calcular qué resistencia necesita un LED para no quemarse
- Determinar a qué velocidad girará un motor con cierto voltaje
- Calcular cuánta fuerza ejerce un cilindro neumático
- Dimensionar el cable adecuado para una instalación

Todas estas operaciones se resuelven con las cuatro operaciones básicas y la regla de tres.

### 0.1.2 Las cuatro operaciones fundamentales

**Suma (+):** Juntar cantidades.

*Ejemplo cotidiano:* Si tienes 3 resistencias en tu caja de componentes y compras 5 más, el total es  $3 + 5 = 8$  resistencias.

*Ejemplo técnico:* Un circuito tiene tres motores que consumen 2 A, 1.5 A y 0.5 A respectivamente. La corriente total es  $2 + 1.5 + 0.5 = 4$  A.

**Resta (-):** Quitar cantidades o encontrar diferencias.

*Ejemplo cotidiano:* Si tenías 8 resistencias y usas 3 en un proyecto, te quedan  $8 - 3 = 5$  resistencias.

*Ejemplo técnico:* La caída de tensión en un cable es la diferencia entre el voltaje al inicio y al final. Si entran 24 V y llegan 22.5 V, la caída es  $24 - 22.5 = 1.5$  V.

**Multiplicación (×):** Sumar repetidas veces la misma cantidad.

*Ejemplo cotidiano:* Si 3 motores consumen 2 amperios cada uno, el consumo total es  $3 \times 2 = 6$  amperios.

*Ejemplo técnico:* La potencia eléctrica se calcula multiplicando voltaje por corriente. Si un motor trabaja a 12 V y consume 3 A, la potencia es  $12 \times 3 = 36$  W.

**División (÷):** Repartir en partes iguales o encontrar cuántas veces cabe una cantidad en otra.

*Ejemplo cotidiano:* Si 12 voltios se dividen entre 3 resistencias iguales en serie, cada una recibe  $12 \div 3 = 4$  voltios.

*Ejemplo técnico:* Para calcular la corriente conocidos el voltaje y la resistencia:  $I = V \div R$ . Con 24 V y 8  $\Omega$ ,  $I = 24 \div 8 = 3$  A.

### 0.1.3 La regla de tres

La regla de tres es, con diferencia, la herramienta matemática más útil en mecatrónica. Sirve para calcular un valor desconocido cuando conoces tres valores que guardan una relación proporcional.

**Estructura básica:**

Si  $A \rightarrow B$   
Entonces  $C \rightarrow X$   
 $X = (C \times B) \div A$

**Ejemplo cotidiano:**

Si 5 metros de cable cuestan 15 euros, ¿cuánto cuestan 8 metros?

Planteamiento:

- 5 metros → 15 euros
- 8 metros → X euros
- $X = (8 \times 15) \div 5 = 120 \div 5 = 24$  euros

### **Ejemplo técnico 1: Velocidad de un motor**

Un motor DC gira a 1500 RPM (revoluciones por minuto) cuando se alimenta con 12 voltios. ¿A qué velocidad girará si lo alimentamos con 9 voltios (suponiendo una relación lineal)?

- 12 V → 1500 RPM
- 9 V → X RPM
- $X = (9 \times 1500) \div 12 = 13500 \div 12 = 1125$  RPM

### **Ejemplo técnico 2: Capacidad de producción**

Una máquina produce 45 piezas en 3 horas. ¿Cuántas producirá en un turno de 8 horas?

- 3 h → 45 piezas
- 8 h → X piezas
- $X = (8 \times 45) \div 3 = 360 \div 3 = 120$  piezas

### **Ejemplo técnico 3: Conversión de unidades**

Un caudal de 60 litros/minuto equivale a 1 litro/segundo. ¿A cuántos litros/segundo equivalen 180 litros/minuto?

- 60 l/min → 1 l/s
- 180 l/min → X l/s
- $X = (180 \times 1) \div 60 = 3$  l/s

## **0.1.4 Aplicación real en la industria**

En una fábrica, los técnicos usan la regla de tres constantemente:

- **Planificación de producción:** “Si la máquina produce 50 piezas en 2 horas, ¿cuántas producirá en una semana de 40 horas?” →  $(40 \times 50) \div 2 = 1000$  piezas/semana.
- **Cálculo de materiales:** “Si para 10 piezas necesito 3 metros de cable, ¿cuánto cable necesito para 250 piezas?” →  $(250 \times 3) \div 10 = 75$  metros.

- **Ajuste de parámetros:** “Si a 50 Hz el motor gira a 1450 RPM, ¿a qué frecuencia debo ajustar el variador para obtener 1200 RPM?”  $\rightarrow (1200 \times 50) \div 1450 = 41.4 \text{ Hz}$ .

### 0.1.5 Resumen

#### Las cuatro operaciones:

- **Suma (+):** juntar cantidades
- **Resta (-):** quitar cantidades
- **Multiplicación (×):** sumar repetidas veces
- **División (÷):** repartir en partes iguales

**Regla de tres:**  $X = (C \times B) \div A$  cuando  $A \rightarrow B$  y  $C \rightarrow X$

### 0.1.6 Ejercicios

**Ejercicio 1:** Un sensor de temperatura envía una señal de 0 a 10 voltios proporcional a la temperatura de 0 a 100 °C. ¿Qué voltaje corresponderá a 65 °C?

(Solución:  $65 \times 10 \div 100 = 6.5 \text{ V}$ )

**Ejercicio 2:** Una fuente de alimentación entrega 150 W de potencia. Si tenemos 3 máquinas que consumen 40 W, 55 W y 35 W respectivamente, ¿cuánta potencia sobra?

(Solución:  $40 + 55 + 35 = 130 \text{ W}$  consumidos;  $150 - 130 = 20 \text{ W}$  disponibles)

**Ejercicio 3:** Un motor consume 2.5 A a 24 V. ¿Cuánta potencia consume? Usa  $P = V \times I$ .

(Solución:  $24 \times 2.5 = 60 \text{ W}$ )

\newpage

## CAPÍTULO 0.2: MATEMÁTICAS BÁSICAS II — PORCENTAJES Y ECUACIONES

### 0.2.1 Porcentajes: una forma de expresar proporciones

Un porcentaje (%) significa “de cada 100”. Es una forma de expresar qué cantidad representa una parte respecto al total.

**Fórmula básica:** Valor porcentual = (Porcentaje  $\div$  100)  $\times$  Cantidad total

**Ejemplo cotidiano:** Un 20% de descuento en un artículo de 80 euros significa que pagas 20 euros menos por cada 100. El descuento es:  $80 \times (20 \div 100) = 80 \times 0.20 = 16$  euros. Pagas  $80 - 16 = 64$  euros.

### **Ejemplo técnico 1: Eficiencia de un motor**

Un motor eléctrico tiene una eficiencia del 85%. Esto significa que de cada 100 vatios de energía eléctrica que consume, solo 85 vatios se convierten en energía mecánica útil. El resto se pierde en forma de calor.

Si el motor consume 500 W eléctricos, la potencia mecánica útil es:  $500 \times 0.85 = 425$  W.

### **Ejemplo técnico 2: Tolerancia de componentes**

Una resistencia de 100  $\Omega$  con tolerancia  $\pm 5\%$  puede tener un valor real entre:

- Mínimo:  $100 - (100 \times 0.05) = 100 - 5 = 95 \Omega$

- Máximo:  $100 + (100 \times 0.05) = 100 + 5 = 105 \Omega$

### **Ejemplo técnico 3: Aumento de corriente en arranque**

Un motor consume el 150% de su corriente nominal durante el arranque. Si la corriente nominal es 8 A, la corriente de arranque es:  $8 \times 1.50 = 12$  A.

Esto es importante para dimensionar los cables y protecciones: deben soportar la corriente de arranque aunque sea por poco tiempo.

## **0.2.2 Ecuaciones simples: la balanza matemática**

Una ecuación es una igualdad matemática que contiene una incógnita (normalmente representada por una letra como x). Resolver una ecuación significa encontrar el valor de la incógnita.

**Principio fundamental:** Una ecuación es como una balanza de dos platillos. Para mantener el equilibrio, todo lo que hagas en un lado debes hacerlo también en el otro.

**Regla práctica:** Lo que está sumando en un lado pasa restando al otro. Lo que está multiplicando pasa dividiendo.

### **Ejemplo 1:**

$$x + 5 = 12$$

$$x = 12 - 5$$

$$x = 7$$

**Ejemplo 2:**

$$3x = 21$$

$$x = 21 \div 3$$

$$x = 7$$

**Ejemplo 3 (combinado):**

$$2x + 4 = 16$$

$$2x = 16 - 4$$

$$2x = 12$$

$$x = 12 \div 2$$

$$x = 6$$

**Ejemplo técnico (Ley de Ohm):**

$V = I \times R$  es una ecuación. Si conocemos  $V = 24 \text{ V}$  y  $R = 8 \Omega$ , podemos encontrar  $I$ :

$$24 = I \times 8$$

$$I = 24 \div 8 = 3 \text{ A}$$

Si conocemos  $V = 12 \text{ V}$  e  $I = 0.5 \text{ A}$ , podemos encontrar  $R$ :

$$12 = 0.5 \times R$$

$$R = 12 \div 0.5 = 24 \Omega$$

**0.2.3 Aplicación real: cálculo de resistencias**

Necesitamos conectar un LED a una fuente de 12 V. El LED requiere 2 V y 20 mA (0.02 A). Necesitamos una resistencia que “se coma” el voltaje sobrante:

- Voltaje sobrante =  $12 \text{ V} - 2 \text{ V} = 10 \text{ V}$
- Usando  $V = I \times R$ :  $10 = 0.02 \times R$
- $R = 10 \div 0.02 = 500 \Omega$

Este cálculo, que combina porcentajes (si los necesitas para tolerancias) y ecuaciones simples, lo harás cientos de veces como técnico en mecatrónica.

**0.2.4 Resumen**

- **Porcentaje:** Valor = Total  $\times$  ( $\% \div 100$ )
- **Ecuación:** despejar la incógnita manteniendo el equilibrio
- **Regla práctica:** lo que suma pasa restando, lo que multiplica pasa dividiendo
- **Ley de Ohm:**  $V = I \times R$ , se despeja según lo que necesites

## 0.2.5 Ejercicios

**Ejercicio 1:** Un condensador se carga al 63% del voltaje máximo en una constante de tiempo. Si el voltaje máximo es 24 V, ¿qué voltaje alcanza en una constante de tiempo?

(Solución:  $24 \times 0.63 = 15.12$  V)

**Ejercicio 2:** Una resistencia tiene un valor nominal de 330  $\Omega$  con tolerancia  $\pm 10\%$ . ¿Cuáles son los valores mínimo y máximo?

(Solución: Mínimo =  $330 - 33 = 297$   $\Omega$ ; Máximo =  $330 + 33 = 363$   $\Omega$ )

**Ejercicio 3:** Resuelve la ecuación  $5x - 8 = 22$ .

(Solución:  $5x = 30$ ;  $x = 6$ )

\newpage

## CAPÍTULO 0.3: ÁLGEBRA Y GEOMETRÍA BÁSICA

### 0.3.1 Álgebra: el lenguaje de las fórmulas técnicas

El álgebra nos permite escribir relaciones generales usando letras que representan números. Esto es esencial en ingeniería porque las fórmulas describen cómo se comportan los sistemas físicos.

#### **Variables y constantes:**

- **Variable:** un valor que puede cambiar. Por ejemplo, la temperatura de un horno.
- **Constante:** un valor fijo. Por ejemplo, el número  $\pi$  ( $\pi$ )  $\approx 3.1416$ .
- **Parámetro:** un valor que se fija para un caso concreto. Por ejemplo, la resistencia de un componente.

#### **Ejemplo: Ley de Ohm como fórmula algebraica**

$$V = I \times R$$

Esta simple fórmula contiene tres variables. Dependiendo de lo que necesitemos calcular, despejamos la que nos interese:

- Si conocemos I y R:  $V = I \times R$  (calculamos voltaje)
- Si conocemos V y R:  $I = V \div R$  (calculamos corriente)
- Si conocemos V e I:  $R = V \div I$  (calculamos resistencia)

### Práctica de despejes:

1.  $F = m \times a$  (Segunda ley de Newton)

-  $m = F \div a$

-  $a = F \div m$

2.  $P = V \times I$  (Potencia eléctrica)

-  $V = P \div I$

-  $I = P \div V$

3.  $v = d \div t$  (Velocidad)

-  $d = v \times t$

-  $t = d \div v$

### 0.3.2 Geometría básica para mecatrónica

La geometría estudia las formas y sus medidas. En mecatrónica la usamos constantemente para calcular dimensiones de componentes.

#### Área de un círculo:

$$A = \pi \times r^2$$

Donde:

-  $\pi$  (pi)  $\approx 3.1416$

-  $r$  es el radio (la mitad del diámetro)

- El resultado se expresa en unidades al cuadrado ( $\text{cm}^2$ ,  $\text{m}^2$ )

**¿Por qué es importante?** El área de un pistón determina la fuerza que puede ejercer un cilindro neumático o hidráulico. La fórmula es: Fuerza = Presión  $\times$  Área.

**Ejemplo:** Un cilindro neumático tiene un pistón de 4 cm de radio.

$$A = 3.1416 \times 4^2 = 3.1416 \times 16 = 50.27 \text{ cm}^2$$

Esto significa que el área sobre la que actúa la presión del aire es de  $50.27 \text{ cm}^2$ . Si la presión es de 6 bar (600,000 Pa), la fuerza será:  $F = 600,000 \times 0.005027 = 3016 \text{ N}$ .

#### Perímetro de un círculo (circunferencia):

$$P = 2 \times \pi \times r$$

**¿Por qué es importante?** Para calcular la distancia lineal que recorre una correa por cada vuelta de una polea, o la distancia que avanza una rueda por cada revolución.

**Ejemplo:** Una polea tiene 6 cm de diámetro (radio = 3 cm). Por cada vuelta, la correa avanza:

$$P = 2 \times 3.1416 \times 3 = 18.85 \text{ cm}$$

### Área de un rectángulo:

$$A = \text{base} \times \text{altura}$$

**Ejemplo:** Necesitamos calcular el área de la sección transversal de un conducto de ventilación de 30 cm × 20 cm:  $A = 30 \times 20 = 600 \text{ cm}^2$ .

### Volumen de un cilindro:

$$V = \pi \times r^2 \times h$$

**Ejemplo:** Un depósito de aire comprimido cilíndrico de 0.5 m de radio y 2 m de altura:

$$V = 3.1416 \times 0.5^2 \times 2 = 3.1416 \times 0.25 \times 2 = 1.57 \text{ m}^3$$

## 0.3.3 Aplicación real: diseño de un sistema hidráulico

Imagina que necesitas diseñar un sistema hidráulico que debe levantar 10,000 kg (10 toneladas). Dispones de una presión de trabajo de 200 bar.

1. La fuerza necesaria es:  $F = 10,000 \times 9.8 = 98,000 \text{ N}$
2. La presión disponible:  $200 \text{ bar} = 20,000,000 \text{ Pa}$
3. Área necesaria del pistón:  $A = F \div P = 98,000 \div 20,000,000 = 0.0049 \text{ m}^2 = 49 \text{ cm}^2$
4. Radio del pistón:  $r = \sqrt{(A \div \pi)} = \sqrt{(49 \div 3.1416)} = \sqrt{15.6} = 3.95 \text{ cm}$
5. Diámetro del pistón:  $2 \times 3.95 = 7.9 \text{ cm} \approx 80 \text{ mm}$

Este es un ejemplo real de cómo las matemáticas básicas y la geometría se combinan para diseñar componentes industriales.

## 0.3.4 Resumen

Concepto	Fórmula	Uso en mecatrónica
Área círculo	$\pi \times r^2$	Pistones, cilindros
Circunferencia	$2 \times \pi \times r$	Poleas, engranajes
Área rectángulo	$b \times h$	Conductos, perfiles

Concepto	Fórmula	Uso en mecatrónica
Volumen cilindro	$\pi \times r^2 \times h$	Depósitos, tanques
Ley de Ohm	$V = I \times R$	Circuitos eléctricos

### 0.3.5 Ejercicios

**Ejercicio 1:** Un motor tiene una polea de 8 cm de diámetro (radio = 4 cm). ¿Qué distancia lineal avanza una correa por cada vuelta completa del motor?

(Solución:  $P = 2 \times 3.1416 \times 4 = 25.13$  cm)

**Ejercicio 2:** Un cilindro neumático tiene un diámetro de 50 mm. Calcula su área en m<sup>2</sup>.

(Solución:  $r = 0.025$  m;  $A = 3.1416 \times 0.025^2 = 0.00196$  m<sup>2</sup>)

**Ejercicio 3:** Usando la Ley de Ohm, despeja: a) I si  $V = 230$  V y  $R = 46$   $\Omega$ ; b) R si  $V = 12$  V e  $I = 0.5$  A.

(Soluciones: a)  $I = 230 \div 46 = 5$  A; b)  $R = 12 \div 0.5 = 24$   $\Omega$ )

\newpage

## CAPÍTULO 0.4: FÍSICA I — FUERZA, MASA Y PESO

### 0.4.1 Diferencias fundamentales

Uno de los conceptos más importantes que debes entender desde el principio es la diferencia entre masa y peso. Son términos que usamos indistintamente en la vida cotidiana, pero en ingeniería significan cosas distintas.

#### Masa:

- Es la cantidad de materia que tiene un objeto.
- Se mide en kilogramos (kg).
- No cambia, estés donde estés.
- Una manzana tiene aproximadamente 0.2 kg de masa, tanto en la Tierra como en la Luna.

#### Peso:

- Es la fuerza con la que la gravedad atrae un objeto.
- Se mide en newtons (N).

- Cambia según la gravedad del lugar.
- Esa misma manzana pesa  $0.2 \times 9.8 = 1.96$  N en la Tierra, pero solo  $0.2 \times 1.6 = 0.32$  N en la Luna.

### **Fuerza:**

- Es cualquier interacción que cambia o tiende a cambiar el estado de movimiento de un objeto.
- Se mide en newtons (N).
- Puede ser de contacto (empujar, tirar) o a distancia (gravedad, magnetismo).

## **0.4.2 La Segunda Ley de Newton: la ecuación fundamental**

La segunda ley de Newton es probablemente la ecuación más importante en mecánica:

$$\mathbf{F} = \mathbf{m} \times \mathbf{a}$$

Donde:

- F = Fuerza (newtons, N)
- m = masa (kilogramos, kg)
- a = aceleración (metros/segundo<sup>2</sup>, m/s<sup>2</sup>)

**Interpretación:** Para acelerar un objeto, necesitas aplicar una fuerza. Cuanto más masivo sea el objeto, más fuerza necesitarás para acelerarlo.

**Ejemplo cotidiano:** Empujar un carro de supermercado:

- Carro vacío (m = 15 kg): lo aceleras fácilmente con poca fuerza.
- Carro lleno (m = 50 kg): necesitas mucha más fuerza para la misma aceleración.

**Ejemplo técnico:** Un robot debe mover una pieza de 20 kg con una aceleración de 2 m/s<sup>2</sup>:

$$F = 20 \times 2 = 40 \text{ N}$$

Pero además, el robot debe vencer la gravedad para levantar la pieza. La fuerza total necesaria es:

$$F_{\text{total}} = F_{\text{gravedad}} + F_{\text{aceleración}} = (20 \times 9.8) + (20 \times 2) = 196 + 40 = 236 \text{ N}$$

## **0.4.3 La gravedad: un caso especial**

La gravedad terrestre acelera todos los objetos hacia el centro de la Tierra a 9.8 m/s<sup>2</sup>. Esto significa que:

$$\mathbf{Peso} = \mathbf{masa} \times \mathbf{gravedad}$$

$$P = m \times g$$

Donde  $g = 9.8 \text{ m/s}^2$  (en la Tierra). En la mayoría de cálculos prácticos redondeamos  $g = 10 \text{ m/s}^2$  para simplificar.

### Ejemplos de peso:

- Una persona de 70 kg pesa:  $70 \times 9.8 = 686 \text{ N}$  ( $\approx 700 \text{ N}$  usando  $g=10$ )
- Un motor de 50 kg pesa:  $50 \times 9.8 = 490 \text{ N}$
- Una pieza de 5 kg pesa:  $5 \times 9.8 = 49 \text{ N}$

## 0.4.4 Aplicación real: dimensionado de actuadores

Cuando seleccionamos un actuador (motor, cilindro) para una aplicación, necesitamos conocer la fuerza que debe ejercer. Esta fuerza depende de:

1. **El peso de la carga** (masa  $\times$  gravedad)
2. **La aceleración requerida** (masa  $\times$  aceleración)
3. **La fricción** (depende de los materiales y superficies)
4. **Un factor de seguridad** (normalmente 1.5 a 2)

**Ejemplo completo:** Un brazo robótico debe levantar una pieza de 20 kg.

- Peso de la pieza:  $20 \times 10 = 200 \text{ N}$
- Aceleración de movimiento:  $3 \text{ m/s}^2 \rightarrow F_{\text{acel}} = 20 \times 3 = 60 \text{ N}$
- Fuerza total sin seguridad:  $200 + 60 = 260 \text{ N}$
- Factor de seguridad 1.5:  $260 \times 1.5 = 390 \text{ N}$
- El actuador debe proporcionar al menos 390 N de fuerza

Si usamos un cilindro neumático a 6 bar (600,000 Pa):

- Área necesaria:  $A = F \div P = 390 \div 600,000 = 0.00065 \text{ m}^2 = 6.5 \text{ cm}^2$
- Diámetro del pistón:  $r = \sqrt{(A \div \pi)} = \sqrt{(6.5 \div 3.1416)} = \sqrt{2.07} = 1.44 \text{ cm}$
- Diámetro  $\approx 29 \text{ mm} \rightarrow$  seleccionamos un cilindro de 32 mm (medida comercial)

## 0.4.5 Resumen

- **Masa (kg):** cantidad de materia — invariante
- **Peso (N):** fuerza gravitatoria — masa  $\times$  gravedad
- **Fuerza (N):** interacción que cambia el movimiento —  $F = m \times a$
- **Gravedad terrestre:**  $g = 9.8 \text{ m/s}^2$  (usamos 10 para cálculos rápidos)
- **Factor de seguridad:** multiplicar por 1.5-2 para garantizar fiabilidad

## 0.4.6 Ejercicios

**Ejercicio 1:** Un actuador lineal debe mover una carga de 30 kg con una aceleración de 3 m/s<sup>2</sup>. ¿Qué fuerza mínima debe generar? (Ignora gravedad si es movimiento horizontal)

(Solución:  $F = 30 \times 3 = 90 \text{ N}$ )

**Ejercicio 2:** Una plataforma elevadora debe levantar 500 kg a velocidad constante (aceleración despreciable). ¿Qué fuerza debe ejercer el cilindro hidráulico?

(Solución:  $F = 500 \times 10 = 5000 \text{ N} \approx 5000 \text{ N} \approx 510 \text{ kg-fuerza}$ )

**Ejercicio 3:** Una masa de 10 kg cae libremente. Calcula: a) Su peso en la Tierra, b) Su peso en la Luna ( $g=1.6 \text{ m/s}^2$ ), c) Su masa en ambos lugares.

(Soluciones: a)  $10 \times 9.8 = 98 \text{ N}$ ; b)  $10 \times 1.6 = 16 \text{ N}$ ; c) 10 kg en ambos)

\newpage

## CAPÍTULO 0.5: FÍSICA II — ENERGÍA Y MOVIMIENTO

### 0.5.1 Concepto de energía

La energía es uno de los conceptos más fundamentales de la física. En términos simples:

**Energía es la capacidad de realizar trabajo.**

La energía no se crea ni se destruye, solo se transforma de un tipo a otro. Esto se conoce como el principio de conservación de la energía.

**Tipos de energía relevantes en mecatrónica:**

Tipo de energía	Descripción	Ejemplo
Eléctrica	Movimiento de electrones	Batería, red eléctrica
Mecánica	Movimiento de objetos	Motor girando, pieza cayendo
Térmica	Calor	Pérdidas en motores, fricción
Neumática	Aire comprimido	Cilindro neumático
Hidráulica	Aceite a presión	Prensa hidráulica

Tipo de energía	Descripción	Ejemplo
Química	Enlaces entre moléculas	Batería, combustible
Cinética	Energía del movimiento	Objeto en movimiento
Potencial	Energía almacenada por posición	Objeto elevado

### 0.5.2 Trabajo mecánico

En física, “trabajo” tiene un significado muy específico:

**Trabajo = Fuerza × Distancia**

$$W = F \times d$$

Donde:

- W = Trabajo (julios, J)
- F = Fuerza (newtons, N)
- d = Distancia (metros, m)

**Ejemplo cotidiano:** Levantar una caja de 10 kg (peso = 100 N) a una altura de 2 metros:

$$W = 100 \times 2 = 200 \text{ J}$$

**Ejemplo técnico:** Un motor empuja una carga con una fuerza de 150 N a lo largo de 3 metros:

$$W = 150 \times 3 = 450 \text{ J}$$

### 0.5.3 Potencia mecánica

La potencia es la velocidad a la que se realiza el trabajo:

**Potencia = Trabajo ÷ Tiempo**

$$P = W \div t$$

Donde:

- P = Potencia (vatios, W)
- W = Trabajo (julios, J)
- t = Tiempo (segundos, s)

**Ejemplo:** Si un motor realiza 450 J de trabajo en 3 segundos:

$$P = 450 \div 3 = 150 \text{ W}$$

**Relación importante:** 1 caballo de vapor (CV) = 735.5 W ≈ 0.736 kW

### 0.5.4 Energía cinética y potencial

**Energía cinética:** la energía del movimiento

$$E_c = \frac{1}{2} \times m \times v^2$$

Donde m es la masa y v es la velocidad.

**Ejemplo:** Un carro de 50 kg que se mueve a 2 m/s:

$$E_c = 0.5 \times 50 \times 2^2 = 25 \times 4 = 100 \text{ J}$$

**Energía potencial gravitatoria:** la energía almacenada por la altura

$$E_p = m \times g \times h$$

Donde m es la masa, g la gravedad y h la altura.

**Ejemplo:** Una pieza de 20 kg elevada a 3 metros:

$$E_p = 20 \times 10 \times 3 = 600 \text{ J}$$

### 0.5.5 Aplicación real: selección de motores

Para seleccionar el motor de una cinta transportadora, seguimos estos pasos:

1. **Fuerza necesaria:** la carga ejerce una fuerza de fricción. Supongamos 200 N.
2. **Distancia:** longitud de la cinta, 5 metros.
3. **Tiempo por ciclo:** deben transportarse piezas cada 10 segundos.
4. **Trabajo por ciclo:**  $W = 200 \times 5 = 1000 \text{ J}$
5. **Potencia necesaria:**  $P = 1000 \div 10 = 100 \text{ W}$

A esto hay que añadir:

- Pérdidas por fricción en rodillos (~20%):  $100 \times 0.2 = 20 \text{ W}$
- Factor de seguridad (1.5):  $120 \times 1.5 = 180 \text{ W}$

Seleccionamos un motor de al menos 180 W (aproximadamente 0.25 CV).

### 0.5.6 Resumen

Magnitud	Fórmula	Unidad	Símbolo
Trabajo	$W = F \times d$	Julio	J
Potencia	$P = W \div t$	Vatio	W
Velocidad	$v = d \div t$	m/s	v

Magnitud	Fórmula	Unidad	Símbolo
Aceleración	$a = \Delta v \div t$	m/s <sup>2</sup>	a
E. cinética	$E_c = \frac{1}{2}mv^2$	Julio	J
E. potencial	$E_p = mgh$	Julio	J

### 0.5.7 Ejercicios

**Ejercicio 1:** Un motor levanta una carga de 50 kg a 3 metros de altura en 6 segundos. ¿Qué potencia desarrolla? ( $g = 10 \text{ m/s}^2$ )

(Solución:  $F = 50 \times 10 = 500 \text{ N}$ ;  $W = 500 \times 3 = 1500 \text{ J}$ ;  $P = 1500 \div 6 = 250 \text{ W}$ )

**Ejercicio 2:** Un vehículo de 1000 kg se mueve a 20 m/s. ¿Cuál es su energía cinética?

(Solución:  $E_c = 0.5 \times 1000 \times 20^2 = 500 \times 400 = 200,000 \text{ J} = 200 \text{ kJ}$ )

**Ejercicio 3:** Una grúa levanta una viga de 200 kg a 15 metros. ¿Qué trabajo realiza? ¿Qué energía potencial adquiere la viga?

