

Tema 4. Sistemas de Control y Programación Básica de Robots

En este tema abordarás uno de los pilares fundamentales de la robótica profesional: los sistemas de control y la programación básica. Comprender cómo un robot recibe instrucciones, procesa información y ejecuta movimientos precisos es esencial para cualquier profesional que aspire a trabajar en automatización, industria 4.0 o robótica educativa. A lo largo de estas secciones desarrollarás los conceptos clave, explorarás arquitecturas de control, aprenderás los fundamentos de la programación robótica y aplicarás todo ello en casos prácticos reales.

Sistemas de Control

Definición, tipos y arquitectura de los sistemas que gobiernan el comportamiento robótico.

Programación Robótica

Lenguajes, algoritmos, lógica de programación y ejemplos prácticos aplicados.

Casos Prácticos

Ejercicios reales de programación de movimiento y respuesta a sensores paso a paso.

Recursos y Evaluación

Herramientas digitales, lecturas recomendadas y ejercicios de autoevaluación.

Concepto de Sistema de Control en Robótica

Definición de sistema de control

Un **sistema de control** es el conjunto de componentes físicos y lógicos que tienen como función regular, dirigir y coordinar el comportamiento de un robot para que este alcance un estado deseado o ejecute una tarea específica. En robótica, el sistema de control actúa como el "cerebro" del robot: recibe información del entorno a través de sensores, la procesa mediante algoritmos y emite órdenes a los actuadores (motores, servos, pistones) para producir movimiento o acción.

Técnicamente, un sistema de control puede definirse como un sistema dinámico que regula la salida de un proceso en función de una referencia o consigna. Esta definición, proveniente de la teoría de control clásica, es aplicable tanto a sistemas industriales como a robots educativos. La diferencia entre un robot bien controlado y uno inestable radica precisamente en la calidad y diseño de su sistema de control.

Relación entre control y movimiento

El movimiento de un robot es, en esencia, la manifestación externa del sistema de control. Cada giro de una rueda, cada extensión de un brazo articulado o cada reacción ante un obstáculo tiene su origen en una decisión tomada por el controlador. La relación entre control y movimiento es bidireccional: el controlador ordena el movimiento, pero también monitoriza si dicho movimiento se ha ejecutado correctamente mediante la retroalimentación de los sensores. Este ciclo continuo es lo que permite al robot adaptarse a cambios en el entorno.

En robótica industrial, por ejemplo, un brazo manipulador debe posicionar su extremo con una precisión de décimas de milímetro. Esto solo es posible gracias a un sistema de control que compara constantemente la posición actual (medida por encoders) con la posición deseada y corrige el error en tiempo real. Como futuro profesional de robótica, debes comprender que el control no es un añadido al robot, sino su esencia funcional.

Importancia del control en la automatización

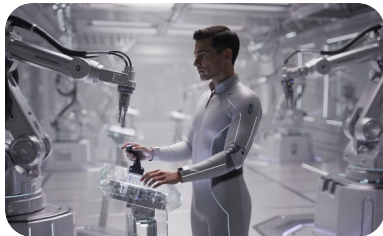
La automatización industrial, que es el contexto donde los robots tienen mayor impacto, depende absolutamente de sistemas de control robustos y fiables. Sin control, un robot no sería más que una máquina estática incapaz de adaptarse. El control permite repetibilidad, precisión, seguridad y eficiencia: las cuatro cualidades que hacen que los robots sean económicamente viables en entornos productivos. Además, los sistemas de control modernos integran inteligencia artificial para aprender de la experiencia y mejorar su rendimiento de forma autónoma.



Ejemplo práctico: En un robot educativo como el LEGO Mindstorms, el sistema de control es el bloque programable (EV3 Brick). Cuando programas al robot para que siga una línea negra, estás definiendo su sistema de control: el sensor de color mide el suelo, el controlador compara la lectura con el valor esperado y ajusta la velocidad de los motores izquierdo y derecho para mantener al robot sobre la línea. Este es un ejemplo real de control en lazo cerrado.

Tipos de Sistemas de Control

Los sistemas de control robótico se clasifican en función de cómo interviene el operador humano y de cómo se genera la señal de control. Comprender estas categorías te permitirá identificar qué tipo de control utiliza cada robot que encuentres en tu carrera profesional, y qué ventajas e inconvenientes presenta cada uno en función del contexto de aplicación.



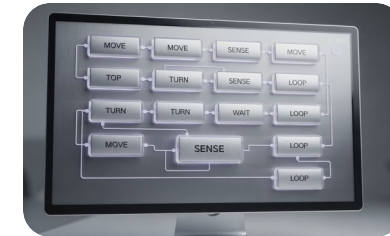
Control Manual

El operador humano dirige directamente el robot mediante mandos, joysticks o interfaces táctiles. Cada movimiento depende de una acción humana en tiempo real. Es característico de robots de teleoperación, como los usados en entornos peligrosos o en cirugía robótica asistida. La precisión depende de la habilidad del operador y no permite altas velocidades ni repetibilidad exacta.