(Solución:  $W = 200 \times 10 \times 15 = 30,000 \text{ J}$ ;  $E_p = \text{misma cantidad} = 30,000 \text{ J}$ )

\newpage

## CAPÍTULO 0.6: ELECTRICIDAD I — VOLTAJE Y CORRIENTE

### 0.6.1 ¿Qué es la electricidad?

La electricidad es el movimiento de electrones a través de un material conductor. Los electrones son partículas subatómicas con carga negativa que se mueven cuando hay una diferencia de potencial (voltaje).

#### Analogía hidráulica (la más útil para entender electricidad):

Imagina un circuito de agua cerrado:

- La **bomba** que empuja el agua es como una **fuentes de voltaje** (batería, pila).
- La **presión del agua** es el **voltaje** (V).
- La **cantidad de agua que fluye** es la **corriente** (I).
- Las **tuberías** son los **cables**.

- Los **obstáculos** en las tuberías son las **resistencias**.
- Los **motores de agua** que hacen trabajo son los **actuadores**.

Esta analogía te acompañará durante todo el libro. Cuando no entiendas un concepto eléctrico, pregúntate: “¿cómo sería esto en un sistema de agua?”.

### 0.6.2 Voltaje (V) — La presión eléctrica

**Definición:** El voltaje es la diferencia de potencial eléctrico entre dos puntos. Es la “presión” que empuja los electrones a través del circuito.

**Analogía:** Un tanque de agua elevado tiene presión por la altura del agua. Cuanto más alto está el tanque, más presión hay. El voltaje es esa presión: una batería de 9 V tiene más “presión” que una de 1.5 V.

**Medición:** Se mide con un voltímetro (o multímetro en modo voltaje). Se conecta en paralelo: las dos puntas del medidor tocan los dos puntos entre los que queremos medir la diferencia.

**Valores típicos:**

- Pila AA, AAA, C, D: 1.5 V
- Batería rectangular: 9 V
- Batería de coche: 12 V (realmente 12.6 V cuando está cargada)
- USB: 5 V
- Enchufe doméstico España: 230 V AC
- Línea de alta tensión: 132,000 V (132 kV)
- PLC industrial: 24 V DC

### 0.6.3 Corriente (I) — El flujo de electrones

**Definición:** La corriente eléctrica es la cantidad de electrones que fluyen por un punto del circuito en cada segundo.

**Analogía:** Es el caudal de agua que circula por una tubería. Cuanto más caudal, más electrones pasan por segundo.

**Medición:** Se mide con un amperímetro (o multímetro en modo corriente). ¡Importante! Se conecta en serie: hay que interrumpir el circuito y colocar el medidor en medio.

**Valores típicos:**

- LED pequeño: 0.02 A (20 miliamperios, mA)
- Bombilla LED doméstica: 0.1-0.3 A

- Motor de taladro: 2-5 A
- Motor industrial: 10-100 A
- Arranque de coche: 100-300 A (momentáneo)

#### 0.6.4 La relación voltaje-corriente

Voltaje y corriente están relacionados por la resistencia del circuito. Con la misma resistencia:

- **Más voltaje** → **más corriente** (más presión → más caudal)
- **Menos voltaje** → **menos corriente** (menos presión → menos caudal)

**Ejemplo cotidiano:** Una manguera de jardín:

- Abres el grifo poco (poco voltaje) → sale poca agua (poca corriente)
- Abres el grifo mucho (mucho voltaje) → sale mucha agua (mucha corriente)
- Si pones el dedo en la salida (añades resistencia) → sale menos agua

#### 0.6.5 Corriente continua (DC) vs corriente alterna (AC)

**Corriente continua (DC):** Los electrones fluyen siempre en la misma dirección.

**Características:**

- Voltaje constante en el tiempo
- La proporcionan pilas, baterías, fuentes de alimentación
- Se usa en electrónica, PLC, sensores
- Polaridad definida (+ y -)

**Corriente alterna (AC):** Los electrones cambian de dirección periódicamente.

**Características:**

- Voltaje variable (forma de onda sinusoidal)
- La proporciona la red eléctrica
- Se usa en motores grandes, iluminación, electrodomésticos
- No tiene polaridad fija (los dos cables son fase y neutro)
- Frecuencia: 50 Hz en Europa (50 ciclos por segundo)

#### 0.6.6 Aplicación real

Los PLC industriales trabajan con 24 V DC por varias razones:

1. **Seguridad:** 24 V es baja tensión, no peligrosa para las personas.
2. **Fiabilidad:** Suficientemente alta para no perderse por interferencias eléctricas.

3. **Estandarización:** Es el voltaje más común en automatización industrial.
4. **Compatibilidad:** La mayoría de sensores y actuadores industriales usan 24 V.

### 0.6.7 Resumen

Concepto	Analogía	Unidad	Símbolo	Instrumento
Voltaje	Presión del agua	Voltio (V)	V	Voltímetro (en paralelo)
Corriente	Caudal de agua	Amperio (A)	I	Amperímetro (en serie)

- El voltaje **empuja** los electrones
- La corriente es el **flujo** de electrones
- Necesitas ambos para que funcione un circuito
- Más voltaje → más corriente (con la misma resistencia)
- DC: continua (baterías); AC: alterna (red eléctrica)

### 0.6.8 Ejercicios

**Ejercicio 1:** Una fuente de alimentación tiene dos ajustes: 5 V y 12 V. Si conectamos la misma resistencia a cada ajuste, ¿cuál hará circular más corriente? ¿Por qué?

(Solución: 12 V, porque hay más presión eléctrica empujando los electrones)

**Ejercicio 2:** Un motor DC funciona a 12 V y consume 3 A. ¿Qué corriente consumiría si lo conectáramos a 6 V (suponiendo resistencia constante)?

(Solución: Como  $V = I \times R$ , la resistencia es  $R = 12 \div 3 = 4 \Omega$ . A 6 V:  $I = 6 \div 4 = 1.5$  A. La corriente se reduce a la mitad.)

**Ejercicio 3:** ¿Qué voltaje necesitamos para que circulen 5 A por una resistencia de 10  $\Omega$ ?

(Solución:  $V = 5 \times 10 = 50$  V)

\newpage

## CAPÍTULO 0.7: ELECTRICIDAD II — RESISTENCIA Y LEY DE OHM

### 0.7.1 ¿Qué es la resistencia eléctrica?

La resistencia es la oposición al paso de la corriente eléctrica. Todos los materiales ofrecen cierta resistencia, aunque en algunos es extremadamente baja (conductores) o extremadamente alta (aislantes).

**Unidad:** Ohmio ( $\Omega$ )

**Analogía:** Piensa en una tubería de agua:

- Tubería ancha = baja resistencia → pasa mucha agua (buen conductor)
- Tubería estrecha = alta resistencia → pasa poca agua (mal conductor)
- Tubería completamente cerrada = resistencia infinita → no pasa agua (aislante)

### 0.7.2 Clasificación de materiales según su resistencia

Material	Tipo	Resistividad	Ejemplo de uso
Plata	Conductor	Muy baja	Contactos eléctricos de alta calidad
Cobre	Conductor	Muy baja	Cables eléctricos (el más usado)
Oro	Conductor	Baja	Contactos de conectores (no se oxida)
Aluminio	Conductor	Baja	Cables de alta tensión (ligero)
Hierro	Conductor	Media	Motores, transformadores
Carbono	Semiconductor	Media	Resistencias, escobillas de motor
Silicio	Semiconductor	Media-alta	Chips, transistores
Agua pura	Aislante	Muy alta	No conduce electricidad
Plástico	Aislante	Altísima	Recubrimiento de cables
Caucho	Aislante	Altísima	Guantes de seguridad
Vidrio	Aislante	Altísima	Aislantes de alta tensión
Aire	Aislante	Altísima	Entre conductores desnudos

### 0.7.3 Factores que afectan la resistencia de un cable

La resistencia de un cable depende de cuatro factores:

- 1. Material:** Cada material tiene una resistividad característica ( $\rho$ ).
- 2. Longitud (L):** A mayor longitud, mayor resistencia ( $R \propto L$ ).
- 3. Sección (S):** A mayor sección (más grueso), menor resistencia ( $R \propto 1/S$ ).
- 4. Temperatura (T):** Al calentarse, la resistencia de los metales aumenta.

**Fórmula:**  $R = \rho \times L \div S$

**Ejemplo práctico:** Un cable de cobre ( $\rho = 0.017 \Omega \cdot \text{mm}^2/\text{m}$ ) de 10 metros de largo y  $1.5 \text{ mm}^2$  de sección:

$$R = 0.017 \times 10 \div 1.5 = 0.17 \div 1.5 = 0.113 \Omega$$

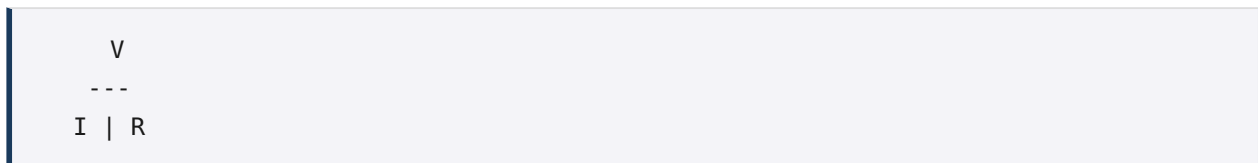
Esta resistencia tan baja es la razón por la que usamos cobre para los cables: ofrece muy poca oposición al paso de la corriente.

### 0.7.4 La Ley de Ohm — La fórmula más importante de la electricidad

$$V = I \times R$$

Esta es, sin duda, la fórmula que más usarás en toda tu carrera como técnico en mecatrónica. Relaciona las tres magnitudes fundamentales de la electricidad.

**El triángulo mágico:**



**Cómo usarlo:**

1. Tapa la variable que quieres calcular
2. Lo que queda es la fórmula

- Tapa V  $\rightarrow V = I \times R$
- Tapa I  $\rightarrow I = V \div R$
- Tapa R  $\rightarrow R = V \div I$

### 0.7.5 Ejemplos prácticos de la Ley de Ohm

**Ejemplo 1: Calcular resistencia necesaria para un LED**

Un LED necesita 2 V y 20 mA (0.02 A) para funcionar. Lo conectamos a una fuente de 5 V.

Primero calculamos el voltaje que debe “sobrar” en la resistencia:

$$V_R = 5 - 2 = 3 \text{ V}$$

Luego calculamos la resistencia:

$$R = V_R \div I = 3 \div 0.02 = 150 \Omega$$

### Ejemplo 2: Calcular corriente en un circuito

Tenemos una resistencia de  $1000 \Omega$  ( $1 \text{ k}\Omega$ ) conectada a  $12 \text{ V}$ :

$$I = V \div R = 12 \div 1000 = 0.012 \text{ A} = 12 \text{ mA}$$

### Ejemplo 3: Calcular voltaje en una resistencia

Por una resistencia de  $470 \Omega$  circulan  $0.05 \text{ A}$  ( $50 \text{ mA}$ ):

$$V = 0.05 \times 470 = 23.5 \text{ V}$$

## 0.7.6 Aplicación real: señal 4-20 mA en sensores industriales

Los sensores industriales suelen transmitir su señal mediante el estándar 4-20 mA. La corriente varía entre  $4 \text{ mA}$  (valor mínimo de la variable medida) y  $20 \text{ mA}$  (valor máximo).

Para convertir esta corriente a voltaje que pueda leer un PLC, se coloca una resistencia de precisión de  $250 \Omega$ :

Corriente	Voltaje ( $V = I \times 250$ )	Significado
$4 \text{ mA}$	$4 \times 0.250 = 1 \text{ V}$	Valor mínimo (ej: $0 \text{ }^\circ\text{C}$ )
$12 \text{ mA}$	$12 \times 0.250 = 3 \text{ V}$	Valor medio (ej: $50 \text{ }^\circ\text{C}$ )
$20 \text{ mA}$	$20 \times 0.250 = 5 \text{ V}$	Valor máximo (ej: $100 \text{ }^\circ\text{C}$ )

La ventaja de usar corriente en lugar de voltaje es que la señal no se degrada con la distancia: 4-20 mA puede viajar cientos de metros sin pérdida apreciable.

## 0.7.7 Resumen

- **Ley de Ohm:**  $V = I \times R$
- **Unidad de resistencia:** Ohmio ( $\Omega$ )
- **Conductores:** baja resistencia (cobre, oro, plata)
- **Aislantes:** alta resistencia (plástico, goma, vidrio)
- **Semiconductores:** resistencia controlable (silicio, carbono)

- **Señal 4-20 mA:** estándar industrial, se convierte a voltaje con resistencia de  $250 \Omega$

## 0.7.8 Ejercicios

**Ejercicio 1:** Un sensor de temperatura entrega 12 V a través de una resistencia de  $300 \Omega$ . ¿Qué corriente circula?

(Solución:  $I = 12 \div 300 = 0.04 \text{ A} = 40 \text{ mA}$ )

**Ejercicio 2:** ¿Qué resistencia necesitamos para limitar a 2 A la corriente en un circuito de 24 V?

(Solución:  $R = 24 \div 2 = 12 \Omega$ )

**Ejercicio 3:** Un cable de cobre de 50 m y  $2.5 \text{ mm}^2$  de sección ( $\rho=0.017$ ). ¿Qué resistencia tiene? ¿Qué corriente circulará con 12 V?

(Solución:  $R = 0.017 \times 50 \div 2.5 = 0.34 \Omega$ ;  $I = 12 \div 0.34 = 35.3 \text{ A}$ . ¡Cuidado! 35 A es mucha corriente para ese cable, se calentaría peligrosamente.)

\newpage

## CAPÍTULO 0.8: POTENCIA Y ENERGÍA ELÉCTRICA

### 0.8.1 ¿Qué es la potencia eléctrica?

La potencia eléctrica es la velocidad a la que se consume o produce energía eléctrica. Es decir, cuánta energía se gasta por unidad de tiempo.

**Fórmula fundamental:**  $P = V \times I$

Donde:

- P = Potencia en vatios (W)
- V = Voltaje en voltios (V)
- I = Corriente en amperios (A)

#### **Ejemplos cotidianos:**

- Una bombilla LED de 10 W consume menos que una incandescente de 60 W dando la misma luz
- Un microondas de 1000 W calienta rápido (muchísima potencia)
- Un cargador de móvil típico:  $5 \text{ V} \times 2 \text{ A} = 10 \text{ W}$
- Una plancha: 2000 W

## 0.8.2 Formas alternativas de calcular potencia

A partir de la Ley de Ohm ( $V = I \times R$ ), podemos derivar dos fórmulas adicionales:

**$P = I^2 \times R$**  — útil cuando conocemos corriente y resistencia

**$P = V^2 \div R$**  — útil cuando conocemos voltaje y resistencia

### Ejemplo 1 ( $P = I^2 \times R$ ):

Una resistencia de 100  $\Omega$  con 0.5 A:

$$P = 0.5^2 \times 100 = 0.25 \times 100 = 25 \text{ W}$$

Esto significa que la resistencia disipa 25 W en forma de calor. Necesitamos una resistencia físicamente grande que pueda soportar esa potencia sin quemarse.

### Ejemplo 2 ( $P = V^2 \div R$ ):

Una resistencia de 100  $\Omega$  conectada a 12 V:

$$P = 12^2 \div 100 = 144 \div 100 = 1.44 \text{ W}$$

Usaríamos una resistencia de 2 W (margen de seguridad).

## 0.8.3 Potencia en los componentes

Cada componente eléctrico tiene una potencia máxima que puede disipar. Superarla significa destruir el componente.

**Resistencias:** vienen en potencias normalizadas: 1/8 W, 1/4 W, 1/2 W, 1 W, 2 W, 5 W, 10 W...

**LEDs:** Los LEDs típicos consumen unos 20 mA a 2-3 V, lo que da unos 40-60 mW (0.04-0.06 W).

**Motores:** La potencia nominal indica cuánta potencia puede entregar el motor de forma continua sin sobrecalentarse.

## 0.8.4 Energía eléctrica

La energía es la potencia consumida durante un tiempo:

**Energía = Potencia  $\times$  tiempo**

**Unidades:**

- Vatio-segundo (Ws) = Julio (J)

- Vatio-hora (Wh): 1 Wh = 3600 J

- Kilovatio-hora (kWh): 1 kWh = 1000 Wh (es la unidad que aparece en tu factura de luz)

**Ejemplo:** Un motor de 500 W funcionando 2 horas:

$$\text{Energía} = 500 \times 2 = 1000 \text{ Wh} = 1 \text{ kWh}$$

**Cálculo de coste:**

Si el kWh cuesta 0.12 €:

$$\text{Coste} = 1 \times 0.12 = 0.12 \text{ € por cada 2 horas de funcionamiento}$$

### 0.8.5 Aplicación real: cálculo de consumo industrial

Una fábrica tiene 10 motores de 2 kW (2000 W) que funcionan 8 horas al día.

**Paso 1: Potencia total**

$$10 \times 2000 = 20,000 \text{ W} = 20 \text{ kW}$$

**Paso 2: Energía consumida al día**

$$20 \times 8 = 160 \text{ kWh}$$

**Paso 3: Coste diario**

$$160 \times 0.12 \text{ €/kWh} = 19.20 \text{ €}$$

**Paso 4: Coste mensual (22 días laborables)**

$$19.20 \times 22 = 422.40 \text{ €}$$

**Paso 5: Coste anual**

$$422.40 \times 12 = 5,068.80 \text{ €}$$

Este cálculo muestra por qué la eficiencia energética es tan importante en la industria. Un motor con 5% más de eficiencia ahorraría más de 250 € al año en este ejemplo.

### 0.8.6 Ley de Joule: el calentamiento de los conductores

Cuando la corriente circula por un conductor, este se calienta. La cantidad de calor generado es:

$$Q = I^2 \times R \times t$$

Donde Q es el calor en julios, I la corriente, R la resistencia y t el tiempo.

**Implicaciones prácticas:**

- Los cables deben tener la sección adecuada para la corriente que van a transportar
- Los motores necesitan ventilación para disipar el calor

- Los disipadores en componentes electrónicos evitan el sobrecalentamiento
- Los cuadros eléctricos necesitan ventilación

### 0.8.7 Resumen

Fórmula	Significado	Cuándo usarla
$P = V \times I$	Potencia básica	Cuando conoces voltaje y corriente
$P = I^2 \times R$	Potencia por corriente	Cuando conoces I y R (calentamiento)
$P = V^2 \div R$	Potencia por voltaje	Cuando conoces V y R
$E = P \times t$	Energía consumida	Para calcular consumo eléctrico

### 0.8.8 Ejercicios

**Ejercicio 1:** Una resistencia tiene 24 V y circulan 2 A. a) ¿Qué potencia disipa? b) ¿Qué energía consume en 3 horas?

(Solución: a)  $P = 24 \times 2 = 48 \text{ W}$ ; b)  $E = 48 \times 3 = 144 \text{ Wh}$ )

**Ejercicio 2:** Un motor de 1.5 kW funciona 6 horas diarias durante 22 días al mes. ¿Cuántos kWh consume al mes? ¿Cuánto cuesta a 0.15 €/kWh?

(Solución:  $1.5 \times 6 \times 22 = 198 \text{ kWh/mes}$ ;  $198 \times 0.15 = 29.70 \text{ €/mes}$ )

**Ejercicio 3:** Una resistencia de 47  $\Omega$  se conecta a 12 V. Calcula la corriente y la potencia disipada.

(Solución:  $I = 12 \div 47 = 0.255 \text{ A} = 255 \text{ mA}$ ;  $P = 12^2 \div 47 = 144 \div 47 = 3.06 \text{ W}$ . Necesitamos una resistencia de al menos 5 W.)

\newpage

---

# NIVEL 1: BASE DE MECATRÓNICA

---

*“Ya tienes los cimientos. Ahora empezamos a construir.”*

En este nivel aprenderás cómo funcionan los componentes mecánicos y electrónicos básicos, y cómo sensores y actuadores convierten señales en movimiento. Estos son los “bloques Lego” de cualquier sistema mecatrónico.

---

## CAPÍTULO 1.1: INTRODUCCIÓN A LA MECÁNICA

### 1.1.1 ¿Qué es la mecánica?

La mecánica es la rama de la física que estudia el movimiento y el equilibrio de los cuerpos, así como las fuerzas que los producen. En el contexto de la mecatrónica, la mecánica nos dice:

- Cómo se mueven las piezas de una máquina
- Cómo se transmiten las fuerzas a través de los componentes
- Cómo se mantienen estables las estructuras
- Cómo se convierten los tipos de movimiento

### 1.1.2 Conceptos fundamentales

**Máquina simple:** Es un dispositivo mecánico que cambia la magnitud o la dirección de una fuerza. Las máquinas simples son los bloques básicos de los que se componen todas las máquinas más complejas.

Las seis máquinas simples clásicas son:

1. **Palanca:** barra que gira sobre un punto de apoyo
2. **Rueda y eje:** rueda fijada a un eje central
3. **Polea:** rueda acanalada por la que pasa una cuerda
4. **Plano inclinado:** superficie inclinada para mover cargas
5. **Cuña:** plano inclinado doble (hacha, cincel)
6. **Tornillo:** plano inclinado enrollado

**Transmisión:** Sistema que lleva el movimiento de un punto a otro. Las transmisiones pueden cambiar la velocidad, la dirección o el tipo de movimiento.

**Mecanismo:** Conjunto organizado de piezas que trabajan juntas para realizar una función específica. Por ejemplo, el mecanismo de un reloj convierte la energía del muelle en movimiento de las agujas.

### 1.1.3 La importancia de la mecánica en mecatrónica

Un robot industrial, una cinta transportadora, una máquina CNC o un ascensor son sistemas mecánicos movidos por componentes eléctricos y controlados por electrónica. Para diseñarlos, mantenerlos o repararlos, es imprescindible entender la parte mecánica.

Los fallos más comunes en sistemas automatizados son de origen mecánico: desgaste de engranajes, rotura de correas, atasco de cilindros, pérdida de precisión por holguras. Un buen técnico en mecatrónica debe saber diagnosticar y solucionar estos problemas.

### 1.1.4 Ejemplo real: brazo robótico

Un brazo robótico típico tiene varias articulaciones (como el hombro, codo y muñeca humanos). Cada articulación es un mecanismo que permite un tipo específico de movimiento:

- **Base:** rotación horizontal (como girar sobre los pies)
- **Hombro:** elevación del brazo (como levantar el brazo)
- **Codo:** extensión del antebrazo (como estirar el brazo)
- **Muñeca:** rotación, inclinación y giro de la herramienta

Cada articulación está movida por un servomotor con engranajes que multiplican el par. El controlador envía las órdenes a los motores, y los sensores (encoders) devuelven la posición exacta de cada articulación.

### 1.1.5 Aplicación práctica en la industria

En una fábrica de automóviles, los robots soldadores deben posicionar el soldador con precisión de  $\pm 0.5$  mm. Esto requiere:

- Estructura mecánica rígida que no se deforme con las cargas
- Transmisiones sin holgura (engranajes de precisión)

- Movimiento suave y controlado (servomotores)
- Sensores que verifiquen la posición constantemente

### 1.1.6 Resumen

- La mecánica estudia el movimiento, las fuerzas y el equilibrio
- Las máquinas simples son los bloques básicos de toda la mecánica
- Los mecanismos transmiten y transforman movimiento
- En mecatrónica, los sistemas mecánicos se combinan con electrónica y control

### 1.1.7 Ejercicio

Identifica 3 máquinas simples en tu casa y explica qué tipo son y para qué sirven.

Ejemplo: Las tijeras son una palanca de 1º género (el fulcro está entre la fuerza y la resistencia, que son los mangos y las hojas).

\newpage

## CAPÍTULO 1.2: TIPOS DE MOVIMIENTO

### 1.2.1 Movimiento lineal

El objeto se mueve en línea recta, de un punto a otro, sin desviarse.

#### **Características:**

- Trayectoria rectilínea
- Puede ser horizontal, vertical o inclinado
- Se mide en metros de desplazamiento

#### **Ejemplos cotidianos:**

- Un ascensor subiendo o bajando
- Un pistón de jeringa
- Un cajón que se abre
- La aguja de una máquina de coser

#### **Ejemplos técnicos en mecatrónica:**

- El vástago de un cilindro neumático que empuja una pieza
- El carro de una impresora 3D moviéndose en el eje X
- La mesa de una fresadora CNC desplazándose

### 1.2.2 Movimiento rotatorio

El objeto gira alrededor de un eje fijo. Es el tipo de movimiento más común en máquinas.

#### **Características:**

- Trayectoria circular
- Se mide en revoluciones por minuto (RPM) o radianes por segundo
- El par (torque) es la fuerza de giro

#### **Ejemplos cotidianos:**

- Una rueda de bicicleta girando
- Un ventilador de techo
- El tambor de una lavadora

#### **Ejemplos técnicos en mecatrónica:**

- El eje de un motor eléctrico
- Un engranaje transmitiendo movimiento
- La polea de una cinta transportadora

### 1.2.3 Movimiento alternativo (lineal recíproco)

Es un movimiento lineal que va y viene entre dos puntos. Esencialmente, un movimiento lineal que cambia de dirección periódicamente.

#### **Características:**

- Trayectoria lineal en dos direcciones
- Tiene un punto de avance máximo y un punto de retroceso máximo
- La frecuencia indica cuántos ciclos completos por segundo

#### **Ejemplos cotidianos:**

- Un pistón de motor de coche
- Una sierra de vaivén
- Un péndulo (aunque es curvo, el concepto es similar)
- El mango de una bomba de bicicleta

#### **Ejemplos técnicos:**

- Un cilindro neumático de doble efecto avanzando y retrocediendo
- El movimiento de una prensa mecánica
- El pistón de una bomba hidráulica

### 1.2.4 Conversión entre tipos de movimiento

En mecatrónica es muy común convertir un tipo de movimiento en otro. De hecho, la mayoría de los mecanismos existen precisamente para hacer estas conversiones.

Conversión	Mecanismo	Ejemplo real
Rotatorio → Lineal	Cremallera y piñón	Dirección de coche
Rotatorio → Lineal	Husillo y tuerca	CNC, impresora 3D
Rotatorio → Lineal	Manivela-biela	Motor de combustión
Rotatorio → Alternativo	Leva y seguidor	Máquina de coser
Lineal → Rotatorio	Cremallera y piñón	Sacacorchos
Lineal → Rotatorio	Cigüeñal-biela	Arranque manual de motor

#### Ejemplo detallado: Máquina CNC

Una máquina CNC (Control Numérico Computarizado) tiene tres ejes (X, Y, Z). Cada eje está movido por un servomotor que gira (movimiento rotatorio). Este giro se convierte en movimiento lineal mediante un husillo de bolas (tornillo de precisión con rodamientos). El resultado: el motor giratorio produce un desplazamiento lineal del carro con precisión de centésimas de milímetro.

### 1.2.5 Aplicación real

Una máquina de inyección de plástico tiene un husillo que gira (rotatorio) para fundir y mezclar el plástico, y también se desplaza linealmente para inyectar el material en el molde. El mismo componente realiza dos tipos de movimiento según la fase del proceso.

### 1.2.6 Resumen

Tipo de movimiento	Descripción	Ejemplo industrial
Lineal	Línea recta	Cilindro neumático empujando piezas
Rotatorio	Giro alrededor de un eje	Motor eléctrico
Alternativo	Vaivén lineal	Pistón de prensa
Conversión	Cambio de un tipo a otro	Husillo (rotatorio → lineal)

## 1.2.7 Ejercicios

**Ejercicio 1:** Identifica qué tipo de movimiento tiene: a) Una cinta transportadora (la banda), b) Un taladro (la broca), c) Una prensa hidráulica (el pistón).

(Soluciones: a) Lineal, b) Rotatorio, c) Lineal/Alternativo)

**Ejercicio 2:** ¿Qué mecanismo usarías para convertir el movimiento rotatorio de un motor en movimiento lineal para abrir una puerta corredera?

(Solución: Cremallera y piñón, o husillo y tuerca)

\newpage

## CAPÍTULO 1.3: ENGRANAJES Y POLEAS

### 1.3.1 Engranajes: ruedas dentadas que transmiten potencia

Los engranajes son ruedas con dientes que encajan entre sí para transmitir movimiento rotatorio de un eje a otro. Son los componentes mecánicos más importantes después de los propios motores.

**Principio básico:** Cuando dos engranajes están acoplados, giran en direcciones opuestas. Si hay tres: el primero y el tercero giran en la misma dirección.

### 1.3.2 Relación de transmisión

La relación de transmisión ( $i$ ) es el cociente entre el número de dientes del engranaje conducido y el del conductor:

$$i = Z_2 \div Z_1$$

Donde:

- $Z_2$  = dientes del engranaje conducido (el que recibe el movimiento)
- $Z_1$  = dientes del engranaje conductor (el que da el movimiento)

#### Efectos de la relación:

Relación	Velocidad	Par (fuerza)	Ejemplo
$i > 1$ (reductora)	Disminuye	Aumenta	Bicicleta: plato pequeño, piñón grande → cuesta arriba

Relación	Velocidad	Par (fuerza)	Ejemplo
$i < 1$ (multiplicadora)	Aumenta	Disminuye	Bicicleta: plato grande, piñón pequeño → llano
$i = 1$ (directa)	Igual	Igual	Acoplamiento directo

### Ejemplo completo:

Un engranaje de 10 dientes (conductor) mueve uno de 40 dientes (conducido):

-  $i = 40 \div 10 = 4$

- Esto significa que por cada 4 vueltas del pequeño, el grande da 1 vuelta

- La velocidad se reduce a 1/4 (si el pequeño gira a 1000 RPM, el grande gira a 250 RPM)

- El par se multiplica por 4 (más fuerza de giro)

### Fórmula completa:

$n_2 = n_1 \div i$  (velocidad)

$M_2 = M_1 \times i$  (par)

Donde n es velocidad y M es par.

### 1.3.3 Tipos de engranajes

Tipo	Imagen conceptual	Característica principal	Aplicación
Rectos	Dientes paralelos al eje	Simples, económicos, ruidosos	Transmisiones donde el ruido no importa
Helicoidales	Dientes inclinados	Silenciosos, más resistencia	Cajas de cambio de coches
Cónicos	Forma de cono	Cambian dirección 90°	Diferencial de automóvil
Sin fin	Tornillo + rueda	Gran reducción, no reversible	Cintas transportadoras, ascensores
Planetarios	Varios engranajes alrededor de uno central	Muy compactos, alta reducción	Transmisiones automáticas, robots

### 1.3.4 Poleas y correas

Las poleas transmiten movimiento entre ejes separados mediante una correa (banda). Son la alternativa a los engranajes cuando los ejes están distantes.

#### **Ventajas de poleas vs engranajes:**

- Más silenciosas
- Absorben vibraciones
- No necesitan lubricación
- Permiten ejes más separados
- Más económicas

#### **Desventajas:**

- Pueden patinar (correas lisas)
- Ocupan más espacio
- Menos precisas en la transmisión

#### **Tipos de correas:**

- **Planas:** simples, baja transmisión de par
- **Trapezoidales (V):** más agarre, las más comunes
- **Dentadas (sincrónicas):** no patinan, precisas (como las de distribución del coche)

#### **Relación de transmisión en poleas:**

$$i = D_2 \div D_1$$

Donde  $D_2$  y  $D_1$  son los diámetros de las poleas.

### 1.3.5 Aplicación real: robot industrial

Los robots industriales usan reductores en sus articulaciones. Un motor eléctrico puede girar a 3000 RPM con poco par. Mediante un reductor (engranajes planetarios o armónicos), se reduce la velocidad a 30 RPM y se multiplica el par por 100. Esto permite que un motor pequeño mueva un brazo robótico que levanta 100 kg.

### 1.3.6 Resumen

- **Engranajes:** ruedas dentadas que transmiten movimiento rotatorio
- **Relación de transmisión:**  $i = Z_2/Z_1$  (dientes) o  $i = D_2/D_1$  (diámetros)
- **$i > 1$ :** reduce velocidad, aumenta par (reductora)

- **$i < 1$** : aumenta velocidad, reduce par (multiplicadora)
- **Poleas**: alternativa a engranajes para ejes separados

### 1.3.7 Ejercicios

**Ejercicio 1:** Un motor gira a 3000 RPM con un engranaje de 20 dientes acoplado a otro de 60 dientes. ¿A qué velocidad gira el segundo engranaje?

(Solución:  $i = 60 \div 20 = 3$ ;  $n_2 = 3000 \div 3 = 1000$  RPM)

**Ejercicio 2:** La polea de un motor (10 cm de diámetro) mueve la polea de una bomba (25 cm de diámetro). Si el motor gira a 1450 RPM, ¿a qué velocidad gira la bomba?

(Solución:  $i = 25 \div 10 = 2.5$ ;  $n_2 = 1450 \div 2.5 = 580$  RPM)

**Ejercicio 3:** Necesitas que un eje gire a 500 RPM usando un motor de 3000 RPM. ¿Qué relación de transmisión necesitas?

(Solución:  $i = 3000 \div 500 = 6$ . Necesitas un engranaje 6 veces más grande en el eje conducido)

\newpage

## CAPÍTULO 1.4: PALANCAS, EJES Y MECANISMOS

### 1.4.1 La palanca

Una palanca es una barra rígida que puede girar alrededor de un punto de apoyo llamado fulcro. Es la máquina simple más intuitiva y una de las más útiles.

**Ley de la palanca:**

$$F \times d_1 = R \times d_2$$

Donde:

- F = Fuerza aplicada
- $d_1$  = Distancia de la fuerza al fulcro
- R = Resistencia (carga)
- $d_2$  = Distancia de la resistencia al fulcro

**Ejemplo:** Una palanca de 1 metro con el fulcro a 20 cm de una carga de 100 kg:

- Brazo de resistencia: 20 cm

- Brazo de fuerza: 80 cm (1 m - 20 cm)
- $F \times 80 = 100 \times 20$  (transformamos kg a N:  $100 \times 9.8 = 980$  N)
- $F \times 80 = 19600$
- $F = 19600 \div 80 = 245$  N  $\approx$  25 kg-fuerza

Hemos multiplicado nuestra fuerza por 4 (ventaja mecánica = 4).

### Tipos de palanca según la posición del fulcro:

Tipo	Orden	Ejemplo	Ventaja mecánica
1º género	Fulcro entre F y R	Balancín, tijeras, alicates	Variable
2º género	R entre F y fulcro	Carretilla, cascanueces	Siempre > 1 (multiplica fuerza)
3º género	F entre fulcro y R	Pinzas, caña de pescar	Siempre < 1 (aumenta velocidad)

### 1.4.2 Ejes y acoplamientos

**Ejes:** Son barras cilíndricas que transmiten movimiento rotatorio y par entre componentes. Pueden ser:

- **Ejes de transmisión:** transmiten potencia (línea de ejes)
- **Ejes de máquina:** soportan componentes giratorios (como en un torno)

**Acoplamientos:** Dispositivos que conectan dos ejes entre sí:

Tipo	Característica	Uso
Rígido	Conexión fija, requiere alineación perfecta	Unión directa motor-bomba
Flexible	Permite pequeñas desalineaciones	Motor-reductor
Cardan (Junta universal)	Permite grandes ángulos	Transmisión de coche
Elástico	Absorbe vibraciones	Motor-compresor
Embrague	Permite conectar/desconectar	Máquina herramienta

### 1.4.3 Mecanismos comunes en mecatrónica

Mecanismo	Función	Ejemplo
Manivela-biela	Convierte rotatorio en alternativo	Motor de coche, prensa
Leva-seguidor	Convierte rotatorio en lineal programado	Máquina de coser, motor de válvulas
Trinquete	Permite giro en un solo sentido	Llave de carraca, freno de mano
Cremallera	Convierte rotatorio en lineal	Dirección de coche, sacacorchos
Husillo-tuerca	Convierte rotatorio en lineal preciso	CNC, impresora 3D, tornillo de banco
Biela-manivela	Convierte alternativo en rotatorio	Motor de combustión

#### La leva en detalle:

Una leva es un disco con forma no circular que gira. Al girar, empuja un seguidor (una pieza que está en contacto con la leva) produciendo un movimiento lineal. La forma del perfil de la leva determina exactamente cómo se mueve el seguidor: con qué aceleración, a qué velocidad y durante qué tiempo. Esto permite programar movimientos complejos sin electrónica.

### 1.4.4 Aplicación real

Las máquinas de envasado usan levas para sellar paquetes: la leva convierte el giro constante de un motor en el movimiento de apertura y cierre de las mordazas de sellado. El perfil de la leva está diseñado para que las mordazas permanezcan cerradas el tiempo justo para sellar el plástico.

### 1.4.5 Resumen

- **Palanca:** multiplica fuerza según la posición del fulcro
- **Ejes:** transmiten rotación entre componentes
- **Acoplamientos:** conectan ejes (rígidos, flexibles, cardan)
- **Mecanismos:** convierten tipos de movimiento (leva, cremallera, husillo, biela-manivela)

### 1.4.6 Ejercicios

**Ejercicio 1:** Una palanca de 1.2 m tiene el fulcro a 30 cm de la carga de 150 kg. ¿Qué fuerza mínima se necesita en el otro extremo?

(Solución: Brazo de carga = 30 cm, Brazo de fuerza = 120 - 30 = 90 cm.  $F \times 90 = 150 \times 30$ ;  $F = 4500 \div 90 = 50$  kg)

**Ejercicio 2:** ¿Qué mecanismo usarías para mover el carro de una impresora 3D con un motor paso a paso?

(Solución: Husillo y tuerca, o correa dentada con poleas)

\newpage

## CAPÍTULO 1.5: COMPONENTES ELECTRÓNICOS BÁSICOS I

### 1.5.1 Resistencias: el componente más común

Una resistencia es un componente que se opone al paso de la corriente eléctrica. Es, junto con los cables, el componente más utilizado en electrónica.

**Símbolo:**  (zigzag) o  (rectángulo)

**Función principal:** Limitar la corriente para proteger otros componentes. Por ejemplo, un LED sin resistencia se quemaría en segundos.

#### Código de colores:

Las resistencias tienen bandas de colores que indican su valor. El código se lee de izquierda a derecha:

Color	Dígito	Multiplicador	Tolerancia
Negro	0	$\times 1$	-
Marrón	1	$\times 10$	$\pm 1\%$
Rojo	2	$\times 100$	$\pm 2\%$
Naranja	3	$\times 1000$	-
Amarillo	4	$\times 10000$	-
Verde	5	$\times 100000$	$\pm 0.5\%$
Azul	6	$\times 1000000$	-

Color	Dígito	Multiplicador	Tolerancia
Violeta	7	-	-
Gris	8	-	-
Blanco	9	-	-
Dorado	-	×0.1	±5%
Plateado	-	×0.01	±10%


**Lectura práctica:** Rojo (2) - Violeta (7) - Naranja (×1000) - Dorado (±5%)  
= 27 × 1000 = 27,000 Ω = 27 kΩ con tolerancia ±5%

### Montaje físico:

Las resistencias tienen dos terminales (no tienen polaridad, se pueden conectar en cualquier dirección). Se montan en la placa insertando los terminales en los agujeros y soldándolos.

### 1.5.2 Condensadores (capacitores)

Un condensador almacena energía eléctrica en forma de campo eléctrico. Es como una batería que se carga y descarga muy rápidamente.

**Símbolo:**  (dos placas paralelas)

**Unidad:** Faradio (F). En la práctica se usan submúltiplos:

- 1 μF (microfaradio) = 0.000001 F
- 1 nF (nanofaradio) = 0.000000001 F
- 1 pF (picofaradio) = 0.000000000001 F

### Tipos de condensadores:

Tipo	Rango típico	Característica	Uso
Cerámico	1 pF - 100 nF	Pequeño, económico, estable	Desacoplo, filtros
Electrolítico	1 μF - 10000 μF	Gran capacidad, polarizado	Filtros fuente alimentación
Poliéster	1 nF - 10 μF	Estable, buena precisión	Temporizadores
Tantalio	0.1 μF - 100 μF	Estable, caro	Circuitos críticos

**Importante:** Los condensadores electrolíticos y de tantalio tienen polaridad (+ y -). Conectarlos al revés puede hacerlos explotar.

### **Carga y descarga de un condensador:**

Cuando se conecta a una fuente de voltaje a través de una resistencia, el condensador no se carga instantáneamente. La tensión en sus bornes sigue una curva exponencial:

**Constante de tiempo:**  $\tau = R \times C$

- $\tau$  es el tiempo en segundos
- R es la resistencia en ohmios
- C es la capacidad en faradios
- En  $1\tau$ , el condensador se carga al 63% del voltaje final
- En  $5\tau$ , se considera completamente cargado (>99%)

**Ejemplo:**  $R = 10,000 \Omega$ ,  $C = 0.0001 \text{ F}$  (100  $\mu\text{F}$ )

$\tau = 10,000 \times 0.0001 = 1$  segundo

- En 1 s: se carga al 63% (7.56 V si la fuente es 12 V)

- En 5 s: se carga al 99% (11.88 V)

### **1.5.3 Asociación de componentes**

**Resistencias en serie:**  $R_{\text{total}} = R_1 + R_2 + R_3 + \dots$

- Dos resistencias de 100  $\Omega$  en serie:  $R = 200 \Omega$

**Resistencias en paralelo:**  $1/R_{\text{total}} = 1/R_1 + 1/R_2 + 1/R_3 + \dots$

- Dos resistencias de 100  $\Omega$  en paralelo:  $R = 50 \Omega$

**Condensadores en serie:**  $1/C_{\text{total}} = 1/C_1 + 1/C_2 + \dots$

**Condensadores en paralelo:**  $C_{\text{total}} = C_1 + C_2 + \dots$

### **1.5.4 Aplicación real: filtro de fuente de alimentación**

En una fuente de alimentación, después de rectificar la corriente alterna (con diodos), obtenemos una tensión pulsante. Un condensador electrolítico de gran capacidad (1000-4700  $\mu\text{F}$ ) se encarga de “alisar” esa tensión, reduciendo el rizado a un nivel aceptable.

### 1.5.5 Resumen

Componente	Función	Unidad	Polaridad
Resistencia	Limita corriente	Ohmio ( $\Omega$ )	No
Condensador	Almacena energía	Faradio (F)	Sí (electrolíticos)
Circuito RC	Temporización	$\tau = R \times C$	-

### 1.5.6 Ejercicios

**Ejercicio 1:** Lee el valor de una resistencia con bandas: Marrón-Negro-Rojo-Dorado.

(Solución:  $1-0-\times 100 = 1000 \Omega = 1 \text{ k}\Omega, \pm 5\%$ )

**Ejercicio 2:** Un circuito RC tiene  $R = 10,000 \Omega$  y  $C = 220 \mu\text{F}$ . ¿Cuánto tarda en cargarse completamente?

(Solución:  $\tau = 10000 \times 0.00022 = 2.2 \text{ s}$ ;  $5\tau = 11 \text{ s}$ )

**Ejercicio 3:** Calcula la resistencia total de: dos de  $100 \Omega$  en serie, y dos de  $100 \Omega$  en paralelo.

(Solución: Serie =  $200 \Omega$ ; Paralelo =  $50 \Omega$ )

\newpage

## CAPÍTULO 1.6: COMPONENTES ELECTRÓNICOS BÁSICOS II

### 1.6.1 Diodos: la válvula unidireccional

Un diodo es un componente semiconductor que permite el paso de la corriente en un solo sentido. Es como una válvula de retención en un circuito de agua.

**Símbolo:**  (la flecha indica la dirección de la corriente)

**Polaridad:**

- **Ánodo (+):** por donde entra la corriente

- **Cátodo (-):** por donde sale la corriente (marcado con una banda en el componente físico)

### **Cómo funciona:**

- Polarización directa (ánodo más positivo que cátodo): el diodo conduce
- Polarización inversa (cátodo más positivo que ánodo): el diodo no conduce

**Caída de tensión:** Cuando un diodo conduce, tiene una caída de tensión típica de 0.7 V (diodo de silicio). Esto significa que si aplicas 5 V al ánodo, obtienes 4.3 V en el cátodo.

### **Aplicación principal: Rectificación**

La rectificación convierte corriente alterna (AC) en corriente continua (DC). En una fuente de alimentación:

1. El transformador reduce el voltaje de 230 V AC a 12 V AC
2. El puente de diodos (4 diodos en configuración de puente) convierte la AC en DC pulsante
3. El condensador alisa la tensión pulsante
4. El regulador de voltaje estabiliza la salida

### **1.6.2 LEDs (Diodos Emisores de Luz)**

Un LED es un diodo que emite luz visible cuando circula corriente a través de él.

#### **Características típicas:**

- Voltaje directo: 1.8-3.3 V (depende del color)
- Corriente típica: 20 mA (0.02 A)
- El rojo tiene el voltaje más bajo (~1.8 V), el azul y blanco el más alto (~3.3 V)

#### **Cálculo de resistencia limitadora:**

$$R = (V_{\text{fuente}} - V_{\text{LED}}) \div I_{\text{LED}}$$

#### **Ejemplos:**

- LED rojo (2V) a 5V:  $R = (5-2) \div 0.02 = 150 \Omega$
- LED azul (3V) a 12V:  $R = (12-3) \div 0.02 = 450 \Omega$
- LED blanco (3.3V) a 24V:  $R = (24-3.3) \div 0.02 = 1035 \Omega \rightarrow$  valor comercial 1 k $\Omega$

#### **Conexión:**

El LED tiene polaridad: el terminal más largo es el ánodo (+), el más corto es el cátodo (-). Además, el cátodo suele tener un lado plano en el encapsulado.

### 1.6.3 Transistores

Un transistor es un semiconductor que puede funcionar como interruptor o como amplificador. Una pequeña corriente (o voltaje) puede controlar una corriente mucho mayor.

#### Tipos principales:

##### BJT (Bipolar Junction Transistor):

- Controlado por corriente
- Tres terminales: Base (B), Colector (C), Emisor (E)
- Una pequeña corriente de base controla una corriente grande de colector a emisor
- Ganancia ( $\beta$  o hFE): típicamente 100-800

##### MOSFET (Metal-Oxide-Semiconductor Field-Effect Transistor):

- Controlado por voltaje
- Tres terminales: Gate (G), Drain (D), Source (S)
- Un voltaje en la puerta controla la corriente entre drenador y fuente
- Consumo de control casi nulo
- Más eficiente que BJT para potencias medias-altas

#### Aplicación práctica: Activar un motor con Arduino

El Arduino solo puede proporcionar 40 mA por pin, insuficiente para un motor. Con un transistor:

```
Arduino (pin 9) —[R 1kΩ]— Base (BJT)
                          Colector — Motor — +12V
                          Emisor — GND
```

Cuando el pin del Arduino se pone a HIGH (5V), la base del transistor recibe corriente, el transistor conduce, y el motor de 12 V se activa.

### 1.6.4 Aplicación real

Los transistores MOSFET se usan en variadores de frecuencia para controlar motores industriales. Mediante conmutación a alta frecuencia (PWM), el variador puede controlar la velocidad del motor con gran eficiencia.

## 1.6.5 Resumen

Componente	Función	Polaridad	Terminales
Diodo	Conduce en 1 dirección	Sí	Ánodo, Cátodo
LED	Diodo que emite luz	Sí	Ánodo (+), Cátodo (-)
BJT	Interruptor por corriente	Sí (tipo NPN/PNP)	Base, Colector, Emisor
MOSFET	Interruptor por voltaje	Sí	Gate, Drain, Source

## 1.6.6 Ejercicios

**Ejercicio 1:** Quieres conectar un LED verde (2.2 V, 20 mA) a una fuente de 9 V. ¿Qué resistencia necesitas?

(Solución:  $R = (9 - 2.2) \div 0.02 = 6.8 \div 0.02 = 340 \Omega \approx 330 \Omega$  valor comercial)

**Ejercicio 2:** Un diodo de silicio conduce 2 A en polarización directa. ¿Qué potencia disipa? (Caída de tensión  $\approx 0.7$  V)

(Solución:  $P = 0.7 \times 2 = 1.4$  W)

**Ejercicio 3:** ¿Por qué no podemos conectar un motor de 24 V directamente a un pin de Arduino?

(Solución: Porque el pin de Arduino solo da 5 V y 40 mA, insuficiente para el motor. Necesitamos un transistor o un relé como intermediario.)

\newpage

# CAPÍTULO 1.7: SENSORES — FUNDAMENTOS

## 1.7.1 ¿Qué es un sensor?

Un sensor es un dispositivo que detecta una magnitud física (temperatura, presión, luz, distancia, etc.) y la convierte en una señal eléctrica que puede ser interpretada por un sistema de control.

En términos simples: los sensores son los “ojos”, “oídos” y “piel” de un sistema automatizado. Sin ellos, el controlador estaría “ciego” y no sabría qué está pasando en el proceso.

### 1.7.2 Tipos de señales

#### Señal digital (discreta):

- Solo dos estados posibles: ON (1) u OFF (0)
- También llamada “todo o nada”
- Fácil de interpretar, inmune al ruido
- Ejemplo: interruptor, final de carrera, pulsador

#### Señal analógica (continua):

- Puede tomar cualquier valor dentro de un rango
- Representa una magnitud que varía continuamente
- Puede ser en voltaje (0-10 V) o corriente (4-20 mA)
- Ejemplo: termopar, potenciómetro, sensor de presión

Característica	Digital	Analógica
Valores	0 o 1	Cualquiera en un rango
Cableado	2-3 hilos	2-4 hilos
Inmunidad ruido	Alta	Media
Resolución	1 bit	10-16 bits
Complejidad	Baja	Media

### 1.7.3 Sensores comunes en mecatrónica

Sensor	Magnitud que mide	Tipo de señal	Principio	Ejemplo de uso
Final de carrera	Presencia/ posición	Digital	Contacto mecánico	Detectar si una pieza llegó a una posición
Inductivo	Metales	Digital	Campo magnético	Detectar objetos metálicos en cinta
Capacitivo	Cualquier material	Digital	Campo eléctrico	Detectar plástico, madera, líquido
Fotoeléctrico	Presencia (luz)	Digital	Haz de luz	Detectar objetos en cinta transportadora
LDR	Luz	Analógica		

Sensor	Magnitud que mide	Tipo de señal	Principio	Ejemplo de uso
			Resistencia variable	Sensor de luminosidad
Termopar	Temperatura	Analógica	Efecto Seebeck	Medir temperatura de horno
PT100/PT1000	Temperatura	Analógica	Resistencia variable	Control de procesos
Presión (piezorresistivo)	Presión	Analógica	Deformación	Monitorear presión neumática
Ultrasónico	Distancia	Analógica/ Digital	Ondas sonoras	Medir nivel de tanque
Encoder	Posición/ velocidad	Digital (pulsos)	Óptico/magnético	Posición exacta de un eje
Acelerómetro	Aceleración	Analógica/ Digital	MEMS	Vibración, inclinación
Célula de carga	Fuerza/peso	Analógica	Galgas extensiométricas	Báscula industrial

### 1.7.4 Características importantes de los sensores

**Rango de medida:** Valores mínimo y máximo que puede medir (ej: -40 a 85 °C).

**Precisión:** Diferencia máxima entre el valor real y el medido (ej:  $\pm 0.5$  °C).

**Resolución:** El cambio más pequeño que puede detectar (ej: 0.1 °C).

**Tiempo de respuesta:** Lo que tarda en reaccionar a un cambio (ej: 10 ms).

**Histéresis:** Diferencia entre el valor de activación y desactivación. Por ejemplo, un sensor de temperatura puede activarse a 30 °C y desactivarse a 28 °C (histéresis de 2 °C).

**Salida eléctrica:** Cómo entrega la señal (PNP, NPN, 4-20 mA, 0-10 V).

### 1.7.5 Conexión típica de sensores industriales

La mayoría de sensores industriales tienen 3 o 4 cables:

Color	Función
Marrón	Alimentación positiva (+24 V DC)
Azul	Alimentación negativa (0 V / GND)
Negro (o blanco)	Señal de salida (va a la entrada del PLC)

### Configuración PNP (la más común):

- Cuando el sensor se activa, el cable negro se conecta a +24 V
- La entrada del PLC recibe +24 V = señal HIGH

### Configuración NPN:

- Cuando el sensor se activa, el cable negro se conecta a 0 V
- La entrada del PLC recibe 0 V = señal LOW

## 1.7.6 Aplicación real

En una fábrica de refrescos, una cinta transportadora lleva botellas. Un sensor fotoeléctrico detecta cada botella que pasa. Cuando el PLC ha contado 12 botellas mediante el sensor, activa un cilindro neumático que empuja el grupo de botellas hacia la máquina de empaquetado.

## 1.7.7 Resumen

- **Sensor:** convierte magnitud física en señal eléctrica
- **Señal digital:** ON/OFF (final de carrera, fotoeléctrico)
- **Señal analógica:** valores variables (temperatura, presión)
- **PNP/NPN:** tipo de salida del sensor (activo en alto/bajo)
- **Cableado típico:** marrón (+24V), azul (GND), negro (señal)

## 1.7.8 Ejercicios

**Ejercicio 1:** ¿Qué tipo de sensor usarías para: a) Detectar si una puerta blindada está abierta o cerrada, b) Medir la temperatura de un horno a 800 °C, c) Contar latas de aluminio en una cinta transportadora?

(Soluciones: a) Final de carrera o sensor magnético, b) Termopar tipo K, c) Sensor inductivo)

**Ejercicio 2:** Un sensor de temperatura PT100 varía de  $100 \Omega$  a  $0^\circ\text{C}$  a  $138.5 \Omega$  a  $100^\circ\text{C}$ . ¿Es una señal digital o analógica? ¿Por qué?

(Solución: Analógica, porque puede tomar valores intermedios continuos, no solo 0 o 1)

\newpage

## CAPÍTULO 1.8: ACTUADORES — FUNDAMENTOS

### 1.8.1 ¿Qué es un actuador?

Un actuador es un dispositivo que convierte una señal eléctrica (o de otro tipo) en acción física: movimiento, fuerza, presión. Si los sensores son los “ojos” del sistema automatizado, los actuadores son los “músculos”.

#### **Tipos de energía que pueden usar:**

- Eléctrica (motores, solenoides, relés)
- Neumática (cilindros, actuadores rotativos, ventosas)
- Hidráulica (cilindros de gran fuerza, motores hidráulicos)

### 1.8.2 Clasificación de actuadores

#### **Por tipo de movimiento:**

Tipo	Movimiento	Ejemplos
Lineal	Movimiento rectilíneo	Cilindros neumáticos, actuadores lineales eléctricos
Rotatorio	Giro continuo	Motores DC, AC, paso a paso, hidráulicos
Angular	Giro limitado ( $\leq 360^\circ$ )	Servomotores, actuadores rotativos
Sujeción	Agarre	Pinzas neumáticas, electroimanes
Conmutación	ON/OFF	Relés, contactores, válvulas solenoide

#### **Por tipo de energía:**

Tipo	Energía	Ventaja	Desventaja	Ejemplo típico
Motor DC	Eléctrica	Control simple, económico	Escobillas se desgastan	Ventilador, taladro
Motor AC	Eléctrica	Robusto, sin escobillas	Control más complejo	Bombas, compresores
Paso a paso	Eléctrica	Posición precisa sin encoder	Pierde pasos a alta velocidad	Impresora 3D, CNC
Servomotor	Eléctrica	Precisión con feedback	Caro, control complejo	Brazo robótico
Cilindro neumático	Aire comprimido	Rápido, simple, económico	Poco preciso	Empujar piezas
Cilindro hidráulico	Aceite presión	Fuerza enorme, preciso	Caro, pesado	Prensa, excavadora

### 1.8.3 Características para seleccionar un actuador

Característica	Definición	Unidad
Fuerza / Par	Capacidad de empuje o giro	N / Nm
Velocidad	Rapidez de movimiento	m/s, RPM
Carrera / Recorrido	Distancia de desplazamiento	mm, m
Precisión	Error de posicionamiento	mm, grados
Ciclo de trabajo	% de tiempo que puede estar activo	%
Consumo	Energía que necesita	W, l/min

### 1.8.4 ¿Cómo elegir el actuador adecuado?

Preguntas clave que debe hacerse un diseñador:

1. **¿Qué tipo de movimiento necesito?** (lineal, rotatorio, angular)
2. **¿Cuánta fuerza o par necesito?** (calcular cargas)
3. **¿A qué velocidad debe moverse?** (tiempo de ciclo)
4. **¿Qué precisión de posicionamiento se requiere?** (milímetros, centésimas)
5. **¿Qué fuente de energía tengo disponible?** (eléctrica, aire comprimido, aceite)

6. **¿Cuál es el entorno de trabajo?** (polvo, temperatura, humedad)

### 1.8.5 Aplicación real: selección para un robot pick-and-place

Un robot que coge piezas de una cinta y las coloca en cajas necesita:

- **Articulaciones:** servomotores con reductores (precisión, control)
- **Pinza:** cilindro neumático (rápido, simple, fuerza suficiente)
- **Movimiento vertical:** actuador lineal eléctrico o neumático
- **Control:** PLC o controlador de robot

Cada actuador se selecciona según la función específica. La pinza no necesita precisión milimétrica (neumática es suficiente), pero las articulaciones sí (servomotores).

### 1.8.6 Resumen

- **Actuador:** convierte señal en movimiento
- **Tipos:** motores (rotatorio), cilindros (lineal), solenoides (conmutación)
- **Selección:** según fuerza, velocidad, precisión y energía disponible
- **Energías:** eléctrica, neumática, hidráulica

### 1.8.7 Ejercicios

**Ejercicio 1:** ¿Qué actuador usarías para: a) Subir una persiana enrollable de 15 kg, b) Prensar chatarra con 50 toneladas de fuerza, c) Mover el cabezal de una impresora 3D con precisión de 0.1 mm?

(Soluciones: a) Motor DC con reductor, b) Cilindro hidráulico, c) Motor paso a paso con husillo)

**Ejercicio 2:** ¿Qué ventajas tiene un servomotor frente a un motor paso a paso?

(Solución: El servomotor tiene encoder (feedback de posición), mayor par a altas velocidades, mejor eficiencia y no pierde pasos. Es más caro pero más preciso y fiable.)

\newpage

# CAPÍTULO 1.9: CIRCUITOS PRÁCTICOS CON SENSORES Y ACTUADORES

## 1.9.1 Circuito 1: LED con interruptor

El circuito más básico demuestra los principios fundamentales.

### Componentes:

- Pila de 9 V
- Resistencia de 330  $\Omega$  (calculada para LED de 2 V, 20 mA)
- LED
- Interruptor
- Cables

### Esquema:

Pila (+) — Interruptor — Resistencia — LED — Pila (-)

**Funcionamiento:** Al cerrar el interruptor, circula corriente y el LED se enciende.

**Aplicación real:** Panel de señales en una máquina:

- Luz verde = máquina funcionando
- Luz roja = máquina parada
- Luz amarilla = máquina en pausa

## 1.9.2 Circuito 2: Sensor de luz (LDR) con LED

Un LDR (Light Dependent Resistor) varía su resistencia según la luz incidente: alta resistencia en oscuridad, baja resistencia con luz.

### Componentes:

- LDR
- Resistencia fija de 10 k $\Omega$
- Transistor BC547 (NPN)
- LED con resistencia de 330  $\Omega$
- Fuente de 9 V

### Funcionamiento:

Cuando oscurece, el LDR aumenta su resistencia. La base del transistor recibe voltaje suficiente para activarse. El transistor conduce y enciende el LED.

**Aplicación real:** Sensor crepuscular para alumbrado exterior automático.

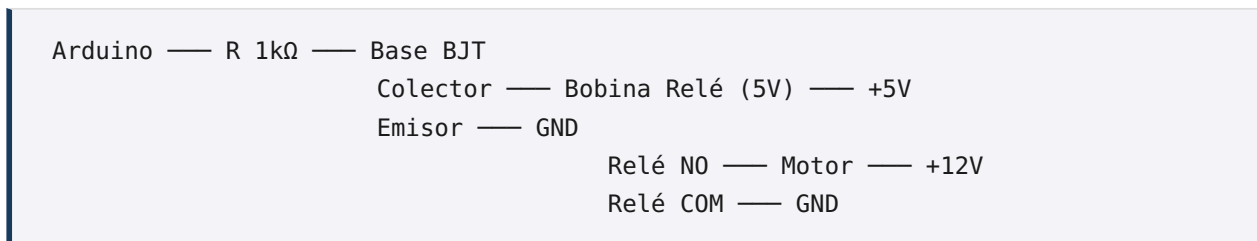
### 1.9.3 Circuito 3: Relé para control de motor

Un relé es un interruptor electromagnético. Con una señal pequeña (5 V, 20 mA) podemos controlar un circuito de potencia (12 V, 5 A).

#### Componentes:

- Relé de 5 V
- Transistor NPN (para activar el relé desde Arduino/sensor)
- Diodo de protección (en paralelo con la bobina del relé)
- Motor de 12 V
- Fuente de 12 V para el motor

#### Esquema:



**Funcionamiento:** Arduino activa el transistor → transistor activa la bobina del relé → relé cierra el contacto → motor gira.

### 1.9.4 Circuito 4: Final de carrera para limite de movimiento

Un final de carrera detecta cuando una pieza alcanza un límite físico.

#### Componentes:

- Final de carrera (normalmente cerrado, NC)
- Relé
- Motor DC
- Fuentes de alimentación

#### Funcionamiento:

Mientras el final de carrera no esté presionado, el motor puede funcionar. Cuando la pieza llega al final y presiona el final de carrera, el circuito se abre y el motor se para.

**Aplicación real:** Puerta automática de garaje. Los finales de carrera detectan cuando la puerta está completamente abierta o cerrada y paran el motor.

### 1.9.5 Sistema integrado simplificado

#### Sistema de llenado de botellas (versión básica):

1. Sensor fotoeléctrico detecta botella en posición
2. Señal llega al sistema de control (relés o PLC)
3. Se activa la válvula solenoide (abre paso de líquido)
4. Transcurre el tiempo de llenado (3 segundos)
5. Se desactiva la válvula solenoide (cierra paso)
6. Se activa el motor de la cinta transportadora
7. La botella avanza y la siguiente ocupa su lugar

Este sistema combina: sensor (fotoeléctrico), actuador (válvula solenoide), temporización y control secuencial.

### 1.9.6 Resumen

- Los circuitos básicos combinan sensores + control + actuadores
- El relé permite que señales pequeñas controlen cargas grandes
- Los finales de carrera detectan límites de movimiento
- Un sistema automatizado integra múltiples sensores y actuadores
- El control puede ser por relés cableados, Arduino o PLC

### 1.9.7 Ejercicio

Diseña (describe con palabras) un sistema que:

1. Detecte con un sensor de temperatura cuando un horno supera 200 °C
2. Active un ventilador para enfriar
3. Cuando la temperatura baja de 180 °C, pare el ventilador

(Solución: Sensor de temperatura (termopar + acondicionador) → Comparador (con histéresis) → Transistor → Relé → Motor ventilador. El comparador tiene dos umbrales: activa a 200 °C, desactiva a 180 °C).

\newpage

---

## NIVEL 2: PROGRAMACIÓN Y CONTROL

---

*“Las máquinas necesitan instrucciones. Aquí aprenderás a dárselas.”*

Pasamos de los componentes físicos a la lógica que los controla. La programación es el “cerebro” de cualquier sistema mecatrónico. Aprenderás desde cero: algoritmos, estructuras de control, Arduino y automatización básica.

---

### CAPÍTULO 2.1: INTRODUCCIÓN A LA LÓGICA DE PROGRAMACIÓN

#### 2.1.1 ¿Qué significa programar?

Programar es dar instrucciones a una máquina para que realice una tarea. Es como escribir una receta de cocina, pero para un ordenador.

#### **Características de las instrucciones para máquinas:**

- Deben ser **precisas** (sin ambigüedad)
- Deben ser **ordenadas** (secuencia correcta)
- Deben ser **completas** (cubrir todos los casos)

#### **Ejemplo: “Si la temperatura supera 30 °C, enciende el ventilador”**

Un humano entiende esta frase incluso si falta contexto. Una máquina necesita instrucciones mucho más detalladas:

1. Lee la temperatura del sensor
2. Compara el valor leído con 30
3. Si el valor es mayor que 30, envía señal HIGH al pin del ventilador
4. Si el valor es menor o igual a 30, envía señal LOW al pin del ventilador
5. Espera 1 segundo
6. Vuelve al paso 1

## 2.1.2 Algoritmos

Un algoritmo es una secuencia finita y ordenada de pasos que resuelve un problema.

### Ejemplo cotidiano: Algoritmo para hacer café con cafetera italiana

```
INICIO
1. Llenar la base de la cafetera con agua hasta la válvula
2. Llenar el filtro con café molido
3. Enroscar la parte superior
4. Colocar la cafetera en el fuego
5. Esperar hasta que el café suba
6. Retirar del fuego
7. Servir en una taza
FIN
```

#### Características de un buen algoritmo:

- **Finito:** debe terminar después de un número de pasos
- **Definido:** cada paso debe estar claramente definido
- **Efectivo:** debe poder ejecutarse con los recursos disponibles
- **Entrada/Salida:** debe producir un resultado

## 2.1.3 Representación de algoritmos

**Pseudocódigo:** Lenguaje intermedio entre el humano y el ordenador. No se ejecuta, se usa para diseñar la lógica.

**Diagramas de flujo:** Representación gráfica con símbolos (rectángulos = acciones, rombos = decisiones, óvalos = inicio/fin).

**Lenguaje de programación:** Código ejecutable por el ordenador.

## 2.1.4 Estructura de un programa en Arduino

Todo programa en Arduino (C++) tiene al menos dos bloques:

```
void setup() {
  // Se ejecuta UNA VEZ al iniciar
  // Aquí se configuran los pines, se inicializan variables
}

void loop() {
```

```
// Se ejecuta REPETIDAMENTE en un bucle infinito
// Aquí va la lógica principal del programa
}
```

### 2.1.5 Aplicación real

Un semáforo inteligente sigue este algoritmo:

```
INICIO
  REPETIR PARA SIEMPRE:
    Luz Verde ON durante 30 segundos
    Luz Verde OFF
    Luz Amarilla ON durante 5 segundos
    Luz Amarilla OFF
    Luz Roja ON durante 30 segundos
    Luz Roja OFF
  FIN REPETIR
FIN
```

### 2.1.6 Resumen

- **Programar:** dar instrucciones precisas y ordenadas a una máquina
- **Algoritmo:** secuencia de pasos para resolver un problema
- **Pseudocódigo:** lógica expresada en lenguaje humano
- **Estructura básica:** setup() → una vez, loop() → siempre

### 2.1.7 Ejercicio

Escribe en pseudocódigo el algoritmo para que un brazo robótico coja una pieza de la posición A y la coloque en la posición B. Incluye: ir a A, bajar, cerrar pinza, subir, ir a B, bajar, abrir pinza, subir.

\newpage

## CAPÍTULO 2.2: ESTRUCTURAS CONDICIONALES (IF/ELSE)

### 2.2.1 ¿Qué es una condición?

Una condición es una expresión que puede evaluarse como verdadera (true) o falsa (false). Permite que el programa tome decisiones.

### 2.2.2 La sentencia IF

```
if (condición) {  
    // código que se ejecuta si la condición es verdadera  
}
```

#### Ejemplo:

```
if (temperatura > 30) {  
    digitalWrite(ventiladorPin, HIGH);  
}
```

### 2.2.3 La sentencia IF-ELSE

```
if (condición) {  
    // código si la condición es verdadera  
} else {  
    // código si la condición es falsa  
}
```

#### Ejemplo:

```
if (temperatura > 30) {  
    digitalWrite(ventiladorPin, HIGH);  
} else {  
    digitalWrite(ventiladorPin, LOW);  
}
```

### 2.2.4 Anidamiento de condiciones

Podemos poner condiciones dentro de otras condiciones:

```

if (temperatura > 30) {
  if (humedad > 80) {
    // Hace calor Y hay humedad
    digitalWrite(ventiladorPin, HIGH);
  } else {
    // Hace calor PERO no hay humedad
    digitalWrite(ventiladorPin, LOW);
  }
} else {
  digitalWrite(ventiladorPin, LOW);
}

```

## 2.2.5 Operadores lógicos

Permiten combinar varias condiciones en una sola expresión:

```

if (temperatura > 30 && humedad > 80) {
  // Ambas condiciones deben cumplirse (Y lógico)
}
if (temperatura > 30 || paradaEmergencia == true) {
  // Al menos una debe cumplirse (O lógico)
}
if (!paradaEmergencia) {
  // La condición se niega (NO lógico)
}

```

## 2.2.6 Aplicación real: control de seguridad

En una máquina industrial, la lógica de seguridad típica:

```

if (botonArranque == true && puertaSeguridad == true && !paradaEmergencia) {
  arrancarMaquina();
} else {
  pararMaquina();
}

```

“Solo arranca la máquina si: se presiona el botón de arranque Y la puerta de seguridad está cerrada Y NO hay emergencia.”

## 2.2.7 Resumen

- **IF:** ejecuta código si la condición es verdadera

- **IF-ELSE:** ejecuta un código u otro según la condición
- **&& (Y):** verdadero solo si ambas condiciones son verdaderas
- **|| (O):** verdadero si al menos una condición es verdadera
- **! (NO):** invierte el valor de la condición

### 2.2.8 Ejercicio

Escribe en código Arduino:

“Si el sensor de nivel (pin A0) lee un valor mayor de 800 Y la válvula (pin 8) está cerrada (LOW), entonces abre la válvula (HIGH). Mantenla abierta 3 segundos y ciérrala.”

(Solución aproximada: `if (analogRead(A0) > 800 && digitalRead(8) == LOW) { digitalWrite(8, HIGH); delay(3000); digitalWrite(8, LOW); }`)

`\newpage`

## CAPÍTULO 2.3: BUCLES (WHILE, FOR)

### 2.3.1 ¿Por qué necesitamos bucles?

Los bucles permiten repetir un bloque de código múltiples veces. Son esenciales porque:

1. Las máquinas realizan tareas repetitivas
2. No queremos escribir el mismo código 100 veces
3. Algunos procesos deben ejecutarse hasta que ocurra una condición

### 2.3.2 Bucle WHILE (mientras)

Repite mientras la condición sea verdadera.

```
while (condición) {
  // código que se repite
}
```

#### Ejemplo: Esperar a que un sensor se active

```
while (digitalRead(sensorPin) == LOW) {
  // Espera aquí mientras el sensor no detecte nada
  delay(10);
}
// Cuando el sensor se activa, continúa
digitalWrite(motorPin, HIGH);
```

### **Ejemplo: Calentar agua hasta ebullición**

```
while (temperatura < 100) {
  mantenerCalefactorEncendido();
  temperatura = leerTemperatura();
  delay(1000);
}
apagarCalefactor();
Serial.println(";Agua hirviendo!");
```

### **2.3.3 Bucle FOR (para)**

Repite un número específico de veces.

```
for (inicialización; condición; incremento) {
  // código que se repite
}
```

### **Ejemplo: Encender LEDs en secuencia**

```
for (int i = 0; i < 8; i++) {
  digitalWrite(ledPins[i], HIGH);
  delay(500);
  digitalWrite(ledPins[i], LOW);
}
// LED 0 enciende 0.5s, luego LED 1, etc., hasta LED 7
```

### **Ejemplo: Llenar 12 botellas**

```
for (int botella = 0; botella < 12; botella++) {
  abrirValvula();
  delay(3000); // llenar 3 segundos
  cerrarValvula();
  avanceCinta();
}
```

```
delay(1000); // mover siguiente botella
}
```

### 2.3.4 Diferencias clave

Bucle	Se usa cuando...	Ejemplo
WHILE	No sabes cuántas veces repetir	“Esperar hasta que el sensor se active”
FOR	Sabes exactamente cuántas veces	“Repetir 10 veces”
loop()	Quieres repetir para siempre	“Bucle principal de Arduino”

### 2.3.5 WHILE infinito (equivalente a loop())

```
while (true) {
  // Esto se repite para siempre
  // Equivalente a la función loop() de Arduino
}
```

### 2.3.6 Aplicación real: Ciclo de una lavadora

```
void cicloLavado() {
  // Fase 1: Llenar
  abrirValvulaAgua();
  while (nivelAgua < NIVEL_MAX) {
    delay(100);
    nivelAgua = leerSensorNivel();
  }
  cerrarValvulaAgua();

  // Fase 2: Lavar (10 minutos)
  for (int minuto = 0; minuto < 10; minuto++) {
    girarTambor(DERECHA, 5);
    girarTambor(IZQUIERDA, 5);
  }

  // Fase 3: Vaciar
  abrirBombaDesague();
  while (nivelAgua > 0) {
    delay(500);
    nivelAgua = leerSensorNivel();
  }
}
```

```
cerrarBombaDesague();  
}
```

### 2.3.7 Resumen

- **WHILE:** repite mientras la condición sea verdadera
- **FOR:** repite un número fijo de veces
- Los bucles son esenciales para tareas repetitivas
- Los procesos industriales son secuencias de bucles

### 2.3.8 Ejercicio

Escribe un bucle FOR que encienda 6 LEDs (del pin 2 al pin 7) uno tras otro con 500 ms de diferencia, y luego los apague todos a la vez.

\newpage

## CAPÍTULO 2.4: VARIABLES, FUNCIONES

### 2.4.1 Variables

Una variable es un espacio en la memoria del ordenador donde guardamos un valor que puede cambiar.

#### Declaración de variables en C++/Arduino:

```
int temperatura = 25;           // número entero (-32768 a 32767)  
unsigned long tiempo = 3000;   // entero positivo grande  
float voltaje = 12.5;          // número decimal  
boolean motorOn = true;       // verdadero/falso  
char letra = 'A';              // un solo carácter  
String mensaje = "Hola";      // texto
```

#### Reglas para nombrar variables:

- Deben empezar con letra o guión bajo (\_)
- No pueden contener espacios
- Distinguen mayúsculas/minúsculas (temperatura ≠ Temperatura)
- Deben ser descriptivas (int temperatura ≠ int t)

## 2.4.2 Ámbito de variables

**Variable global:** declarada fuera de cualquier función, accesible desde todo el programa.

**Variable local:** declarada dentro de una función, solo existe dentro de ella.

```
int contadorGlobal = 0; // global: accesible desde cualquier parte

void setup() {
  int contadorLocal = 0; // local: solo existe en setup()
}

void loop() {
  contadorGlobal++; // OK: global es accesible
  // contadorLocal++; // ERROR: no existe fuera de setup()
}
```

## 2.4.3 Funciones

Una función es un bloque de código con nombre que realiza una tarea específica. Se puede “llamar” desde cualquier parte del programa.

**Estructura:**

```
tipo_de_retorno nombreFuncion(tipo parametro1, tipo parametro2) {
  // cuerpo de la función
  return valor; // (opcional, solo si el tipo no es void)
}
```

**Ejemplos:**

```
// Función sin retorno (void)
void encenderLED(int pin) {
  digitalWrite(pin, HIGH);
}

// Función con retorno
int sumar(int a, int b) {
  return a + b;
}

// Función que lee un sensor y devuelve la temperatura
float leerTemperatura(int pinSensor) {
  int valor = analogRead(pinSensor);
}
```

```
float voltaje = valor * (5.0 / 1023.0);
return voltaje * 100.0; // LM35: 10mV/°C
}
```

#### 2.4.4 Ventajas de las funciones

1. **Reutilización:** escribes una vez, usas muchas veces
2. **Organización:** código más legible y estructurado
3. **Mantenimiento:** cambiar la función una vez actualiza todas las llamadas
4. **Abstracción:** no necesitas saber cómo funciona internamente, solo qué hace

#### Ejemplo: control de un brazo robótico

```
// Funciones de bajo nivel
void moverBase(int grados) { /* envía comando al servo base */ }
void moverHombro(int grados) { /* envía comando al servo hombro */ }
void moverCodo(int grados) { /* envía comando al servo codo */ }
void abrirPinza() { /* activa servo pinza */ }
void cerrarPinza() { /* desactiva servo pinza */ }

// Función de alto nivel (compuesta)
void recogerPieza() {
    moverHombro(45);
    moverBase(-30);
    moverCodo(90);
    cerrarPinza();
    delay(500);
    moverCodo(0);
    moverHombro(0);
}

// Programa principal
void loop() {
    if (digitalRead(sensorPieza) == HIGH) {
        recogerPieza();
        delay(2000);
    }
}
```

### 2.4.5 Aplicación real

Los programas de PLC industriales se estructuran en bloques de función (FB). Cada bloque realiza una operación estándar: ControlMotor, ControlValvula, ControlTanque. Esto permite reutilizar código probado en múltiples máquinas.

### 2.4.6 Resumen

- **Variables:** espacios de memoria con nombre, tipo y valor
- **Tipos:** int, float, boolean, String, char
- **Funciones:** bloques de código reutilizables
- **Void:** función sin retorno; otros tipos devuelven un valor

### 2.4.7 Ejercicio

Escribe una función llamada `controlTemperatura` que reciba la temperatura actual y:

- Si  $temp > 40$ : active LED rojo (pin 13) y apague verde (pin 12)
- Si  $temp < 30$ : active LED verde y apague rojo
- Si  $30 \leq temp \leq 40$ : apague ambos LEDs

\newpage

## CAPÍTULO 2.5: ARDUINO — INTRODUCCIÓN

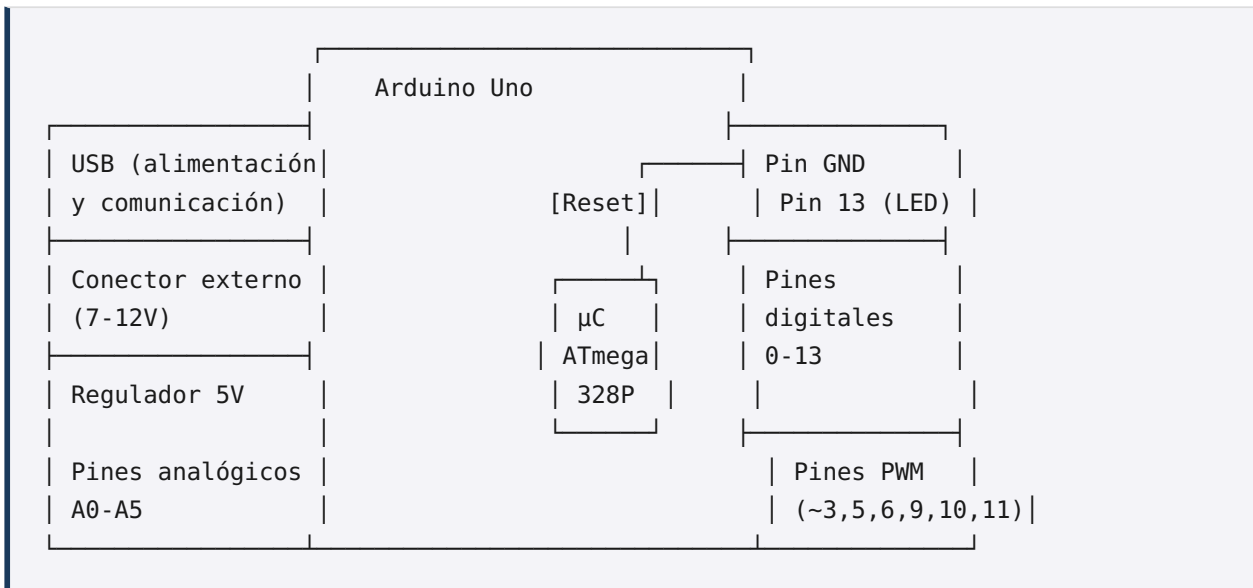
### 2.5.1 ¿Qué es Arduino?

Arduino es una plataforma de hardware libre basada en un microcontrolador. Un microcontrolador es un “miniordenador” en un solo chip que incluye procesador, memoria y entradas/salidas.

#### Características que hacen único a Arduino:

- **Económico:** una placa cuesta 20-30 €
- **Open source:** los diseños son públicos, cualquiera puede fabricarlos
- **Fácil de programar:** el entorno de desarrollo (IDE) es simple e intuitivo
- **Comunidad enorme:** miles de proyectos, librerías y tutoriales disponibles
- **Robusto:** soporta conexiones incorrectas, voltajes fuera de rango

## 2.5.2 Componentes del Arduino UNO



### Especificaciones técnicas:

Característica	Valor
Microcontrolador	ATmega328P
Voltaje de trabajo	5 V
Voltaje de entrada (recomendado)	7-12 V
Pines digitales	14 (6 con PWM)
Pines analógicos	6
Corriente máxima por pin	40 mA
Corriente máxima total	200 mA
Memoria Flash	32 KB (0.5 KB para bootloader)
SRAM	2 KB
EEPROM	1 KB
Velocidad de reloj	16 MHz

## 2.5.3 El primer programa: Blink

El “Hola Mundo” de Arduino es un LED que parpadea.

```

void setup() {
  pinMode(13, OUTPUT); // Configura el pin 13 como salida
}

void loop() {
  digitalWrite(13, HIGH); // Enciende el LED (5V en pin 13)
  delay(1000);           // Espera 1000 ms (1 segundo)
  digitalWrite(13, LOW); // Apaga el LED (0V en pin 13)
  delay(1000);           // Espera 1 segundo
}

```

### Explicación:

- `pinMode(pin, modo)` : configura un pin como entrada (INPUT) o salida (OUTPUT)
- `digitalWrite(pin, valor)` : escribe HIGH (5V) o LOW (0V) en un pin de salida
- `delay(tiempo)` : pausa el programa durante X milisegundos

### 2.5.4 El proceso de trabajo

1. Escribes el código en el Arduino IDE (entorno de desarrollo)
2. Conectas la placa Arduino al ordenador mediante USB
3. Seleccionas el tipo de placa y el puerto USB
4. Haces clic en “Subir” (cargar)
5. El IDE compila el código y lo transfiere al Arduino
6. El Arduino se reinicia y ejecuta el programa
7. A partir de aquí, el Arduino funciona independientemente del ordenador

### 2.5.5 Aplicación real

Arduino se usa en:

- **Prototipado industrial:** probar conceptos antes de implementarlos en PLC
- **Pequeñas automatizaciones:** control de invernaderos, sistemas de riego
- **Educación:** aprender programación y electrónica
- **Productos comerciales:** impresoras 3D, drones, robots educativos

En el contexto de FP Superior, Arduino es el puente entre la teoría y los PLC industriales. Los conceptos aprendidos aquí se aplican directamente a sistemas más grandes y robustos.

## 2.5.6 Resumen

- Arduino = microcontrolador fácil de usar y programar
- UNO: 14 pines digitales, 6 analógicos, 5V, 16 MHz
- pinMode(): configura el pin
- digitalWrite(): escribe HIGH/LOW en un pin
- delay(): pausa el programa
- loop(): se ejecuta para siempre

## 2.5.7 Ejercicio

¿Qué hace este programa?

```
void setup() {  
  pinMode(13, OUTPUT);  
  pinMode(12, OUTPUT);  
}  
void loop() {  
  digitalWrite(13, HIGH);  
  digitalWrite(12, LOW);  
  delay(2000);  
  digitalWrite(13, LOW);  
  digitalWrite(12, HIGH);  
  delay(500);  
}
```

(Solución: LED en pin 13 se enciende 2 segundos, luego se apaga y LED en pin 12 se enciende 0.5 segundos. Esto se repite infinitamente.)

\newpage

# CAPÍTULO 2.6: ARDUINO — ENTRADAS Y SALIDAS

## 2.6.1 Entradas digitales

Un pin digital puede leer si hay voltaje (HIGH, 5V) o no (LOW, 0V).

```
int valor = digitalRead(pin); // Devuelve HIGH o LOW
```

## Ejemplo completo: Botón que controla un LED

```
int botonPin = 2; // Pin del pulsador
int ledPin = 13;  // Pin del LED

void setup() {
  pinMode(botonPin, INPUT); // Pin del botón como entrada
  pinMode(ledPin, OUTPUT);  // Pin del LED como salida
}

void loop() {
  int estadoBoton = digitalRead(botonPin);

  if (estadoBoton == HIGH) { // Botón presionado
    digitalWrite(ledPin, HIGH);
  } else {                  // Botón suelto
    digitalWrite(ledPin, LOW);
  }
}
```

### Problema: rebotes (bouncing)

Cuando presionas un pulsador, el contacto no es perfecto: hace y deshace contacto varias veces en milisegundos. Esto puede hacer que el Arduino lea múltiples pulsaciones en lugar de una.

### Solución: antirrebote por software

```
if (digitalRead(botonPin) == HIGH) {
  delay(50); // Esperar a que se estabilice
  if (digitalRead(botonPin) == HIGH) {
    // Ahora sí, es una pulsación válida
  }
}
```

## 2.6.2 Entradas analógicas

Los pines analógicos (A0-A5) pueden leer voltajes variables de 0 a 5 V y convertirlos a un valor digital de 0 a 1023 (10 bits de resolución).

```
int valor = analogRead(pin); // Devuelve 0-1023
```

### Relación voltaje-valor digital:

- 0 V → 0
- 2.5 V → 512
- 5 V → 1023

**Cálculo:** Voltaje = (valor × 5.0) ÷ 1023

### Ejemplo: Potenciómetro controla brillo de LED

```
int potPin = A0;
int ledPin = 9; // Pin con capacidad PWM (~)

void setup() {
  pinMode(ledPin, OUTPUT);
}

void loop() {
  int valorPot = analogRead(potPin); // 0-1023
  int brillo = map(valorPot, 0, 1023, 0, 255); // Convertir a 0-255
  analogWrite(ledPin, brillo); // PWM para controlar brillo
}
```

### La función map():

`map(valor, desdeMin, desdeMax, hastaMin, hastaMax)` — reescala un valor de un rango a otro.

### 2.6.3 Salidas PWM (Pulse Width Modulation)

PWM es una técnica que permite simular voltajes variables entre 0 y 5 V conmutando rápidamente entre HIGH y LOW.

- **Valor 0:** siempre LOW (0V)
- **Valor 128:** 50% del tiempo HIGH, 50% LOW (≈2.5V promedio)
- **Valor 255:** siempre HIGH (5V)

### Usos del PWM:

- Control de brillo de LEDs
- Control de velocidad de motores DC
- Generación de tonos con zumbadores
- Señales de control para servomotores

**Pines PWM en Arduino UNO:** 3, 5, 6, 9, 10, 11 (marcados con ~)

## 2.6.4 Sensor + actuador completo

### Proyecto: Ventilador controlado por temperatura

```
int sensorTemp = A0; // LM35 (sensor de temperatura)
int motorPin = 9;    // Motor DC vía transistor (PWM)

void setup() {
  pinMode(motorPin, OUTPUT);
  Serial.begin(9600); // Para depuración
}

void loop() {
  int lectura = analogRead(sensorTemp);
  float voltaje = lectura * (5.0 / 1023.0);
  float temperatura = voltaje * 100.0; // LM35: 10mV/°C

  Serial.print("Temperatura: ");
  Serial.println(temperatura);

  if (temperatura < 25) {
    analogWrite(motorPin, 0); // Apagado
  } else if (temperatura < 30) {
    analogWrite(motorPin, 100); // Velocidad baja
  } else if (temperatura < 35) {
    analogWrite(motorPin, 180); // Velocidad media
  } else {
    analogWrite(motorPin, 255); // Máxima velocidad
  }

  delay(5000); // Leer cada 5 segundos
}
```

## 2.6.5 Resumen

Función	Lee/Escribe	Rango	Ejemplo
digitalRead(pin)	Lee digital	HIGH/LOW	Leer pulsador
digitalWrite(pin, v)	Escribe digital	HIGH/LOW	Encender LED
analogRead(pin)	Lee analógica	0-1023	Leer potenciómetro
analogWrite(pin, v)	Escribe PWM	0-255	Controlar motor
map(valor, a, b, c, d)	Reescala	Variable	0-1023 → 0-255

## 2.6.6 Ejercicio

Escribe un programa que lea un potenciómetro (A0) y:

- Si el valor  $< 341$ : encienda LED verde (pin 11)
- Si  $341 \leq \text{valor} < 682$ : encienda LED amarillo (pin 12)
- Si valor  $\geq 682$ : encienda LED rojo (pin 13)

\newpage

# CAPÍTULO 2.7: PROYECTOS ARDUINO PARA MECATRÓNICA

## 2.7.1 Proyecto 1: Semáforo automático

Un clásico que introduce temporización y secuencias.

**Componentes:** 3 LEDs (rojo, amarillo, verde), 3 resistencias de  $220 \Omega$ , Arduino

```
int rojo = 12, amarillo = 11, verde = 10;

void setup() {
  pinMode(rojo, OUTPUT);
  pinMode(amarillo, OUTPUT);
  pinMode(verde, OUTPUT);
}

void loop() {
  // Verde 5 segundos
  digitalWrite(verde, HIGH);
  delay(5000);
  digitalWrite(verde, LOW);

  // Amarillo 2 segundos
  digitalWrite(amarillo, HIGH);
  delay(2000);
  digitalWrite(amarillo, LOW);

  // Rojo 5 segundos
  digitalWrite(rojo, HIGH);
  delay(5000);
  digitalWrite(rojo, LOW);
}
```

## 2.7.2 Proyecto 2: Control de motor con sensor de distancia

Detecta objetos cercanos con un sensor ultrasónico HC-SR04 y activa un motor.

**Componentes:** HC-SR04, motor DC + transistor, Arduino

```
int trigPin = 9, echoPin = 8;
int motorPin = 10;

void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(motorPin, OUTPUT);
}

long medirDistancia() {
  // Enviar pulso ultrasónico
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Medir tiempo de retorno del eco
  long duracion = pulseIn(echoPin, HIGH);

  // Convertir a distancia en cm
  // Velocidad del sonido: 343 m/s
  // Distancia = tiempo × velocidad / 2 (ida y vuelta)
  return duracion / 58; // Aproximación: duración/58 = cm
}

void loop() {
  long distancia = medirDistancia();

  if (distancia < 20) {
    digitalWrite(motorPin, HIGH);
  } else {
    digitalWrite(motorPin, LOW);
  }

  delay(100);
}
```

### 2.7.3 Proyecto 3: Control de servomotor (brazo robótico simple)

Controla la posición de un servomotor con un potenciómetro.

**Componentes:** Servomotor SG90, potenciómetro 10kΩ, Arduino

```
#include <Servo.h>

Servo miServo;
int potPin = A0;

void setup() {
  miServo.attach(9); // Servo conectado al pin 9
}

void loop() {
  int valorPot = analogRead(potPin);
  int angulo = map(valorPot, 0, 1023, 0, 180);
  miServo.write(angulo);
  delay(15); // Tiempo para que el servo se mueva
}
```

### 2.7.4 Proyecto 4: Sistema de alarma con sensor PIR

Detecta movimiento y activa una alarma visual y sonora.

**Componentes:** Sensor PIR (movimiento), LED rojo, zumbador, Arduino

```
int pirPin = 2;
int ledPin = 13;
int buzzerPin = 12;

void setup() {
  pinMode(pirPin, INPUT);
  pinMode(ledPin, OUTPUT);
  pinMode(buzzerPin, OUTPUT);
  Serial.begin(9600);
  Serial.println("Calibrando sensor...");
  delay(10000); // Tiempo de calibración del PIR
  Serial.println("Listo!");
}

void loop() {
  if (digitalRead(pirPin) == HIGH) {
    Serial.println("¡Movimiento detectado!");
  }
}
```

```

// Alarma visual y sonora
for (int i = 0; i < 10; i++) {
  digitalWrite(ledPin, HIGH);
  tone(buzzerPin, 1000);
  delay(200);
  digitalWrite(ledPin, LOW);
  noTone(buzzerPin);
  delay(200);
}
}
}

```

### 2.7.5 Aplicación real

Estos proyectos son versiones educativas de sistemas industriales reales:

- **Semáforo** → similar a paneles de señalización en máquinas
- **Sensor distancia + motor** → cinta transportadora con detector de presencia
- **Servomotor + potenciómetro** → control de válvula o compuerta
- **Alarma PIR** → sistemas de seguridad industrial

### 2.7.6 Ejercicio

Modifica el proyecto del semáforo para que funcione con un pulsador: cuando se presiona, el semáforo cambia a rojo para peatones durante 5 segundos, emite un zumbido y vuelve al ciclo normal.

\newpage

## CAPÍTULO 2.8: AUTOMATIZACIÓN BÁSICA — ON/OFF

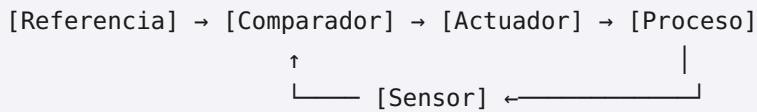
### 2.8.1 Control ON/OFF

Es el tipo de control más simple y el más extendido en la industria: el actuador está completamente encendido o completamente apagado. No hay valores intermedios.

#### Ejemplos cotidianos:

- **Termostato:** La calefacción se enciende o se apaga
- **Cisterna:** La válvula de llenado se abre o se cierra
- **Plancha:** La resistencia se enciende o se apaga para mantener temperatura

## Diagrama de bloques:



### 2.8.2 El problema: oscilación

El control ON/OFF puro tiene un problema: si el punto de conmutación es único, el sistema oscila constantemente.

**Ejemplo:** Un horno debe mantener 200 °C.

- Si el control ON/OFF activa a 200 °C y desactiva a 200 °C:
- El horno se enciende → temperatura sube → 200 °C → se apaga
- Pero el horno está caliente → sigue subiendo → apagado por inercia
- La temperatura baja → 200 °C → se enciende
- Pero estaba frío → sigue bajando → sobreoscilación

El resultado es una oscilación continua alrededor del punto de consigna.

### 2.8.3 Solución: histéresis

La histéresis crea una “banda muerta” alrededor del punto de consigna. El sistema se activa en un punto y se desactiva en otro diferente.

**Ejemplo con histéresis:**

- **Activar calefacción:** cuando temperatura baja de 19 °C
- **Desactivar calefacción:** cuando temperatura sube de 21 °C
- Entre 19 y 21 °C: la calefacción mantiene el estado anterior

La banda de 2 °C (19-21) evita las oscilaciones. Cuanto mayor es la histéresis, menos ciclos de conmutación pero peor precisión.

### 2.8.4 Implementación en Arduino

```
const int tempMin = 19;      // Temperatura de activación
const int tempMax = 21;     // Temperatura de desactivación
bool calefaccionEncendida = false;

void loop() {
  float temp = leerTemperatura(); // Función que lee el sensor
```

```

// Lógica ON/OFF con histéresis
if (temp < tempMin && !calefaccionEncendida) {
    digitalWrite(calefactorPin, HIGH);
    calefaccionEncendida = true;
    Serial.println("Calefacción ENCENDIDA");
}

if (temp > tempMax && calefaccionEncendida) {
    digitalWrite(calefactorPin, LOW);
    calefaccionEncendida = false;
    Serial.println("Calefacción APAGADA");
}

delay(5000); // Leer temperatura cada 5 segundos
}

```

### 2.8.5 Aplicación real

El control ON/OFF con histéresis se usa en:

- **Termostatos industriales:** control de temperatura en hornos, cámaras frigoríficas
- **Control de nivel:** bombas que llenan tanques (activan al 20%, desactivan al 80%)
- **Control de presión:** compresores que mantienen la presión en un circuito neumático
- **Neveras y congeladores:** el compresor se enciende y apaga según la temperatura

### 2.8.6 Resumen

- **Control ON/OFF:** encendido o apagado, sin valores intermedios
- **Problema:** oscilación si el punto de conmutación es único
- **Solución:** histéresis (banda entre activación y desactivación)
- **Aplicaciones:** temperatura, nivel, presión — procesos simples

### 2.8.7 Ejercicio

Diseña un control ON/OFF con histéresis para el nivel de un tanque de agua:

- La bomba se enciende cuando el nivel baja del 25%
- La bomba se apaga cuando el nivel sube al 85%
- Implementa la lógica en pseudocódigo

\newpage

## CAPÍTULO 2.9: TEMPORIZADORES Y CONTROL SECUENCIAL

### 2.9.1 Temporizadores

Un temporizador es un dispositivo (o función) que mide intervalos de tiempo y ejecuta acciones cuando estos transcurren.

#### Tipos de temporizadores:

Tipo	Comportamiento
TON (On Delay)	Se activa la entrada → espera tiempo → activa la salida
TOF (Off Delay)	Se activa la entrada → activa salida inmediato; desactiva entrada → espera → desactiva salida
TP (Pulse)	Se activa entrada → genera pulso de duración fija

#### Implementación de temporizador TON en Arduino:

```
unsigned long tiempoInicio = 0;
bool temporizadorActivo = false;
const unsigned long TIEMPO_ESPERA = 5000; // 5 segundos

void iniciarTemporizador() {
    tiempoInicio = millis(); // millis() = ms desde que arrancó Arduino
    temporizadorActivo = true;
}

bool temporizadorCumplido() {
    if (temporizadorActivo && (millis() - tiempoInicio >= TIEMPO_ESPERA)) {
        temporizadorActivo = false;
        return true;
    }
    return false;
}

void loop() {
    if (digitalRead(botonPin) == HIGH && !temporizadorActivo) {
        iniciarTemporizador();
        digitalWrite(ledPin, HIGH);
    }
}
```

```

    }

    if (temporizadorCumplido()) {
        digitalWrite(ledPin, LOW);
    }
}

```

### 2.9.2 Control secuencial (máquina de estados)

Las máquinas industriales realizan secuencias de operaciones. Cada paso tiene condiciones de avance (tiempo cumplido, sensor activado, etc.).

#### Ejemplo: Secuencia de una prensa automática

1. REPOS0: Esperar pieza en posición
2. BAJAR: Cilindro baja la prensa (3 segundos)
3. Prensar: Activar presión durante 5 segundos
4. SUBIR: Cilindro sube la prensa (3 segundos)
5. EXPULSAR: Cilindro expulsa pieza (2 segundos)
6. Volver a 1

### 2.9.3 Implementación con máquina de estados

```

enum Estado {
    REPOS0,
    BAJAR,
    PRENSAR,
    SUBIR,
    EXPULSAR
};

Estado estadoActual = REPOS0;
unsigned long tiempoEstado = 0;

void cambiarEstado(Estado nuevoEstado) {
    estadoActual = nuevoEstado;
    tiempoEstado = millis();
}

void loop() {
    switch(estadoActual) {
        case REPOS0:
            if (digitalRead(sensorPieza) == HIGH) {
                cambiarEstado(BAJAR);
            }

```

```

    }
    break;

    case BAJAR:
        digitalWrite(cilindroBajar, HIGH);
        if (millis() - tiempoEstado >= 3000) {
            digitalWrite(cilindroBajar, LOW);
            cambiarEstado(PRENSAR);
        }
        break;

    case PRENSAR:
        digitalWrite(valvulaPresion, HIGH);
        if (millis() - tiempoEstado >= 5000) {
            digitalWrite(valvulaPresion, LOW);
            cambiarEstado(SUBIR);
        }
        break;

    case SUBIR:
        digitalWrite(cilindroSubir, HIGH);
        if (millis() - tiempoEstado >= 3000) {
            digitalWrite(cilindroSubir, LOW);
            cambiarEstado(EXPULSAR);
        }
        break;

    case EXPULSAR:
        digitalWrite(cilindroExpulsar, HIGH);
        if (millis() - tiempoEstado >= 2000) {
            digitalWrite(cilindroExpulsar, LOW);
            cambiarEstado(REPOS0);
        }
        break;
}
}

```

## 2.9.4 La función millis() vs delay()

delay()	millis()
Bloquea el programa	No bloquea
No puedes hacer otra cosa mientras esperas	Puedes leer sensores, comunicarte, etc.
Simple de usar	Más complejo

<code>delay()</code>	<code>millis()</code>
Adecuado para programas simples	Necesario para programas profesionales

**Regla práctica:** Usa `millis()` siempre que necesites hacer varias cosas a la vez (leer sensores mientras esperas).

### 2.9.5 Aplicación real

Las máquinas de inyección de plástico, las prensas hidráulicas y las líneas de ensamblaje funcionan con control secuencial basado en máquinas de estados. Cada etapa tiene su tiempo y condiciones de transición.

### 2.9.6 Resumen

- **Temporizadores:** miden tiempo para retardar acciones (TON, TOF, TP)
- **Control secuencial:** secuencia de pasos con condiciones de avance
- **Máquina de estados:** organización clara de programas secuenciales
- **`millis()` vs `delay()`:** `millis()` permite multitarea; `delay()` bloquea

### 2.9.7 Ejercicio

Diseña la máquina de estados para un semáforo con pulsador peatonal:

Estados: VERDE, AMARILLO, ROJO, ROJO\_PEATON

Condiciones de transición y tiempos:

- VERDE 30s → AMARILLO
- AMARILLO 5s → ROJO
- ROJO (normal): si pulsador fue presionado → ROJO\_PEATON (30s) → VERDE
- ROJO (sin pulsador): 30s → VERDE

\newpage

---

# NIVEL 3: FP SUPERIOR — MECATRÓNICA PROFESIONAL

---

*“Has llegado al nivel de Formación Profesional Superior. Aquí verás cómo funcionan las máquinas reales de una fábrica.”*

En este nivel aprenderás PLC (automatización industrial), neumática, hidráulica y sistemas mecatrónicos integrados. Son los conocimientos que se exigen en FP Superior de Mecatrónica y Automatización.

---

## CAPÍTULO 3.1: PLC — INTRODUCCIÓN Y ARQUITECTURA

### 3.1.1 ¿Qué es un PLC?

Un PLC (Programmable Logic Controller) es un ordenador industrial diseñado específicamente para controlar máquinas y procesos en tiempo real. A diferencia de un PC, está construido para funcionar en entornos industriales hostiles: polvo, vibración, temperaturas extremas, interferencias eléctricas.

#### Historia breve:

Antes de los PLC (años 60), el control industrial se hacía con paneles de relés. Para cambiar la lógica de funcionamiento, había que recablear todo el panel. Los PLC permitieron cambiar la lógica mediante software, sin modificar el cableado.

### 3.1.2 PLC vs PC vs Arduino

Característica	PC	Arduino	PLC
Entorno típico	Oficina	Laboratorio	Fábrica
Temperatura	10-35 °C	0-50 °C	-20 a +60 °C
Tiempo de respuesta	ms-s	µs-ms	µs (tiempo real)
Fiabilidad	Media	Media	

Característica	PC	Arduino	PLC
			Muy alta (MTBF > 100,000 h)
E/S integradas	Teclado, pantalla	14 digital, 6 analógica	100+ digital/analógica
Programación	C, Python, Java	C++	Ladder, FBD, SFC, ST
Coste	500-2000 €	20-30 €	200-5000 €

### 3.1.3 Componentes internos de un PLC

#### 1. Fuente de alimentación:

Convierte la tensión de red (230 V AC) en la tensión de trabajo del PLC (24 V DC). Alimenta la CPU, los módulos de E/S y los sensores/actuadores.

#### 2. CPU (Unidad Central de Proceso):

- Procesador (microprocesador o microcontrolador)
- Memoria de programa (almacena el código del usuario)
- Memoria de datos (almacena variables, temporizadores, contadores)
- Batería (mantiene los datos cuando se apaga)

#### 3. Módulos de entrada:

Conectan los sensores al PLC. Convierten señales del mundo real (24 V, 4-20 mA) en señales digitales que la CPU puede procesar.

- Entradas digitales (DI): 24 V DC, 110 V AC, 230 V AC
- Entradas analógicas (AI): 0-10 V, 4-20 mA, termopar, PT100

#### 4. Módulos de salida:

Convierten las órdenes de la CPU en señales que activan actuadores.

- Salidas digitales (DO): relé, transistor (24 V DC), triac (230 V AC)
- Salidas analógicas (AO): 0-10 V, 4-20 mA

#### 5. Módulos especiales:

- Comunicaciones (Profinet, Ethernet/IP, Modbus)
- Control de movimiento (servomotores, paso a paso)
- Contaje rápido (encoders de alta velocidad)
- Seguridad funcional (módulos de seguridad)

### 3.1.4 Marcas principales

Marca	Gama baja	Gama media	Gama alta	Software
Siemens	LOGO!	S7-1200	S7-1500	TIA Portal
Allen Bradley	MicroLogix	CompactLogix	ControlLogix	Studio 5000
Omron	CP1	CJ2	NX/NJ	Sysmac Studio
Mitsubishi	FX	FX5U	Q Series	GX Works
Schneider	Zelio	M340	M580	EcoStruxure

**Siemens S7-1200** es el más usado en FP Superior por su relación calidad-precio y la disponibilidad de documentación educativa.

### 3.1.5 Ciclo de scan (escanéo)

El PLC ejecuta el programa en un ciclo continuo:

1. LEER ENTRADAS  
| Copiar el estado de los sensores a la memoria  
▼
2. EJECUTAR PROGRAMA  
| Procesar la lógica del usuario (Ladder, FBD, etc.)  
▼
3. ESCRIBIR SALIDAS  
| Copiar los resultados a los actuadores  
▼
4. TAREAS INTERNAS  
| Comunicaciones, diagnosis, watchdog  
▼
5. (Volver a 1)

**Tiempo de scan:** normalmente 1-100 ms. Es el tiempo que tarda el PLC en completar un ciclo completo.

**Importante:** El programa se ejecuta de arriba a abajo y de izquierda a derecha, peldaño por peldaño. Esto es fundamental para entender el comportamiento de ciertas instrucciones.

### 3.1.6 Aplicación real

Una planta embotelladora utiliza un PLC Siemens S7-1200 para controlar:

- **24 entradas digitales:** sensores fotoeléctricos, finales de carrera, pulsadores
- **4 entradas analógicas:** sensor de nivel (4-20 mA), temperatura (PT100)
- **16 salidas digitales:** válvulas solenoide, contactores de motor, relés
- **2 salidas analógicas:** variador de velocidad para cinta transportadora
- **Ciclo de scan:** 8 ms
- **Programa:** 350 peldaños de ladder, 12 temporizadores, 8 contadores

### 3.1.7 Resumen

- **PLC:** ordenador industrial para control en tiempo real
- **Componentes:** fuente, CPU, módulos de E/S, módulos especiales
- **Ciclo de scan:** leer entradas → ejecutar programa → escribir salidas
- **Marcas principales:** Siemens (S7-1200/1500), Allen Bradley, Omron, Mitsubishi

### 3.1.8 Ejercicio

Explica con tus palabras tres razones por las que un PLC es mejor que un Arduino para controlar una máquina industrial de gran tamaño.

\newpage

## CAPÍTULO 3.2: LENGUAJE LADDER (Diagrama de Escalera)

### 3.2.1 Origen y filosofía

El lenguaje Ladder (o KOP en alemán) se llama así porque visualmente parece una escalera. Fue diseñado para que los electricistas pudieran programar PLC sin aprender informática, ya que se basa en los esquemas eléctricos de relés que ya conocían.

#### Estructura:

- **Rail izquierdo:** línea de alimentación (fase)
- **Rail derecho:** línea de retorno (neutro)
- **Peldaños (rungs):** cada línea horizontal es un peldaño

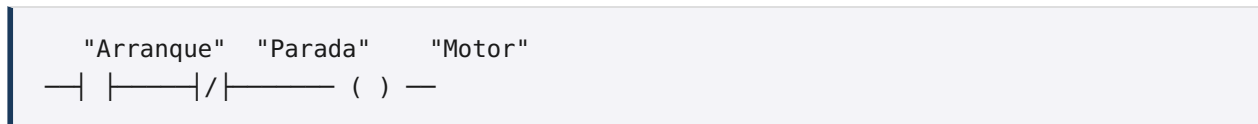
- **Contactos:** elementos que permiten pasar corriente (condiciones)
- **Bobinas:** elementos que se activan (acciones)

### 3.2.2 Elementos básicos

Símbolo	Nombre	Alemán	Significado
	Contacto NA (Normal Abierto)	Schließer	TRUE si la entrada está activada
	Contacto NC (Normal Cerrado)	Öffner	TRUE si la entrada NO está activada
	Bobina (Coil)	Spule	Activa la salida cuando TRUE
	Bobina SET	Setzen	Activa y mantiene
	Bobina RESET	Rücksetzen	Desactiva y mantiene

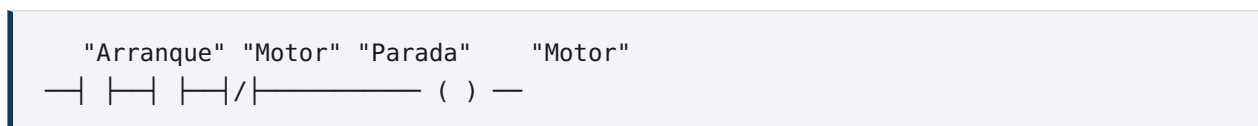
### 3.2.3 Primer programa: arranque y parada de motor

#### Peldaño 1 (sin retención):



Problema: el motor solo funciona mientras presionas Arranque. Al soltar, se para.

#### Peldaño 2 (con retención o enclavamiento):

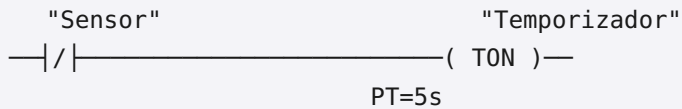


#### Cómo funciona:

1. Presionas Arranque (contacto NA se cierra)
2. Bobina Motor se activa
3. Contacto Motor (NA) se cierra (auto-retención)
4. Sueltas Arranque → la corriente pasa por el contacto Motor
5. Presionas Parada → se abre el contacto NC → Motor se desactiva

### 3.2.4 Temporizadores

**TON (Timer ON Delay):** Retardo a la conexión.



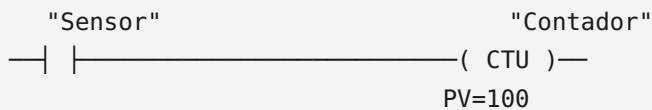
- Cuando "Sensor" se activa → Timer empieza a contar 5s
- Pasados 5s → la salida "Temporizador" se activa
- Si "Sensor" se desactiva antes de 5s → Timer se reinicia

**TOF (Timer OFF Delay):** Retardo a la desconexión.

- Cuando "Sensor" se activa → salida se activa inmediatamente
- Cuando "Sensor" se desactiva → Timer empieza a contar
- Pasado el tiempo → salida se desactiva

### 3.2.5 Contadores

**CTU (Counter Up):** Contador ascendente.

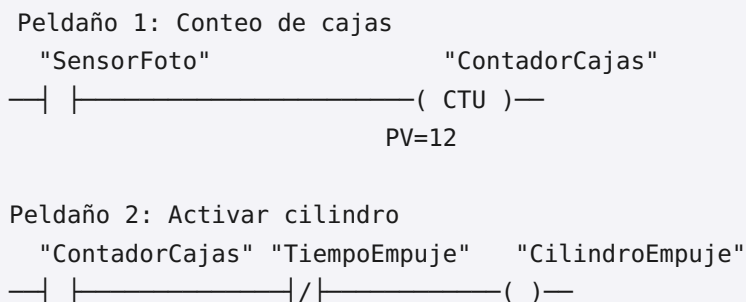


- Cada pulso de "Sensor" incrementa el contador
- Cuando alcanza 100 (PV), la salida se activa

**CTD (Counter Down):** Contador descendente.

### 3.2.6 Aplicación real: control de cinta con conteo

**Situación:** Una cinta transportadora lleva cajas. Cuando se acumulan 12 cajas, un cilindro las empuja a una rampa.



```

Peldaño 3: Temporizar empuje (2s)
  "CilindroEmpuje"          "TiempoEmpuje"
  —| |—————( TON )—
                                PT=2s

```

```

Peldaño 4: Reiniciar contador
  "TiempoEmpuje"          "Reset Contador"
  —| |—————( R )—

```

### 3.2.7 Operaciones combinadas (AND, OR)

**AND (Y):** contactos en serie.

```

  "A"    "B"    "C"
  —|/|—|/|—|/|—( )—
  // C se activa solo si A Y B Y C están activados

```

**OR (O):** contactos en paralelo.

```

      "A"
  —|/|—( )
      |
      "B"
  —|/|—

```

// la salida se activa si A O B están activados

### 3.2.8 Resumen

- Contactos NA (|/|): activos cuando hay señal
- Contactos NC (|/|): activos cuando NO hay señal
- Bobinas ( ): activan salidas
- TON: retardo a la conexión; TOF: retardo a la desconexión
- CTU: contador ascendente; CTD: contador descendente
- Serie = AND; Paralelo = OR

### 3.2.9 Ejercicio

Escribe en ladder: Dos pulsadores (A y B). La salida Motor se activa si se presiona A Y NO se presiona B. El motor debe mantenerse aunque se suelte A (auto-retención).

\newpage

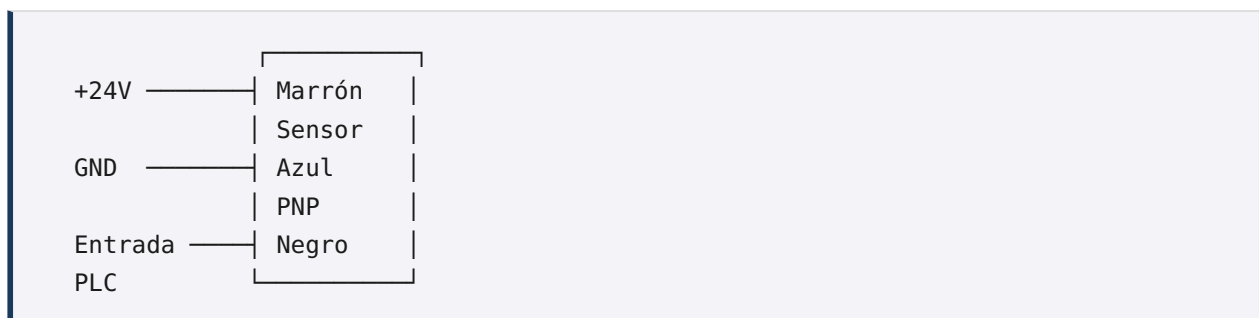
## CAPÍTULO 3.3: ENTRADAS Y SALIDAS INDUSTRIALES

### 3.3.1 Clasificación de señales

Señal	Tipo	Rango	Uso típico
DI (Digital Input)	ON/OFF	24 V DC	Pulsadores, finales de carrera
DO (Digital Output)	ON/OFF	24 V DC / relé	Válvulas, contactores, LEDs
AI (Analog Input)	Analógica	0-10 V, 4-20 mA	Temperatura, presión, nivel
AO (Analog Output)	Analógica	0-10 V, 4-20 mA	Variadores de velocidad

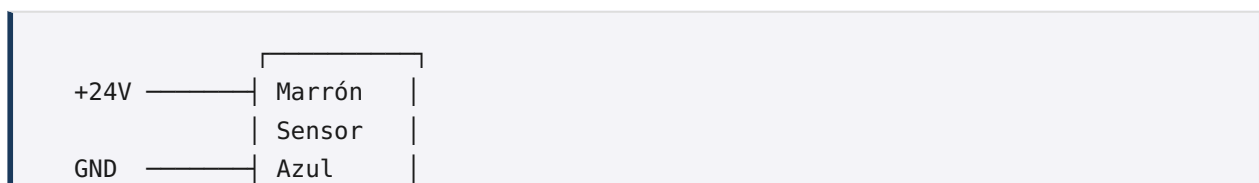
### 3.3.2 Cableado de sensores (entradas)

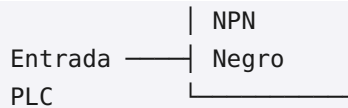
**Sensor con salida PNP (activo en alto):**



Cuando el sensor detecta, el cable Negro se conecta a +24V. La entrada del PLC recibe 24V = 1 lógico.

**Sensor con salida NPN (activo en bajo):**





Cuando el sensor detecta, el cable Negro se conecta a GND. La entrada del PLC recibe 0V = 1 lógico (con resistencia pull-up).

### 3.3.3 Tipos de sensores según conexión

#### 3 hilos (más común en sensores industriales):

- Marrón: +24V
- Azul: GND (0V)
- Negro: señal

#### 2 hilos (sensor se alimenta del propio bucle de señal):

- Marrón: +24V (alimentación y señal)
- Azul: GND

#### 4 hilos (sensores analógicos con alimentación separada):

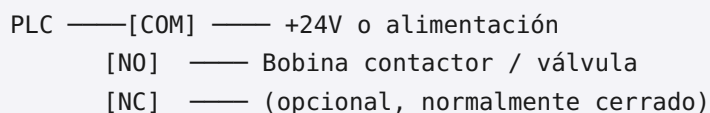
- Marrón: +24V
- Azul: GND
- Negro: señal analógica (0-10V)
- Blanco: señal de referencia

### 3.3.4 Cableado de actuadores (salidas)

#### Salida por relé (contacto seco):

Ventaja: puede conmutar diferentes tensiones (24V DC, 230V AC).

Desventaja: vida limitada (desgaste mecánico), más lento.



#### Salida por transistor (estado sólido):

Ventaja: rápida, larga vida, sin desgaste.

Desventaja: solo conmuta un tipo de tensión (24V DC).

### 3.3.5 Direccionamiento (Siemens S7-1200)

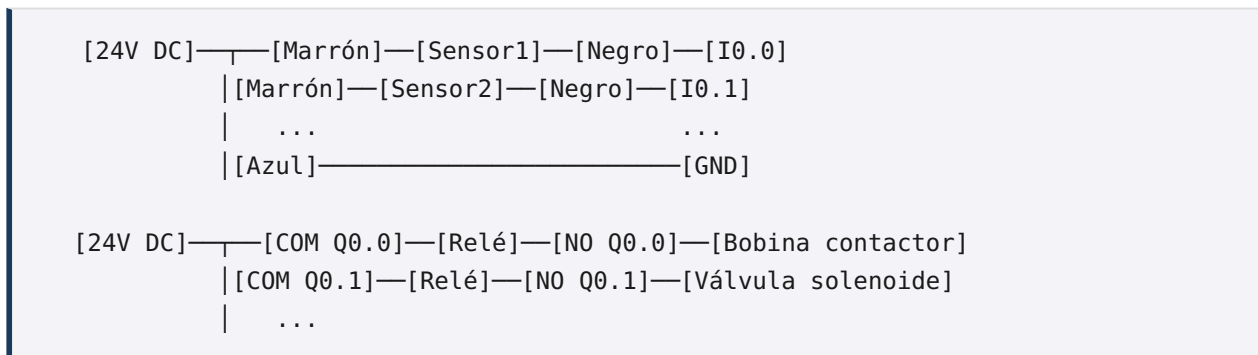
Cada entrada y salida tiene una dirección única:

Dirección	Significado
%I0.0	Entrada digital, módulo 0, borne 0
%I0.1	Entrada digital, módulo 0, borne 1
%I1.0	Entrada digital, módulo 1, borne 0
%Q0.0	Salida digital, módulo 0, borne 0
%IW64	Entrada analógica, palabra 64 (valor 0-27648)
%QW64	Salida analógica, palabra 64

En el programa Ladder, se usan sin el %: I0.0, Q0.0, IW64, etc.

### 3.3.6 Aplicación real: esquema de conexión

Un PLC S7-1200 con 8 entradas y 6 salidas típico:



### 3.3.7 Resumen

- **DI/DO:** señales digitales (ON/OFF)
- **AI/AO:** señales analógicas (0-10V, 4-20mA)
- **PNP:** activo en alto (+24V); NPN: activo en bajo (0V)
- **Direccionamiento:** I = entrada, Q = salida, seguido de módulo y borne
- **Cableado:** marrón (+24V), azul (GND), negro (señal)

### 3.3.8 Ejercicio

¿Qué tipo de entrada o salida usarías para cada caso?

- Pulsador de parada de emergencia
- Sensor de temperatura con salida 4-20 mA

- c) Cilindro neumático con electroválvula 24V DC
- d) Variador de frecuencia (para controlar velocidad de motor)
- e) Final de carrera para detectar posición

(Soluciones: a) DI, b) AI, c) DO, d) AO, e) DI)

\newpage

## CAPÍTULO 3.4: PROGRAMACIÓN AVANZADA DE PLC

### 3.4.1 Temporizadores avanzados

#### TON (Timer ON Delay):

Parámetros:

IN (BOOL): entrada que activa el temporizador  
PT (TIME): tiempo de preselección (ej: T#5s)  
Q (BOOL): salida (activa cuando tiempo  $\geq$  PT)  
ET (TIME): tiempo transcurrido (lectura)

Funcionamiento:

IN = TRUE  $\rightarrow$  ET empieza a incrementar  
ET  $\geq$  PT  $\rightarrow$  Q = TRUE  
IN = FALSE  $\rightarrow$  ET = 0, Q = FALSE

#### TOF (Timer OFF Delay):

IN = TRUE  $\rightarrow$  Q = TRUE (inmediato)  
IN = FALSE  $\rightarrow$  ET empieza a incrementar  
ET  $\geq$  PT  $\rightarrow$  Q = FALSE  
IN = TRUE  $\rightarrow$  ET = 0, Q = TRUE

#### TP (Timer Pulse):

IN = TRUE (flanco)  $\rightarrow$  Q = TRUE durante PT  
El pulso se completa aunque IN se desactive

### 3.4.2 Contadores avanzados

#### CTU (Counter Up):

Parámetros:

CU (BOOL): entrada de cuenta (flanco ascendente)  
R (BOOL): reset (pone el contador a 0)  
PV (INT): valor de preselección  
Q (BOOL): salida activa cuando  $CV \geq PV$   
CV (INT): valor actual del contador

### CTD (Counter Down):

CD (BOOL): entrada de cuenta descendente  
LD (BOOL): load (carga el valor inicial)  
PV (INT): valor de preselección inicial  
Q (BOOL): salida activa cuando  $CV \leq 0$

**CTUD (Counter Up/Down):** Combinación de los dos anteriores.

## 3.4.3 Comparadores

Permiten comparar valores y activar salidas según el resultado:

CMP == IW64, 1000 → Q0.0 = TRUE si IW64 = 1000  
CMP > IW64, 1000 → Q0.0 = TRUE si IW64 > 1000  
CMP < IW64, 1000 → Q0.0 = TRUE si IW64 < 1000

### Aplicación: control de temperatura

```
"TempActual" 1500 "TempAlta"  
—|CMP > |—( )—  
IW64
```

## 3.4.4 Operaciones matemáticas

ADD: resultado = valor1 + valor2  
SUB: resultado = valor1 - valor2  
MUL: resultado = valor1 × valor2  
DIV: resultado = valor1 ÷ valor2  
MOVE: copia un valor de una variable a otra

### 3.4.5 Bloques de función (FB)

Los FB son subrutinas reutilizables con memoria propia. Cada instancia tiene sus propios datos internos.

#### Ejemplo: FB "ControlMotor"

- Entradas: START, STOP, TIMER\_SET
- Salidas: RUN, FAULT
- Interno: temporizador, contador de arranques

Ventaja: puedes usar el mismo FB para 10 motores diferentes, cada uno con sus propios parámetros y estado.

### 3.4.6 Aplicación real: máquina de llenado completa

```
// Peldaño 1: Detectar botella
  "SensorFoto"          "ContadorBotellas"
—| |—————( CTU )— PV=1

// Peldaño 2: Posicionar botella bajo la boquilla
  "ContadorBotellas"  "MotorCinta"
—|/|—————( )—

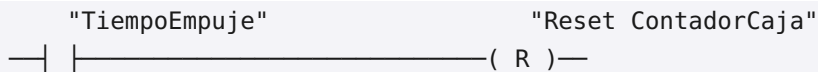
// Peldaño 3: Iniciar llenado
  "ContadorBotellas"  "ValvulaLlenado"  "TanqueLleno"  "ValvulaLlenado"
—| |—————|/|—————|/|—————( S )—

// Peldaño 4: Temporizar llenado
  "ValvulaLlenado"    "TiempoLlenado"
—| |—————( TON )— PT=3s

// Peldaño 5: Cerrar válvula y contar botella llenada
  "TiempoLlenado"    "Reset ContadorBotellas"
—| |—————( R )—
  "TiempoLlenado"    "ContadorCaja"
—| |—————( CTU )— PV=12

// Peldaño 6: Activar empaquetado cuando haya 12 botellas
  "ContadorCaja"      "CilindroEmpaque"
—| |—————( )—

// Peldaño 7: Temporizar empuje y reiniciar
  "CilindroEmpaque"  "TiempoEmpuje"
—| |—————( TON )— PT=2s
```



### 3.4.7 Resumen

- **TON/TOF/TP:** temporizadores (retardo conexión, desconexión, pulso)
- **CTU/CTD/CTUD:** contadores (ascendente, descendente, bidireccional)
- **Comparadores:** CMP (==, >, <, >=, <=)
- **Operaciones:** ADD, SUB, MUL, DIV, MOVE
- **FB:** Bloques de función reutilizables con memoria propia

### 3.4.8 Ejercicio

Diseña en ladder: Un tanque se llena cuando el nivel está por debajo del 20% (valor analógico < 5530 sobre 27648). Cuando alcanza el 80% (> 22118), se para la bomba y se activa un mezclador durante 10 segundos.

\newpage

## CAPÍTULO 3.5: NEUMÁTICA — FUNDAMENTOS

### 3.5.1 Introducción

La neumática utiliza aire comprimido para realizar trabajo mecánico. Es una de las tecnologías más importantes en automatización industrial por su simplicidad, fiabilidad y seguridad.

#### ¿Por qué aire comprimido?

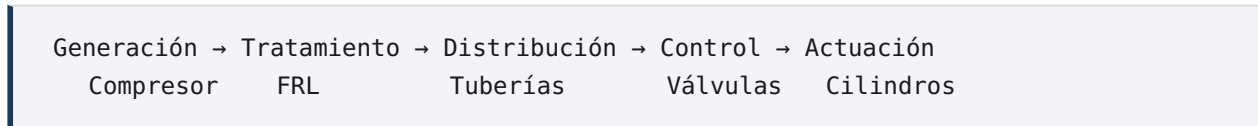
- El aire es gratuito e ilimitado
- No produce chispas (seguro en atmósferas explosivas)
- Los componentes son robustos y fiables
- El movimiento es rápido y fácil de controlar
- Fácil almacenamiento (tanques de presión)

#### Limitaciones:

- El aire es compresible (menos preciso que hidráulica)
- Ruidoso (escape de aire)

- Requiere mantenimiento (filtros, lubricación)
- Menor eficiencia energética que sistemas eléctricos

### 3.5.2 Componentes de un sistema neumático



**1. Compresor:** Genera aire comprimido (típicamente 6-10 bar).

- Compresor de pistón (pequeñas instalaciones)
- Compresor de tornillo (grandes instalaciones, más eficiente)

**2. Tanque de almacenamiento:** Acumula aire para mantener presión estable y compensar picos de consumo.

**3. Unidad de mantenimiento (FRL):**

- **Filtro:** elimina impurezas, polvo, agua condensada
- **Regulador:** mantiene la presión constante (ajustable)
- **Lubricador:** añade una fina niebla de aceite para lubricar componentes

**4. Tuberías y conexiones:** Distribuyen el aire a los puntos de consumo.

**5. Válvulas:** Controlan la dirección, presión y caudal del aire.

**6. Actuadores:** Convierten la energía neumática en movimiento (cilindros, motores neumáticos).

### 3.5.3 Unidades de medida en neumática

Magnitud	Unidad	Relación
Presión	Bar	1 bar = 100,000 Pa = 0.987 atm
Caudal	l/min (litros/minuto)	Cantidad de aire que fluye
Fuerza	N (newton)	$F = P \times A$
Volumen	litros (l)	Capacidad del tanque

**Presiones típicas en la industria:**

- Red general de fábrica: 6-8 bar
- Herramientas neumáticas: 6 bar

- Cilindros: 4-10 bar
- Sistemas de vacío: -0.7 a -0.9 bar

### 3.5.4 Cálculo de fuerza de un cilindro

#### Fórmula fundamental:

$$F = P \times A$$

Donde:

- F = Fuerza (N)
- P = Presión (Pa) — ¡cuidado! 1 bar = 100,000 Pa
- A = Área del pistón (m<sup>2</sup>) —  $A = \pi \times r^2$

#### Ejemplo:

Cilindro de 50 mm de diámetro (radio = 25 mm = 0.025 m), presión 6 bar.

Paso 1: Área del pistón

$$A = \pi \times 0.025^2 = 3.1416 \times 0.000625 = 0.00196 \text{ m}^2$$

Paso 2: Presión en Pascal

$$P = 6 \times 100,000 = 600,000 \text{ Pa}$$

Paso 3: Fuerza

$$F = 600,000 \times 0.00196 = 1178 \text{ N}$$

Paso 4: En kilogramos-fuerza (dividir entre 9.8)

$$1178 \div 9.8 \approx 120 \text{ kg}$$

#### Fuerza en retroceso (cilindro de doble efecto):

En el retroceso, el área efectiva es menor porque el vástago ocupa parte del área:

$$A_{\text{retroceso}} = \pi \times (r_{\text{pistón}}^2 - r_{\text{vástago}}^2)$$

### 3.5.5 Consumo de aire

#### Cálculo del consumo de un cilindro:

$$\text{Consumo} = \text{Área} \times \text{Carrera} \times \text{Presión}_{\text{absoluta}} \times \text{Ciclos/segundo}$$

**Ejemplo:** Cilindro de 50 mm diámetro, 200 mm carrera, 6 bar, 10 ciclos/minuto.

$$\begin{aligned} \text{Volumen por ciclo (avance + retroceso)} &= A \times \text{carrera} \times 2 \\ &= 0.00196 \times 0.2 \times 2 = 0.000784 \text{ m}^3 = 0.784 \text{ litros} \end{aligned}$$

Consumo por minuto =  $0.784 \times 10 = 7.84$  l/min

Consumo absoluto (incluyendo presión) =  $7.84 \times (6+1) = 54.9$  l/min

Este cálculo es importante para dimensionar el compresor.

### 3.5.6 Aplicación real

En una fábrica de automóviles, la red de aire comprimido:

- Compresores de tornillo de 75 kW
- Presión de red: 7 bar
- Consumo total: 5000 l/min
- Tubería principal: 4 pulgadas
- Puntos de consumo: cilindros, herramientas neumáticas, pintura

### 3.5.7 Resumen

- **Neumática:** aire comprimido para generar movimiento
- **Componentes:** compresor → tanque → FRL → válvulas → cilindros
- **Presión típica:** 6 bar
- **Fuerza:**  $F = P \times A$  (presión × área del pistón)
- **Consumo:** depende de área, carrera, presión y frecuencia

### 3.5.8 Ejercicio

Calcula la fuerza que puede ejercer un cilindro de 40 mm de diámetro (radio = 20 mm = 0.02 m) con una presión de 5 bar.

(Solución:  $A = \pi \times 0.02^2 = 0.001257$  m<sup>2</sup>;  $P = 500,000$  Pa;  $F = 500,000 \times 0.001257 = 628$  N  $\approx$  64 kg-fuerza)

\newpage

## CAPÍTULO 3.6: CILINDROS Y VÁLVULAS NEUMÁTICAS

### 3.6.1 Cilindros neumáticos

**Cilindro de simple efecto:**

- Tiene un solo puerto de entrada de aire

- El aire empuja el vástago hacia fuera (avance)
- Un muelle interno devuelve el vástago (retroceso)
- Se usa cuando solo se necesita fuerza en una dirección
- Menor consumo de aire

### Cilindro de doble efecto:

- Dos puertos de entrada de aire
- El aire por el puerto A empuja el vástago hacia fuera
- El aire por el puerto B empuja el vástago hacia dentro
- Fuerza disponible en ambas direcciones
- Más control sobre el movimiento

### Parámetros de selección:

Parámetro	Descripción	Rango típico
Diámetro	Diámetro del pistón	16, 20, 25, 32, 40, 50, 63, 80, 100 mm
Carrera	Distancia que recorre el vástago	25, 50, 80, 100, 160, 200, 320, 400, 500 mm
Presión máxima	Presión de trabajo máxima	10 bar (estándar), 16 bar
Fuerza	$F = P \times A$	Varía según diámetro y presión
Velocidad	Rapidez del movimiento	0.1 - 1.5 m/s
Amortiguación	Sistema de frenado al final de carrera	Neumática, elástica

### Tipos de montaje:

- Montaje fijo (patas, bridas)
- Montaje articulado (oscilante)
- Montaje en eje (guía)

### 3.6.2 Válvulas neumáticas

Las válvulas se identifican por el número de vías y posiciones:

**Notación:** X/Y donde X = número de vías (conexiones), Y = número de posiciones

#### Válvula 3/2 (3 vías, 2 posiciones):

- 3 puertos: Presión (P), Salida (A), Escape (R)

- 2 posiciones: reposo (P bloqueado, A → R) / activada (P → A, R bloqueado)
- Se usa para cilindros de simple efecto

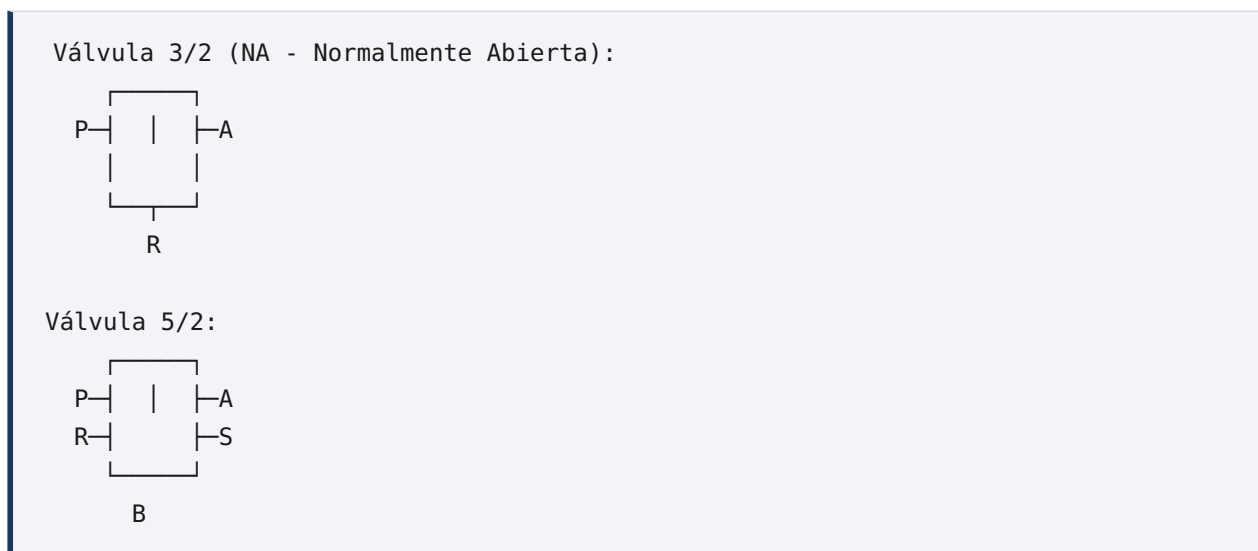
**Válvula 5/2 (5 vías, 2 posiciones):**

- 5 puertos: Presión (P), Salida A, Salida B, Escape R, Escape S
- 2 posiciones: reposo (P→A, B→R) / activada (P→B, A→S)
- Es la más común para cilindros de doble efecto

**Válvula 5/3 (5 vías, 3 posiciones):**

- Como la 5/2 pero con posición central
- Posición central: puede ser centrada cerrada, centrada abierta, centrada a escape
- Permite parar el cilindro en cualquier posición

**Representación de válvulas:**



**3.6.3 Accionamiento de válvulas**

Tipo de accionamiento	Símbolo	Descripción
Manual por pulsador	☒	Botón que permite el paso de aire
Manual por palanca	☐	Palanca con dos posiciones
Mecánico por rodillo	○	Actuado por una leva o pieza
Eléctrico por solenoide	☒/	Electroválvula (24V DC o 230V AC)
Neumático por pilotaje	→	Activado por aire a presión

## 6.4 Aplicación real: secuencia de empuje

### Sistema de transferencia de piezas:

1. Sensor fotoeléctrico detecta pieza en posición
2. Señal al PLC
3. PLC activa electroválvula 5/2 (solenoides Y1)
4. Válvula conmuta → aire a puerto A del cilindro
5. Cilindro avanza y empuja la pieza fuera de la cinta
6. Transcurre 1 segundo (temporizador del PLC)
7. PLC desactiva solenoides Y1
8. La válvula vuelve a reposo → aire a puerto B del cilindro
9. Cilindro retrocede

### 3.6.5 Resumen

Componente	Función
Cilindro simple efecto	Fuerza en 1 dirección + retroceso por muelle
Cilindro doble efecto	Fuerza en avance y retroceso
Válvula 3/2	Controla cilindro simple efecto
Válvula 5/2	Controla cilindro doble efecto (2 posiciones)
Válvula 5/3	Controla cilindro doble efecto (3 posiciones, permite parada)

### 3.6.6 Ejercicio

Selecciona la válvula adecuada para:

- a) Un cilindro de simple efecto que debe avanzar al presionar un pulsador y retroceder al soltarlo
- b) Un cilindro de doble efecto controlado por un PLC (avance y retroceso mediante dos solenoides)
- c) Un cilindro de doble efecto que debe poder pararse en cualquier posición intermedia

(Soluciones: a) Válvula 3/2 con retorno por muelle; b) Válvula 5/2 biestable (dos solenoides); c) Válvula 5/3 centrada cerrada)

\newpage

## CAPÍTULO 3.7: ELECTRONEUMÁTICA

### 3.7.1 ¿Qué es la electroneumática?

Es la combinación de componentes eléctricos/electrónicos (sensores, PLC) con componentes neumáticos (válvulas, cilindros). El PLC procesa las señales de los sensores y activa las electroválvulas que controlan los cilindros.

#### **Ventajas de la combinación:**

- Control flexible (cambiar lógica sin modificar neumática)
- Secuencias complejas posibles
- Integración con otros sistemas
- Diagnóstico y monitorización

### 3.7.2 Componentes electroneumáticos

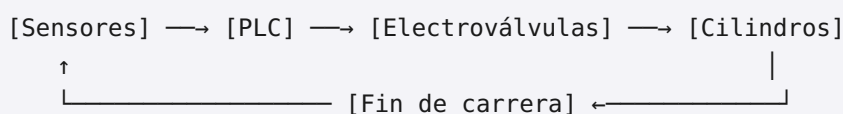
#### **Electroválvula:**

- Válvula neumática con accionamiento por solenoide (bobina eléctrica)
- Tensión típica: 24 V DC (también 110 V AC, 230 V AC)
- Potencia: 2-5 W
- Tiempo de respuesta: 10-50 ms
- Con o sin LED indicador

#### **Finales de carrera (sensores de posición para cilindros):**

- Detectan la posición del vástago (dentro/fuera)
- Montaje: en el cuerpo del cilindro, en ranura longitudinal
- Tipos: magnéticos (imán en el pistón), mecánicos
- Salida: PNP (más común) o NPN, 24 V DC

### 3.7.3 Sistema electroneumático completo



### 3.7.4 Ejemplo de aplicación: mesa transferencia

Una mesa de transferencia mueve piezas de una cinta a otra.

#### Secuencia:

1. Sensor A detecta pieza en cinta de entrada
2. PLC activa solenoide Y1 → cilindro 1 avanza (empuja pieza sobre mesa)
3. Sensor B (cilindro 1 extendido) confirma posición
4. PLC activa Y2 → cilindro 2 (pinza) cierra
5. PLC desactiva Y1 → cilindro 1 retrocede (lleva pieza)
6. PLC activa Y3 → cilindro 3 baja pieza
7. PLC desactiva Y2 → pinza suelta pieza
8. PLC desactiva Y3 → cilindro 3 sube
9. Ciclo completo

#### Programa Ladder para la secuencia:

```
// Peldaño 1: Detectar pieza (inicio de ciclo)
"SensorA"  "Ciclo0cupado"      "IniciarCiclo"
—|/|—————|/|—————( S )—

// Peldaño 2: Avance cilindro 1
"IniciarCiclo"      "Y1_Cil1Avanza"
—|/|—————( )—

// Peldaño 3: Confirmar avance y cerrar pinza
"SensorB"  "IniciarCiclo"      "Y2_PinzaCierra"
—|/|—————|/|—————( S )—

// Peldaño 4: Retroceso cilindro 1
"Y2_PinzaCierra"  "IniciarCiclo"  "TON T1" PT=500ms
—|/|—————|/|—————( TON )—
"T1.Q"      "Reset Y1_Cil1Avanza"
—|/|—————( R )—
"T1.Q"      "Y3_Cil3Baja"
—|/|—————( S )—

// ... continuar con el resto de pasos
```

### 3.7.5 Aplicación real

Las máquinas de montaje automático en la industria electrónica utilizan intensivamente la electroneumática:

- **Pick-and-place:** cilindros para movimiento vertical + ventosas neumáticas
- **Sujeción:** pinzas neumáticas para sujetar componentes
- **Indexado:** mesas rotativas neumáticas con control PLC
- **Corte y marcado:** cilindros para cizallas, punzones

### 3.7.6 Resumen

- **Electroneumática:** integración de sensores, PLC y neumática
- **Electroválvula:** válvula neumática accionada por solenoide (24 V DC)
- **Sensores de cilindro:** detectan posición del vástago (imán en el pistón)
- **Secuencias:** el PLC coordina los movimientos paso a paso

### 3.7.7 Ejercicio

Describe la secuencia electroneumática para un sistema que:

1. Detecta una pieza metálica con un sensor inductivo
2. La eleva con un cilindro vertical
3. La gira 90° con un actuador rotativo neumático
4. La deposita en una segunda cinta transportadora
5. Vuelve a la posición inicial

Incluye: sensores, actuadores, electroválvulas y lógica de control.

\newpage

## CAPÍTULO 3.8: HIDRÁULICA INDUSTRIAL

### 3.8.1 Introducción

La hidráulica utiliza aceite a presión para transmitir potencia y realizar trabajo. Es la tecnología ideal cuando se necesitan fuerzas muy grandes (toneladas) en espacios reducidos.

### Principio de Pascal:

La presión aplicada a un fluido confinado se transmite por igual en todas direcciones. Esto significa que podemos multiplicar la fuerza mediante la diferencia de áreas entre pistones.

### Comparativa neumática vs hidráulica:

Característica	Neumática	Hidráulica
Fluido de trabajo	Aire (compresible)	Aceite (prácticamente incompresible)
Presión típica	6-10 bar	100-350 bar (hasta 700 bar)
Fuerza máxima	Media (unos 3000 kg)	Muy alta (miles de toneladas)
Velocidad	Alta (hasta 2 m/s)	Media (hasta 0.5 m/s)
Precisión de posicionamiento	Baja (aire compresible)	Alta (aceite incompresible)
Mantenimiento	Bajo	Medio (cambios de aceite, filtros)
Coste de instalación	Bajo	Alto
Ruido	Alto (escape de aire)	Bajo

## 3.8.2 Componentes de un sistema hidráulico

### 1. Depósito de aceite:

Almacena el fluido hidráulico. También permite que el aceite se enfríe y que las burbujas de aire escapen.

### 2. Bomba hidráulica:

Convierte energía mecánica (motor eléctrico) en energía hidráulica (caudal a presión). Tipos: bombas de engranajes, de paletas, de pistones.

### 3. Válvula de seguridad (limitadora de presión):

Protege el sistema limitando la presión máxima. Si la presión supera el límite, la válvula deriva aceite al depósito.

### 4. Válvula direccional:

Controla la dirección del flujo de aceite (similar a las válvulas neumáticas). Normalmente 4/3 (4 vías, 3 posiciones) para cilindros de doble efecto.

### 5. Válvula reguladora de caudal:

Controla la velocidad del actuador limitando el caudal de aceite.

## 6. Cilindro hidráulico:

Convierte la presión en movimiento lineal. Similar a los neumáticos pero construidos para presiones mucho más altas (paredes más gruesas, sellos más robustos).

## 7. Motor hidráulico:

Convierte presión y caudal en movimiento rotatorio. Similar a una bomba funcionando al revés.

## 8. Filtros:

Mantienen el aceite limpio. La contaminación es la principal causa de fallos en sistemas hidráulicos.

### 3.8.3 Fuerza en cilindros hidráulicos

La misma fórmula que en neumática:  $F = P \times A$

Pero con presiones mucho mayores.

#### Ejemplo:

Cilindro hidráulico de 100 mm de diámetro (radio = 0.05 m), presión 200 bar.

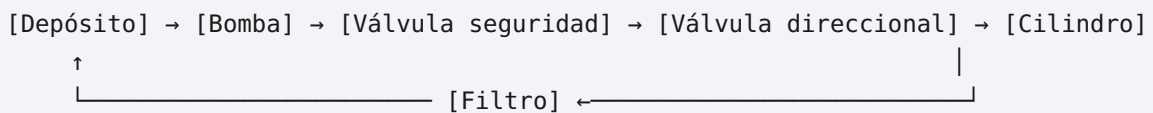
$$A = \pi \times 0.05^2 = 0.00785 \text{ m}^2$$

$$P = 200 \times 100,000 = 20,000,000 \text{ Pa (20 MPa)}$$

$$F = 20,000,000 \times 0.00785 = 157,000 \text{ N}$$

En toneladas:  $157,000 \div 9,800 = 16$  toneladas

### 3.8.4 Circuito hidráulico básico



#### Funcionamiento:

1. La bomba aspira aceite del depósito y lo envía a presión
2. La válvula de seguridad protege contra sobrepresión
3. La válvula direccional dirige el flujo al lado del cilindro que debe moverse
4. El aceite que sale del cilindro vuelve al depósito
5. El filtro limpia el aceite antes de la reentrada

### 3.8.5 Aplicación real

#### **Prensa hidráulica de 500 toneladas:**

- Cilindro principal: 400 mm de diámetro
- Presión de trabajo: 350 bar
- Fuerza máxima:  $3.1416 \times 0.2^2 \times 35,000,000 = 4,398,000 \text{ N} \approx 450 \text{ toneladas}$
- Bomba: 75 kW, caudal 150 l/min
- Aceite: ISO VG 46 (viscosidad media)
- Ciclo: 15 segundos

### 3.8.6 Resumen

- **Hidráulica:** aceite a presión para transmitir fuerza
- **Presiones típicas:** 100-350 bar (10-35 veces mayores que neumática)
- **Fuerzas:** de decenas a miles de toneladas
- **Componentes:** depósito, bomba, válvulas, cilindro/motor, filtros
- **Aplicaciones:** prensas, excavadoras, grúas, maquinaria pesada

### 3.8.7 Ejercicio

Un cilindro hidráulico de 80 mm de diámetro trabaja a 150 bar.

- Calcula la fuerza en el avance.
- ¿Cuántas toneladas son?

(Solución:  $A = \pi \times 0.04^2 = 0.005027 \text{ m}^2$ ;  $P = 15,000,000 \text{ Pa}$ ;  $F = 15,000,000 \times 0.005027 = 75,398 \text{ N}$ ;  $75,398 \div 9800 = 7.7 \text{ toneladas}$ )

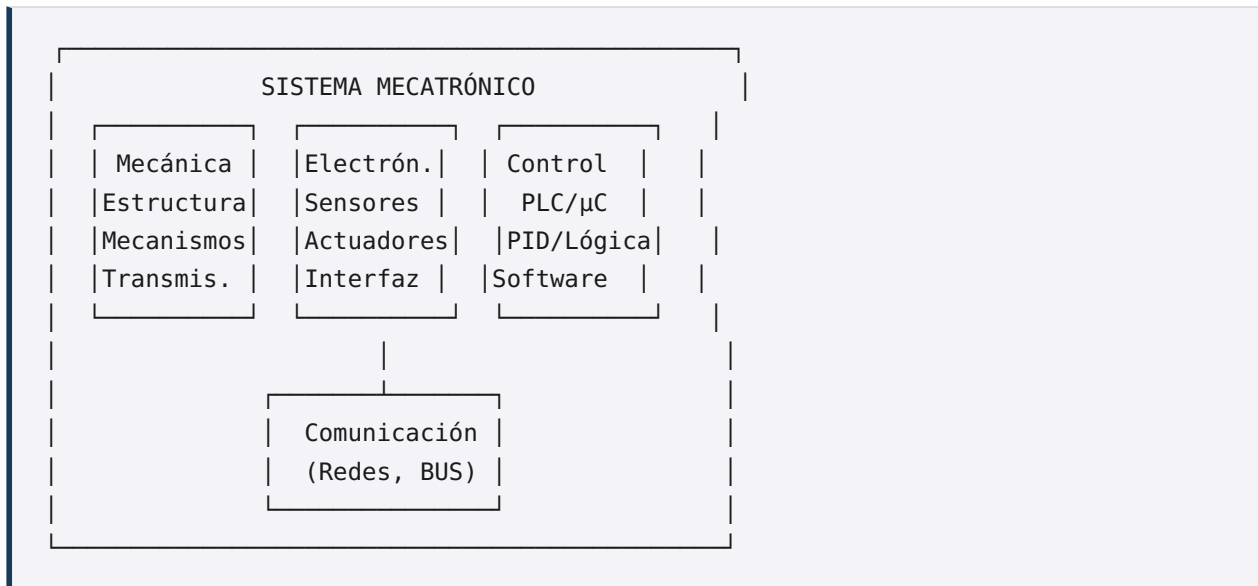
\newpage

## **CAPÍTULO 3.9: SISTEMAS MECATRÓNICOS INTEGRADOS**

### **3.9.1 El enfoque mecatrónico**

Un sistema mecatrónico no es simplemente la suma de partes mecánicas, electrónicas y de software. Es una integración donde todas las disciplinas se diseñan juntas desde el inicio.

#### **Los 4 subsistemas:**



### 3.9.2 Ejemplo 1: Cinta transportadora clasificadora

#### Componentes:

- **Mecánica:** cinta con rodillos, estructura de acero, rampas
- **Sensores:** fotoeléctrico (detección), báscula (peso), inductivo (metal)
- **Actuadores:** motor DC con variador, cilindros neumáticos
- **Control:** PLC Siemens S7-1200
- **Interfaz:** HMI KTP400 (pantalla táctil)

#### Funcionamiento:

1. Un sensor fotoeléctrico detecta la pieza al entrar en la cinta
2. La báscula mide el peso (señal analógica 4-20 mA al PLC)
3. El sensor inductivo detecta si la pieza es metálica
4. El PLC decide el destino según: tipo (metal/plástico) y peso
5. Activa el cilindro neumático correspondiente
6. El cilindro empuja la pieza a la rampa adecuada
7. Un contador registra el total clasificado por cada categoría

### 3.9.3 Ejemplo 2: Máquina de marcado láser

#### Componentes:

- **Mecánica:** mesa XY con husillos de bolas, estructura de pórtico
- **Sensores:** finales de carrera (límites), encoder (posición X-Y), cámara (visión)
- **Actuadores:** servomotores (ejes X,Y), láser de CO<sub>2</sub> (marcado)
- **Control:** PC industrial + controlador CNC
- **Software:** programa G-code, interfaz de operador

### **Funcionamiento:**

1. El operador coloca una pieza en la mesa
2. Pulsa inicio en la HMI
3. La cámara captura la posición exacta de la pieza
4. El software ajusta las coordenadas de marcado
5. Los servomotores posicionan la mesa XY
6. El láser marca el código QR y el texto
7. La mesa vuelve a la posición inicial

### **3.9.4 Integración de tecnologías**

Un sistema mecatrónico moderno integra:

<b>Tecnología</b>	<b>Cuándo usarla</b>	<b>Ventaja</b>
Neumática	Movimientos rápidos, fuerza media	Simple, rápida, económica
Hidráulica	Grandes fuerzas (toneladas)	Potencia, precisión
Servomotores	Posicionamiento preciso	Control exacto, flexible
PLC	Lógica de control, secuencias	Robusto, fiable, programable
HMI	Interfaz con operario	Visualización, control
Red industrial	Comunicación entre dispositivos	Datos compartidos, sincronía

### **3.9.5 El ciclo de diseño mecatrónico**

1. **Definir requisitos:** qué debe hacer el sistema (velocidad, precisión, coste)
2. **Diseño conceptual:** elegir tecnologías adecuadas
3. **Simulación:** verificar funcionamiento antes de construir
4. **Diseño detallado:** dimensionar componentes, crear planos
5. **Implementación:** construir, cablear, programar
6. **Puesta en marcha:** probar, ajustar, verificar
7. **Mantenimiento:** operar, diagnosticar, reparar

### **3.9.6 Resumen**

- **Sistema mecatrónico:** integración de mecánica, electrónica, control y software
- **Las disciplinas se diseñan juntas,** no por separado

- **Ejemplos:** cintas clasificadoras, máquinas CNC, robots, líneas de envasado
- **Tecnologías combinadas:** neumática, hidráulica, servos, PLC, HMI, redes

### 3.9.7 Ejercicio

Identifica los 4 subsistemas mecatrónicos (mecánica, sensores, actuadores, control) en un ascensor moderno. Describe brevemente los componentes de cada subsistema.

\newpage

---

## NIVEL 4: AVANZADO PROFESIONAL

---

*“Nivel profesional: robótica, automatización avanzada, Industria 4.0.”*

Has llegado al nivel más alto. Aquí verás robótica industrial, sistemas SCADA, integración de fábricas completas y las tecnologías de la Industria 4.0.

---

### CAPÍTULO 4.1: ROBÓTICA INDUSTRIAL — INTRODUCCIÓN

#### 4.1.1 ¿Qué es un robot industrial?

Un robot industrial es un manipulador programable multiuso con varios grados de libertad, diseñado para mover materiales, piezas, herramientas o dispositivos en operaciones industriales.

**Definición ISO 8373:** “Manipulador programable multiuso con varios grados de libertad, capaz de manejar materiales, piezas, herramientas o dispositivos especializados mediante movimientos variables programados para la realización de tareas diversas.”

#### **Componentes de un robot industrial:**

1. **Controlador:** computadora industrial que ejecuta el programa y controla los servomotores
2. **Servomotores:** proporcionan el movimiento de cada articulación con feedback de posición (encoder)
3. **Estructura mecánica:** eslabones y articulaciones que forman el brazo
4. **Efecto final (end-effector):** herramienta acoplada al robot (pinza, soldador, cámara, pintor)
5. **Sensores:** encoders (posición), finales de carrera, sensores de fuerza, visión artificial

#### 4.1.2 Grados de libertad (GDL)

Cada movimiento independiente que puede realizar el robot es un grado de libertad.

GDL	Capacidad	Ejemplo
1	Movimiento en un solo eje	Cinta transportadora, motor lineal
2	Movimiento en un plano (X, Y)	Mesa XY
3	Posición en el espacio 3D (X, Y, Z)	Grúa cartesiana
4	Posición 3D + orientación de herramienta	SCARA
5	Posición 3D + orientación parcial	Robot de soldadura simple
6	Posición 3D + orientación completa	Brazo robótico industrial estándar

### Los 6 GDL de un brazo robótico articulado:

1. **J1 (Base):** rotación del robot sobre su base (giro horizontal)
2. **J2 (Hombro):** elevación del brazo completo
3. **J3 (Codo):** extensión del antebrazo
4. **J4 (Muñeca 1):** rotación de la muñeca
5. **J5 (Muñeca 2):** inclinación de la muñeca
6. **J6 (Muñeca 3):** rotación final de la herramienta

#### 4.1.3 Tipos de robots industriales

Tipo	Configuración	Aplicaciones típicas
<b>Cartesiano (XYZ)</b>	3 ejes lineales perpendiculares	Pick-and-place, CNC, montaje
<b>SCARA</b>	2 articulaciones rotatorias + 1 lineal	Ensamblaje de componentes electrónicos
<b>Articulado (6 ejes)</b>	Brazo con 6 articulaciones rotatorias	Soldadura, pintura, manipulación
<b>Delta (paralelo)</b>	3 brazos conectados a base común	Empaquetado rápido (alta velocidad)
<b>Colaborativo (cobot)</b>	Similar a articulado, con sensores de seguridad	Trabajo junto a humanos, ensamblaje ligero

#### 4.1.4 Especificaciones técnicas típicas

Parámetro	Robot pequeño	Robot mediano	Robot grande
Carga máxima	3-10 kg	10-50 kg	50-500 kg

Parámetro	Robot pequeño	Robot mediano	Robot grande
Alcance	0.5-1 m	1-2 m	2-4 m
Repetibilidad	±0.02 mm	±0.05 mm	±0.1 mm
Velocidad máxima	2 m/s	2 m/s	1.5 m/s
Peso del robot	50 kg	300 kg	1500 kg

#### 4.1.5 Aplicación real: robot soldador KUKA

Un robot KUKA KR 6 (robot articulado de 6 ejes) en una línea de fabricación de automóviles:

- Carga máxima: 6 kg
- Alcance máximo: 1.6 m
- Repetibilidad: ±0.05 mm
- Velocidad máxima: 2 m/s
- Programa de soldadura: 5000 puntos de soldadura por coche
- Tiempo de ciclo: 60 segundos (soldadura completa de una puerta)
- Comunicación: Profinet con el PLC de la línea

#### 4.1.6 Resumen

- **Robot industrial:** manipulador programable multiuso
- **GDL:** número de movimientos independientes
- **6 GDL:** posicionamiento y orientación completos en 3D
- **Tipos:** cartesiano, SCARA, articulado, delta, colaborativo
- **Repetibilidad:** capacidad de volver al mismo punto (µm o mm)

#### 4.1.7 Ejercicio

¿Cuántos grados de libertad tiene? a) Un brazo humano (sin contar los dedos), b) Una grúa torre de construcción, c) El brazo de un robot de soldadura de 6 ejes.

\newpage

## CAPÍTULO 4.2: CINEMÁTICA DE ROBOTS

### 4.2.1 Cinemática directa e inversa

**Cinemática directa:** Dados los ángulos de las articulaciones, calcular la posición y orientación del efector final.

**Cinemática inversa:** Dada una posición y orientación deseadas del efector final, calcular los ángulos necesarios de cada articulación.

### 4.2.2 Sistemas de coordenadas

**Sistema de coordenadas cartesiano (X, Y, Z):**

- X: eje horizontal (izquierda-derecha)
- Y: eje horizontal (adelante-atrás)
- Z: eje vertical (arriba-abajo)

**Sistema de coordenadas articular (J1, J2, J3, ...):**

- Cada valor representa el ángulo de una articulación

### 4.2.3 Cinemática directa para un robot de 2 articulaciones

$$\begin{aligned} X &= L1 \cdot \cos(\theta1) + L2 \cdot \cos(\theta1 + \theta2) \\ Y &= L1 \cdot \sin(\theta1) + L2 \cdot \sin(\theta1 + \theta2) \end{aligned}$$

Donde:

- L1, L2 = longitudes de los eslabones
- $\theta1$ ,  $\theta2$  = ángulos de las articulaciones

**Ejemplo:** L1 = 30 cm, L2 = 20 cm,  $\theta1 = 45^\circ$ ,  $\theta2 = 30^\circ$

- $X = 30 \cdot 0.707 + 20 \cdot 0.259 = 21.21 + 5.18 = 26.39$  cm
- $Y = 30 \cdot 0.707 + 20 \cdot 0.966 = 21.21 + 19.32 = 40.53$  cm

### 4.2.4 Cinemática inversa para un robot de 2 articulaciones

$$\begin{aligned} \cos(\theta2) &= (X^2 + Y^2 - L1^2 - L2^2) / (2 \cdot L1 \cdot L2) \\ \theta1 &= \text{atan2}(Y, X) - \text{atan2}(L2 \cdot \sin(\theta2), L1 + L2 \cdot \cos(\theta2)) \end{aligned}$$

La cinemática inversa puede tener:

- **0 soluciones:** el punto está fuera del alcance del robot

- **1 solución:** el punto está en el límite del alcance
- **2 soluciones:** configuración “codo arriba” y “codo abajo”
- **Infinitas soluciones:** el robot tiene más GDL de los necesarios (redundancia)

#### 4.2.5 Aplicación real

En la programación de robots industriales, el operario suele trabajar en coordenadas cartesianas (X, Y, Z) porque son más intuitivas. El controlador del robot resuelve internamente la cinemática inversa para convertir esas coordenadas en ángulos de articulación.

#### 4.2.6 Resumen

- **Cinemática directa:** ángulos → posición
- **Cinemática inversa:** posición → ángulos
- La cinemática inversa puede tener múltiples soluciones
- El controlador del robot resuelve la cinemática internamente

#### 4.2.7 Ejercicio

Un robot de 2 ejes tiene  $L_1 = 40$  cm,  $L_2 = 30$  cm. Si  $\theta_1 = 30^\circ$  y  $\theta_2 = 60^\circ$ , calcula la posición (X, Y) de la punta. ( $\cos 30 = 0.866$ ,  $\sin 30 = 0.5$ ,  $\cos 90 = 0$ ,  $\sin 90 = 1$ )

(Solución:  $X = 40 \times 0.866 + 30 \times 0 = 34.64$  cm;  $Y = 40 \times 0.5 + 30 \times 1 = 20 + 30 = 50$  cm)

\newpage

## CAPÍTULO 4.3: BRAZOS ROBÓTICOS Y CONTROL

### 4.3.1 Estructura de un brazo robótico de 6 ejes

Articulación	Rango típico	Velocidad típica	Componente
J1 (Base)	$\pm 180^\circ$	150°/s	Servomotor + reductor planetario
J2 (Hombro)	$\pm 135^\circ$	150°/s	Servomotor + reductor RV
J3 (Codo)	$\pm 150^\circ$	175°/s	Servomotor + reductor RV

Articulación	Rango típico	Velocidad típica	Componente
J4 (Muñeca rot.)	±270°	250°/s	Servomotor armónico
J5 (Muñeca incl.)	±120°	250°/s	Servomotor armónico
J6 (Muñeca rot.)	±360°	300°/s	Servomotor armónico

### 4.3.2 Reductores de precisión

Los servomotores giran a alta velocidad (3000-6000 RPM) con poco par. Necesitan reductores para:

- Aumentar el par (fuerza de giro)
- Reducir la velocidad
- Aumentar la precisión (eliminar holguras)

Tipo	Relación	Rendimiento	Holgura	Aplicación
Planetario	3:1 a 10:1	95%	5-10 min	Articulación pequeña
Cicloidal	30:1 a 100:1	90%	1-3 min	Articulación media
Armónico	50:1 a 160:1	85%	0-1 min	Muñeca (cero holgura)
RV (Rot Vector)	30:1 a 160:1	95%	<1 min	Articulación principal

### 4.3.3 Trayectorias de movimiento

Tipo	Descripción	Ventaja	Desventaja
PTP (Point to Point)	Movimiento articular directo	Más rápido	No controla el camino
LIN (Lineal)	Línea recta en el espacio	Trayectoria predecible	Más lento
CIRC (Circular)	Arco circular	Preciso en curvas	Complejo de programar

### 4.3.4 Programación de robots

#### Teach pendant (colgante de programación):

Es el método más común. El operario:

1. Mueve el robot manualmente con el teach pendant

2. Guarda puntos de la trayectoria
3. Define velocidades, zonas de aproximación, lógica
4. Prueba el programa en modo lento
5. Ajusta y ejecuta en modo automático

### **Programación offline:**

Se programa en un PC con software de simulación (RobotStudio, KUKA.Sim).

Ventajas:

- No se detiene la producción para programar
- Se pueden simular colisiones
- Se optimizan trayectorias

### **Lenguajes propietarios:**

- KRL (KUKA Robot Language)
- RAPID (ABB)
- Inform (FANUC)
- AS (Mitsubishi)

## **4.3.5 Aplicación real: programa de pick-and-place**

```
; KUKA KRL - Pick and Place
DEF PICK_PLACE( )
  ; Configuración inicial
  BAS(#INITMOV,0)

  ; Ir a home
  PTP HOME

  ; Aproximación a la pieza
  LIN {X 500, Y 0, Z 200, A 0, B 0, C 0} C_PTP

  ; Bajar a la pieza
  LIN {X 500, Y 0, Z 50, A 0, B 0, C 0}

  ; Cerrar pinza
  OUT 1, TRUE
  WAIT SEC 0.5

  ; Subir la pieza
  LIN {X 500, Y 0, Z 200, A 0, B 0, C 0}

  ; Mover a la posición de colocación
  LIN {X 800, Y 200, Z 200, A 0, B 0, C 0}
  LIN {X 800, Y 200, Z 50, A 0, B 0, C 0}
```

```
; Abrir pinza
OUT 1, FALSE
WAIT SEC 0.5

; Retirada
LIN {X 800, Y 200, Z 200, A 0, B 0, C 0}

; Volver a home
PTP HOME
END
```

### 4.3.6 Resumen

- **6 ejes:** articulaciones con servomotor + reductor
- **Reductores:** aumentan par, reducen velocidad, eliminan holgura
- **Trayectorias:** PTP (rápido), LIN (recto), CIRC (curvo)
- **Programación:** teach pendant, offline, lenguaje propio

### 4.3.7 Ejercicio

¿Por qué es necesario usar reductores en las articulaciones de un robot industrial?  
¿Qué pasaría si conectáramos el servomotor directamente a la articulación?

\newpage

## CAPÍTULO 4.4: CONTROL DE PRECISIÓN Y SERVO SISTEMAS

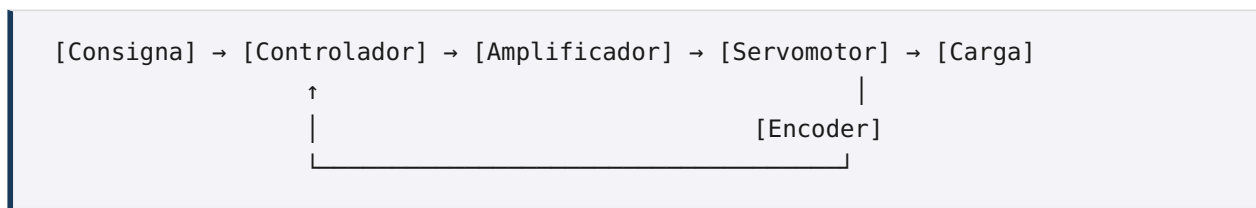
### 4.4.1 El servosistema

Un servosistema es un sistema de control realimentado que controla posición, velocidad o par de un motor.

#### Componentes:

1. **Controlador:** compara la consigna con la medición y calcula la corrección
2. **Amplificador (driver):** convierte la señal de control en potencia para el motor
3. **Servomotor:** motor diseñado para control preciso (DC brushless, AC síncrono)
4. **Encoder:** sensor que mide la posición real del motor (resolución: 1000-10000 pulsos/vuelta)

## Diagrama de bloques:



### 4.4.2 Control PID

PID = Proporcional + Integral + Derivativo. Es el algoritmo de control más utilizado en la industria.

#### Ecuación:

$$\text{Salida}(t) = K_p \cdot e(t) + K_i \cdot \int e(t) dt + K_d \cdot de(t)/dt$$

Donde:

- $e(t)$  = error = consigna - valor medido
- $K_p$  = ganancia proporcional
- $K_i$  = ganancia integral
- $K_d$  = ganancia derivativa

#### Efecto de cada término:

Término	Efecto en el sistema	Si es demasiado alto
<b>Proporcional (Kp)</b>	Responde al error actual. Reduce el tiempo de subida.	Oscilación, inestabilidad
<b>Integral (Ki)</b>	Corrige errores acumulados. Elimina el error en estado estacionario.	Sobreoscilación, lentitud
<b>Derivativo (Kd)</b>	Anticipa el error futuro. Aumenta la estabilidad.	Ruido amplificado, inestabilidad

### 4.4.3 Sintonización práctica (Ziegler-Nichols)

1. Poner  $K_i = 0$ ,  $K_d = 0$
2. Aumentar  $K_p$  gradualmente hasta que el sistema oscile de forma sostenida
3. Anotar esa ganancia ( $K_u$  = ganancia última)
4. Medir el período de oscilación ( $T_u$ )

5. Calcular los parámetros:

- $K_p = 0.6 \times K_u$
- $K_i = 2 \times K_p / T_u$
- $K_d = K_p \times T_u / 8$

#### 4.4.4 Control en cascada

En robótica se usa control en cascada: un lazo externo de posición, uno intermedio de velocidad y uno interno de corriente/par.

```
[Posición] → [PID pos] → [Velocidad] → [PID vel] → [Corriente] → [PID corr] → [Motor]
```

Cada lazo interno es más rápido que el externo. Típicamente:

- Lazo de corriente: cada 0.1 ms
- Lazo de velocidad: cada 0.5 ms
- Lazo de posición: cada 1-2 ms

#### 4.4.5 Aplicación real

Un robot de soldadura debe seguir una trayectoria con precisión de  $\pm 0.5$  mm. El controlador calcula la posición deseada 1000 veces por segundo, y el PID de cada articulación ajusta el voltaje del servomotor para corregir el error.

#### 4.4.6 Resumen

- **Servosistema:** motor + encoder + controlador PID
- **PID:** P (reacción), I (precisión), D (estabilidad)
- **Sintonización:** método de Ziegler-Nichols
- **Control en cascada:** posición → velocidad → corriente

#### 4.4.7 Ejercicio

Un horno debe mantenerse a 200 °C. Tiene  $K_p = 10$ ,  $K_i = 0.5$ ,  $K_d = 0.1$ . La temperatura actual es 180 °C, el error anterior era 25 °C y han pasado 2 segundos. Calcula el término proporcional y el derivativo de la salida PID.

(Solución: Error = 200 - 180 = 20; P = 10 × 20 = 200; D = 0.1 × (20-25)/2 = 0.1 × (-2.5) = -0.25; La contribución D frena el sobrecalentamiento)

\newpage

## CAPÍTULO 4.5: LÍNEAS DE PRODUCCIÓN AUTOMATIZADAS

### 4.5.1 Concepto de línea de producción

Una línea de producción es una secuencia organizada de estaciones de trabajo donde se realizan operaciones para transformar materias primas en productos terminados.

#### Tipos de líneas:

Tipo	Característica	Ejemplo
Transferencia	El producto se mueve automáticamente entre estaciones	Línea de montaje de motores
Ensamblaje	Se añaden componentes progresivamente	Línea de montaje de teléfonos
Proceso continuo	El material fluye sin interrupción	Refinería de petróleo
Lote	Producción por lotes de productos diferentes	Farmacéutica

### 4.5.2 Elementos de una línea automatizada

Elemento	Función	Ejemplo
<b>Cinta transportadora</b>	Transportar producto entre estaciones	Cinta de rodillos, banda
<b>Robot</b>	Realizar operaciones sobre el producto	Soldar, pintar, montar
<b>Estación de trabajo</b>	Punto donde se realiza una operación	Prensa, taladro
<b>PLC</b>	Controlar cada estación	Siemens S7-1500
<b>HMI</b>	Interfaz de operación	Pantalla táctil
<b>Sensores</b>	Verificar estado del producto	Cámaras de visión
<b>Almacén intermedio</b>	Buffer entre estaciones	Acumulador de piezas

### 4.5.3 Ejemplo: línea de envasado de bebidas

[Entrada] → [Soplado] → [Llenado] → [Etiquetado] → [Encapsulado] → [Empaquetado]  
→ [Paletizado]

Botellas de 96 PET	Plástico + calor	Líquido + CO <sub>2</sub>	Etiqueta adhesiva	Tapón rosca	Caja de 12 film	Palé flejes
--------------------------	---------------------	------------------------------	----------------------	----------------	--------------------	----------------

#### Parámetros de la línea:

- Velocidad: 36,000 botellas/hora (10 botellas/segundo)
- Takt time: 0.36 segundos por botella
- Estaciones: 12 (sin contar transporte)
- PLCs: 12 (uno por estación), comunicados por Profinet
- SCADA: WinCC supervisión central

### 4.5.4 Conceptos clave

**Takt time:** es el tiempo máximo permitido para producir una pieza. Marca el ritmo de la línea.

Takt time = Tiempo disponible / Demanda del cliente  
Ejemplo: 8 horas = 28,800 s. Demanda = 1000 piezas.  
Takt time = 28,800 / 1000 = 28.8 segundos/pieza

**OEE (Overall Equipment Effectiveness):** métrica global de eficiencia.

OEE = Disponibilidad × Rendimiento × Calidad

- Disponibilidad: (Tiempo operativo - Paradas) / Tiempo operativo
- Rendimiento: (Piezas reales × Takt time) / Tiempo operativo
- Calidad: Piezas buenas / Piezas totales

OEE de clase mundial: > 85%

**Cuello de botella:** la estación más lenta de la línea. Determina la velocidad máxima de toda la línea. Mejorar el cuello de botella mejora toda la línea.

### 4.5.5 Ley de Little (para buffers)

El inventario en un sistema de producción está relacionado con el tiempo de ciclo:

$$\text{WIP} = \text{Throughput} \times \text{Tiempo de ciclo}$$

Donde WIP = Work In Progress (producto en proceso).

#### 4.5.6 Aplicación real

Línea de montaje de automóviles (ejemplo Volkswagen):

- 400 estaciones de trabajo
- 800 robots
- Takt time: 61 segundos por coche
- Producción: ~1400 coches/día (2 turnos)
- OEE objetivo: 90%
- Parada de 1 minuto = 1 coche perdido

#### 4.5.7 Resumen

- **Línea de producción:** secuencia de estaciones para fabricar productos
- **Takt time:** ritmo de producción marcado por la demanda
- **OEE:** disponibilidad  $\times$  rendimiento  $\times$  calidad
- **Cuello de botella:** estación que limita toda la línea

#### 4.5.8 Ejercicio

Una línea produce 120 piezas por hora en 3 estaciones en serie:

- Estación A: 45 segundos/pieza
- Estación B: 30 segundos/pieza
- Estación C: 40 segundos/pieza

a) ¿Cuál es el cuello de botella? b) ¿Cuál es el takt time real? c) ¿Qué producción máxima se puede alcanzar?

(Soluciones: a) Estación A (45s); b) 45 segundos; c)  $3600/45 = 80$  piezas/hora)

\newpage

## CAPÍTULO 4.6: SCADA — SUPERVISIÓN Y CONTROL

### 4.6.1 ¿Qué es SCADA?

SCADA = Supervisory Control And Data Acquisition. Es un sistema informático que permite supervisar y controlar procesos industriales desde una computadora central, proporcionando información en tiempo real del estado de la planta.

#### Diferencias entre HMI y SCADA:

Característica	HMI	SCADA
Alcance	Una máquina	Planta completa
Número de PLCs	1-2	Decenas o cientos
Históricos	Limitados	Extensos (años)
Alarmas	Locales	Centralizadas
Usuarios	1-2 operarios	Múltiples (operarios, supervisores, ingenieros)
Base de datos	No	Sí (SQL, Oracle)
Coste	500-3000 €	10,000-100,000 €

### 4.6.2 Componentes de un sistema SCADA

#### 1. Servidor SCADA:

Computadora central que ejecuta el software SCADA. Recoge datos de los PLC, los procesa, los almacena y los presenta en pantallas HMI.

#### 2. Estaciones de operación (HMI):

Pantallas donde los operarios ven el proceso y pueden intervenir.

#### 3. PLC / RTU (Remote Terminal Unit):

Controladores de campo que recogen datos de sensores y ejecutan órdenes.

#### 4. Red de comunicación:

Conecta todos los componentes. Puede ser:

- Ethernet industrial (Profinet, EtherNet/IP)
- Fibra óptica
- Radio / 4G / 5G
- RS-485 (Modbus)

## 5. Base de datos:

Almacena históricos de producción, alarmas, eventos y recetas.

### 4.6.3 Funciones de un SCADA

Función	Descripción	Ejemplo
<b>Supervisión</b>	Mostrar estado en tiempo real	Ver temperatura de hornos en pantalla
<b>Control</b>	Enviar órdenes a los PLCs	Arrancar/parar motores remotamente
<b>Alarmas</b>	Notificar condiciones anormales	“Temperatura excede límite máximo”
<b>Históricos</b>	Guardar datos para análisis	Producción por hora/día/mes
<b>Tendencias</b>	Gráficos de evolución de variables	Presión en el tiempo
<b>Recetas</b>	Cargar parámetros para diferentes productos	Temperatura y tiempo para cada material
<b>Informes</b>	Reportes automáticos	OEE diario, producción semanal

### 4.6.4 Software SCADA común

Software	Fabricante	Plataforma	Notas
WinCC	Siemens	Windows	Integración con TIA Portal
FactoryTalk	Allen Bradley	Windows	Integración con Studio 5000
Ignition	Inductive Automation	Multiplataforma	Web-based, moderno
Movicon	Progea	Windows	Flexible, SCADA/HMI
Vijeo Citect	Schneider	Windows	Para grandes instalaciones
GENESIS64	ICONICS	Windows	3D, realidad aumentada

### 4.6.5 Aplicación real: parque eólico

Un parque eólico con 50 aerogeneradores:

- Cada aerogenerador tiene su propio PLC
- Los PLC se comunican por fibra óptica al centro de control
- El servidor SCADA (Ignition) procesa los datos

- Pantallas: mapa del parque, producción total, estado de cada generador
- Alarmas: “Aerogenerador 23: sobrecalentamiento en multiplicadora”
- Históricos: producción diaria, mensual, anual, kWh totales
- Informes automáticos: producción, disponibilidad, ingresos

#### 4.6.6 Resumen

- **SCADA:** sistema central de supervisión y control de planta
- **Componentes:** servidor, HMI, PLCs, red, base de datos
- **Funciones:** supervisar, controlar, alarmas, históricos, tendencias, informes
- **SCADA cubre toda la planta** (el HMI solo una máquina)

#### 4.6.7 Ejercicio

Explica la diferencia entre el HMI de una máquina individual y el SCADA de una planta completa. ¿Pueden coexistir ambos en la misma instalación?

\newpage

## CAPÍTULO 4.7: INDUSTRIA 4.0

### 4.7.1 Las cuatro revoluciones industriales

Revolución	Período	Tecnología clave	Cambio principal
1.0	~1784	Máquina de vapor	Mecanización de la producción
2.0	~1870	Electricidad	Producción en masa
3.0	~1969	Electrónica, PLC	Automatización
4.0	Hoy	IoT, IA, Cloud	Digitalización y conexión

### 4.7.2 Pilares tecnológicos de la Industria 4.0

Tecnología	Descripción	Aplicación en fábrica
<b>IIoT (Industrial IoT)</b>	Sensores conectados a internet	Monitoreo remoto de máquinas

Tecnología	Descripción	Aplicación en fábrica
<b>Big Data</b>	Análisis de grandes volúmenes de datos	Predicción de fallos, optimización
<b>Cloud Computing</b>	Almacenamiento y procesamiento en la nube	SCADA cloud, datos accesibles globalmente
<b>Digital Twin</b>	Gemelo digital (réplica virtual de la máquina)	Simular cambios sin riesgo
<b>Inteligencia Artificial</b>	Sistemas que aprenden de datos	Optimización automática, control adaptativo
<b>Ciberseguridad</b>	Protección de sistemas industriales	Prevención de ataques a infraestructuras
<b>Realidad Aumentada</b>	Información virtual sobre el mundo real	Mantenimiento asistido, formación
<b>5G</b>	Comunicación inalámbrica de baja latencia	AGVs, robots móviles, control remoto
<b>Blockchain</b>	Registro distribuido e inalterable	Trazabilidad de productos, certificaciones
<b>Edge Computing</b>	Procesamiento cerca de los datos (no en la nube)	Tiempo real, baja latencia, privacidad

### 4.7.3 Conceptos clave

#### **CPS (Cyber-Physical Systems):**

Integración de procesos físicos con computación y comunicaciones. Un robot no es solo una máquina, sino un sistema ciberfísico que recibe datos, los procesa, actúa y se comunica.

#### **M2M (Machine to Machine):**

Comunicación directa entre máquinas sin intervención humana. Las máquinas negocian la producción, se ajustan mutuamente y reportan su estado.

#### **MES (Manufacturing Execution System):**

Sistema que gestiona y monitoriza la producción en tiempo real. Está entre el ERP (planificación empresarial) y el SCADA/PLC (control).

ERP (SAP, Oracle) → MES → SCADA → PLC → Máquina

### Digital Twin (Gemelo digital):

Réplica virtual exacta de un sistema físico. Permite:

- Simular cambios sin afectar la producción real
- Predecir comportamiento y fallos
- Formar a operarios sin riesgo

#### 4.7.4 Ejemplo: fábrica inteligente de componentes

Una fábrica 4.0 de componentes de automoción:

1. Cada pieza tiene un **código RFID** (identificación por radiofrecuencia)
2. Las máquinas se comunican entre sí y **auto-ajustan parámetros**
3. Los **robots colaborativos** trabajan junto a operarios sin vallas
4. **Mantenimiento predictivo:** sensores de vibración y temperatura avisan antes de un fallo
5. La producción se **reconfigura en minutos** para diferentes productos
6. **Trazabilidad total:** cada pieza sabe su historial completo
7. **Optimización energética:** las máquinas apagan sistemas no necesarios

#### 4.7.5 Ventajas de la Industria 4.0

Beneficio	Descripción	Impacto típico
Flexibilidad	Producir lotes pequeños al coste de grandes series	Reducción de costes de cambio 50%
Mantenimiento predictivo	Menos paradas no planificadas	Reducción de downtime 30%
Trazabilidad total	Saber historia completa de cada producto	Mejora calidad, recall rápido
Eficiencia energética	Optimización automática del consumo	Ahorro energético 15%
Personalización	Cada producto puede ser diferente	Nuevos modelos de negocio

#### 4.7.6 Resumen

- **Industria 4.0:** digitalización y conexión de la producción
- **Pilares:** IIoT, Big Data, Cloud, IA, Digital Twin, Ciberseguridad
- **CPS = Sistemas Ciberfísicos:** integración de lo físico y lo digital

- **Objetivo:** producción flexible, eficiente, personalizada y conectada

### 4.7.7 Ejercicio

¿Cómo aplicarías el concepto de Gemelo Digital (Digital Twin) a una línea de envasado de bebidas? Describe qué datos recogerías, qué simularías y qué ventajas obtendrías.

\newpage

## CAPÍTULO 4.8: INTEGRACIÓN TOTAL DE SISTEMAS

### 4.8.1 La pirámide de automatización

```
NIVEL 4: ERP (Enterprise Resource Planning)
  | SAP, Oracle, Microsoft Dynamics
  | Planificación empresarial, finanzas, compras, ventas
  ▼
NIVEL 3: MES (Manufacturing Execution System)
  | Gestión de producción, calidad, trazabilidad, mantenimiento
  ▼
NIVEL 2: SCADA / Control de Supervisión
  | Supervisión, alarmas, históricos, tendencias, informes
  ▼
NIVEL 1: PLC / Controladores
  | Control en tiempo real de máquinas y procesos
  ▼
NIVEL 0: Sensores y Actuadores
  | Finales de carrera, fotocélulas, válvulas, motores, cilindros
```

#### Comunicación entre niveles:

- Nivel 0-1: cableado directo (24V DC, 4-20 mA, 0-10 V)
- Nivel 1-2: red industrial (Profinet, Ethernet/IP, Modbus TCP)
- Nivel 2-3: OPC UA (protocolo estándar de comunicación)
- Nivel 3-4: servicios web, bases de datos, APIs REST

## 4.8.2 Protocolos de comunicación industrial

Protocolo	Velocidad	Medio	Fabricante	Uso principal
Profinet	100 Mbps	Ethernet	Siemens	Comunicación PLC-SCADA
EtherNet/IP	100 Mbps	Ethernet	Allen Bradley	Comunicación PLC-SCADA
Modbus TCP	100 Mbps	Ethernet	Abierto	Equipos diversos
Profibus	12 Mbps	RS-485	Siemens	PLC-Sensor (heredado)
Modbus RTU	115 kbps	RS-485	Abierto	Sensores, variadores
OPC UA	Variable	Ethernet/Web	Estándar	SCADA-MES
IO-Link	230 kbps	Cable 3 hilos	Estándar	Sensores inteligentes

## 4.8.3 OPC UA (Unified Architecture)

OPC UA es el estándar moderno de comunicación industrial. Sus características principales:

- **Independiente de plataforma:** funciona en Windows, Linux, dispositivos embebidos
- **Seguro:** cifrado AES, autenticación, certificados
- **Modelado de información:** no solo transmite datos, también su significado y relaciones
- **Descubrimiento:** los dispositivos se encuentran automáticamente en la red
- **Pub/Sub:** comunicación publish/subscribe eficiente para grandes volúmenes

## 4.8.4 Ejemplo: integración completa de una fábrica

### Fábrica de componentes de automoción:

1. **Nivel 0:** 5000 sensores, 2000 actuadores, 100 variadores de velocidad
2. **Nivel 1:** 200 PLCs distribuidos (Siemens S7-1200/1500)
3. **Nivel 2:** 5 servidores SCADA (WinCC), 20 pantallas HMI
4. **Nivel 3:** MES que gestiona órdenes de fabricación, calidad y mantenimiento
5. **Nivel 4:** ERP SAP que gestiona pedidos, compras, finanzas, logística

### Flujo de información ante un pedido:

Cliente hace pedido → ERP crea orden de fabricación  
→ MES planifica la producción  
→ SCADA supervisa la ejecución  
→ PLC ejecuta el programa de la máquina  
→ Sensores confirman que la pieza se ha fabricado correctamente  
→ PLC reporta datos al SCADA  
→ SCADA envía datos de producción al MES  
→ MES actualiza el ERP  
→ ERP genera la factura

#### 4.8.5 Rol del técnico superior en mecatrónica

Un técnico en mecatrónica debe ser capaz de:

1. **Nivel 0-1:** Cablear y configurar sensores y actuadores. Diagnóstico de fallos de campo.
2. **Nivel 1:** Programar PLCs (Ladder, FBD, SFC). Configurar módulos de E/S.
3. **Nivel 1-2:** Configurar comunicaciones (Profinet, Modbus). Interpretar pantallas HMI.
4. **Nivel 2:** Diagnosticar alarmas y eventos en SCADA. Realizar históricos básicos.
5. **Todos los niveles:** Diagnosticar problemas, proponer mejoras, documentar sistemas.

#### Habilidades clave:

- Lectura de planos eléctricos, neumáticos e hidráulicos
- Programación de PLC (al menos un fabricante principal)
- Configuración de redes industriales
- Conocimiento de seguridad funcional
- Capacidad de diagnóstico y resolución de problemas

#### 4.8.6 Resumen

- **Pirámide:** 5 niveles (0: sensores a 4: ERP empresarial)
- **Protocolos:** Profinet, EtherNet/IP, Modbus, OPC UA
- **OPC UA:** estándar moderno de comunicación industrial
- **El técnico en mecatrónica** opera principalmente en niveles 0-2

### 4.8.7 Ejercicio

Describe el flujo de información desde que un cliente hace un pedido por internet hasta que la máquina produce la pieza, indicando qué nivel de la pirámide interviene en cada paso.

\newpage

## CAPÍTULO 4.9: PROYECTO FINAL INTEGRADOR

### 4.9.1 El proyecto: celda de fabricación flexible

Diseña conceptualmente una celda de fabricación que clasifique piezas por material y color, las procese y las empaquete.

### 4.9.2 Componentes del sistema

#### **Mecánica:**

- Cinta transportadora principal (motor DC con variador)
- Ramal de rechazo con cilindro neumático (doble efecto, 50mm diámetro)
- Mesa giratoria para posicionamiento (servomotor)
- Brazo robótico de 4 GDL para pick-and-place

#### **Sensores:**

- Fotoeléctrico (detección de presencia)
- Inductivo (detección de metal)
- Sensor de color (RGB)
- Encoder en la cinta (posición/velocidad)
- Finales de carrera (límites de cilindros)

#### **Actuadores:**

- Motor DC + variador de velocidad (cinta transportadora)
- Electroválvula 5/2 (cilindro neumático de rechazo)
- Electroimán (sujeción de piezas metálicas)
- Servomotores (brazo robótico, 4 unidades)
- Pinza neumática (efector final del robot)

#### **Control:**

- PLC Siemens S7-1200 (lógica principal de la celda)
- Arduino (lectura del sensor de color, comunicación con PLC)

- HMI KTP400 Basic (interfaz de operación)
- SCADA WinCC (supervisión de la celda)

### 4.9.3 Secuencia de funcionamiento

1. [INICIO] Alimentar pieza a la cinta
2. [DETECTAR] Sensor fotoeléctrico detecta pieza → para la cinta
3. [IDENTIFICAR] Sensor inductivo + sensor de color identifican pieza
 

¿Es metálica?	¿Color?	Destino
Sí + Rojo		Caja A
Sí + Azul		Caja B
No (plástico)		Caja C
4. [ACTUAR] PLC activa el actuador correspondiente
5. [REGISTRAR] HMI muestra contadores por tipo
6. [REPETIR] Vuelve al paso 1

### 4.9.4 Lo que has aprendido en este libro

Nivel	Conocimiento adquirido
<b>NIVEL 0</b>	Matemáticas, física y electricidad desde cero
<b>NIVEL 1</b>	Mecánica, electrónica, sensores y actuadores
<b>NIVEL 2</b>	Programación, Arduino, control ON/OFF y secuencial
<b>NIVEL 3</b>	PLC industrial, Ladder, neumática, hidráulica
<b>NIVEL 4</b>	Robótica, servosistemas, SCADA, Industria 4.0

#### Al completar este libro, eres capaz de:

- Entender el funcionamiento de máquinas industriales reales
- Leer y diseñar circuitos electrónicos básicos
- Programar lógica de control en Arduino y Ladder
- Comprender y programar PLCs industriales
- Diseñar sistemas neumáticos e hidráulicos básicos
- Conocer los fundamentos de robótica industrial
- Interpretar sistemas SCADA
- Comprender el contexto de la Industria 4.0

**¡Enhorabuena! Has completado el manual de Mecatrónica desde cero hasta nivel profesional FP Superior.**

\newpage

## APÉNDICES

### APÉNDICE A: GLOSARIO DE TÉRMINOS

Término	Definición
<b>Actuador</b>	Dispositivo que convierte una señal (eléctrica, neumática) en movimiento
<b>Algoritmo</b>	Secuencia ordenada de pasos para resolver un problema
<b>Arduino</b>	Plataforma de hardware libre basada en microcontrolador
<b>Cinemática</b>	Estudio del movimiento sin considerar las fuerzas que lo producen
<b>Cilindro neumático</b>	Actuador que produce movimiento lineal mediante aire comprimido
<b>Condensador</b>	Componente que almacena energía eléctrica en forma de campo eléctrico
<b>Corriente (I)</b>	Flujo de electrones a través de un conductor (amperios)
<b>Electroneumática</b>	Combinación de control eléctrico/electrónico con actuación neumática
<b>Encoder</b>	Sensor que mide posición y/o velocidad de un eje
<b>Engranaje</b>	Rueda dentada que transmite movimiento rotatorio
<b>GDL (Grado de Libertad)</b>	Movimiento independiente de una articulación
<b>Hidráulica</b>	Tecnología que usa aceite a presión para transmitir fuerza
<b>Histéresis</b>	Banda de tolerancia entre activación y desactivación en control ON/OFF
<b>HMI</b>	Interfaz Hombre-Máquina (pantalla táctil para operar máquinas)
<b>Industria 4.0</b>	Cuarta revolución industrial basada en digitalización y conexión
<b>Ladder</b>	Lenguaje gráfico de programación de PLC basado en esquemas de relés

<b>Término</b>	<b>Definición</b>
<b>Ley de Ohm</b>	$V = I \times R$ ; relación entre voltaje, corriente y resistencia
<b>Máquina de estados</b>	Modelo de programación para sistemas con estados y transiciones
<b>Mecatrónica</b>	Integración sinérgica de mecánica, electrónica y control
<b>Neumática</b>	Tecnología que usa aire comprimido para generar movimiento
<b>OEE</b>	Overall Equipment Effectiveness; métrica de eficiencia global
<b>Palanca</b>	Máquina simple que multiplica fuerza mediante un punto de apoyo
<b>PID</b>	Control Proporcional-Integral-Derivativo
<b>PLC</b>	Controlador Lógico Programable; ordenador industrial
<b>PWM</b>	Modulación por Ancho de Pulso; técnica para simular voltajes variables
<b>Regla de tres</b>	Método para calcular un valor desconocido a partir de tres conocidos
<b>SCADA</b>	Sistema de Control de Supervisión y Adquisición de Datos
<b>Sensor</b>	Dispositivo que convierte magnitud física en señal eléctrica
<b>Servomotor</b>	Motor con encoder para control preciso de posición/velocidad/par
<b>Takt time</b>	Ritmo de producción (tiempo máximo por pieza)
<b>Transistor</b>	Semiconductor que funciona como interruptor o amplificador
<b>Variador de frecuencia</b>	Dispositivo que controla la velocidad de motores AC
<b>Voltaje (V)</b>	Diferencia de potencial eléctrico (voltios)

## APÉNDICE B: FÓRMULAS ÚTILES

<b>Área</b>	<b>Fórmula</b>	<b>Descripción</b>
Electricidad	$V = I \times R$	Ley de Ohm
Electricidad	$P = V \times I = I^2 \times R = V^2/R$	Potencia eléctrica
Electricidad	$E = P \times t$	Energía eléctrica
Electricidad	$\tau = R \times C$	Constante de tiempo RC

Área	Fórmula	Descripción
Mecánica	$F = m \times a$	Segunda ley de Newton
Mecánica	$P = m \times g$	Peso (fuerza gravitatoria)
Mecánica	$W = F \times d$	Trabajo mecánico
Mecánica	$P = W / t$	Potencia mecánica
Mecánica	$E_c = \frac{1}{2} \times m \times v^2$	Energía cinética
Mecánica	$E_p = m \times g \times h$	Energía potencial
Geometría	$A = \pi \times r^2$	Área del círculo
Geometría	$P = 2 \times \pi \times r$	Circunferencia
Neumática/Hidráulica	$F = P \times A$	Fuerza de un cilindro
Neumática/Hidráulica	$i = Z_2/Z_1 = D_2/D_1$	Relación de transmisión
Producción	$OEE = D \times R \times C$	Eficiencia global
Producción	$Takt = T. \text{ disponible} / \text{ Demanda}$	Ritmo de producción

## APÉNDICE C: EJERCICIOS ADICIONALES

**Ejercicio C1:** Una cinta transportadora mueve 200 kg a 0.5 m/s. El coeficiente de fricción es 0.3. ¿Qué potencia necesita el motor?

**Ejercicio C2:** Diseña un circuito ladder con dos pulsadores (A y B) donde la salida Motor se activa solo si A está presionado Y B no, y se mantiene al soltar A.

**Ejercicio C3:** Un cilindro neumático de 63 mm de diámetro trabaja a 7 bar. ¿Qué fuerza ejerce en el avance? ¿Y en el retroceso (diámetro del vástago: 20 mm)?

**Ejercicio C4:** Escribe un programa Arduino que lea dos sensores (temperatura en A0, luz en A1). Si temperatura > 35°C Y luz < 100, activa motor en pin 9. Si no, activa LED en pin 13.

**Ejercicio C5:** ¿Qué relación de engranajes necesitas para que un motor de 3000 RPM mueva un eje a 750 RPM? Si el par del motor es 2 Nm, ¿qué par tendrá el eje de salida?

**Ejercicio C6:** Calcula la fuerza de un cilindro hidráulico de 80 mm de diámetro a 200 bar.

**Ejercicio C7:** Una línea tiene 4 estaciones con estos tiempos: 25s, 30s, 20s, 28s. ¿Cuál es el cuello de botella? ¿Cuál es la producción máxima por hora?

**Ejercicio C8:** Escribe en ladder el programa para un semáforo con pulsador peatonal usando temporizadores TON.