Control Automático

El robot opera sin intervención humana continua, basándose en un programa o algoritmo predefinido combinado con la realimentación de sensores. El controlador toma decisiones autónomas para mantener el sistema en el estado deseado. Es el modelo dominante en la industria manufacturera, donde los robots realizan tareas repetitivas con alta precisión. Utiliza bucles de control como el PID (Proporcional-Integral-Derivativo).



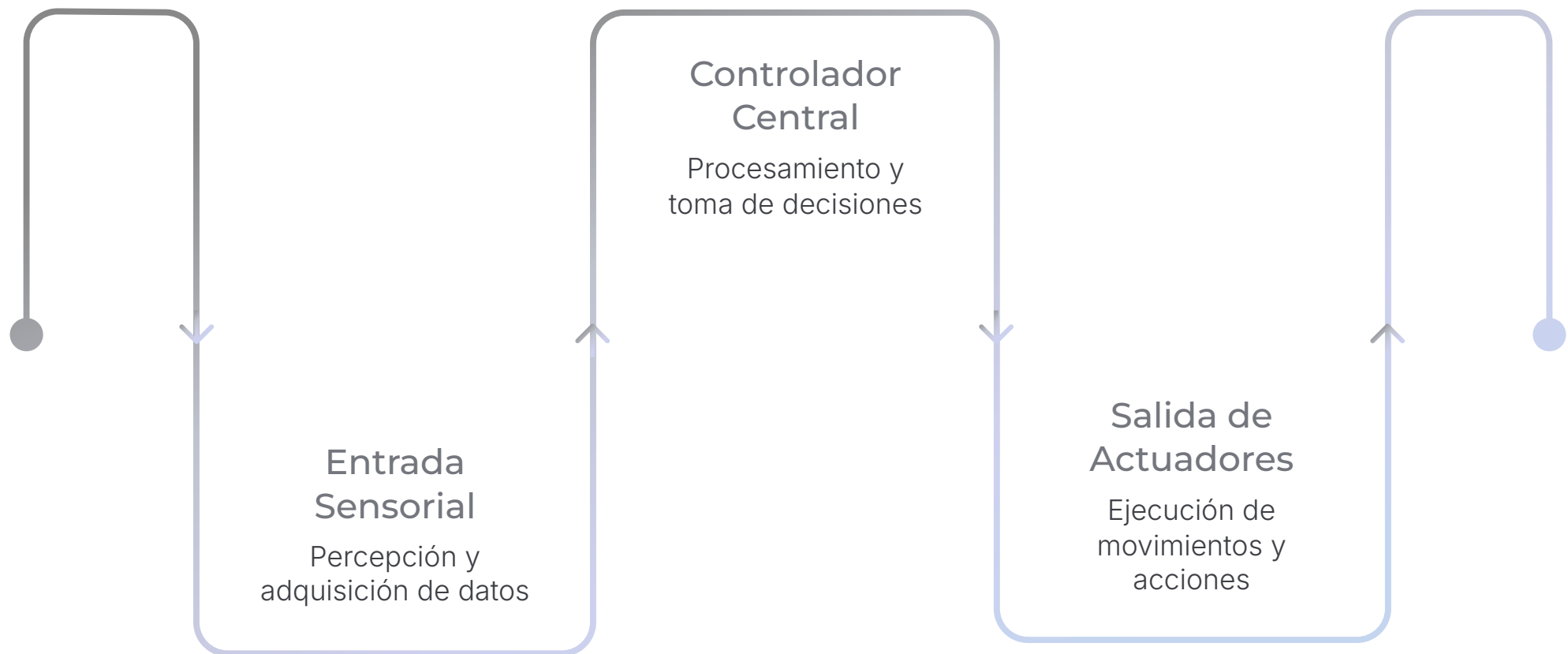
Control Programado

El comportamiento del robot queda definido por un programa que el profesional crea previamente. A diferencia del control automático puro, aquí el énfasis está en la secuencia de instrucciones codificadas. El robot ejecuta exactamente lo que el programador ha especificado. Es el modelo más común en robots educativos y en aplicaciones donde la tarea es conocida y estructurada. Permite gran flexibilidad y adaptabilidad mediante reprogramación.

En la práctica profesional, estos tres tipos de control suelen combinarse. Un robot industrial puede operar en modo automático durante la producción, pasar a control manual para mantenimiento, y ser reconfigurado mediante control programado cuando cambia la tarea productiva. Como profesional de robótica, dominarás los tres modos y sabrás cuándo aplicar cada uno.

Arquitectura de Control de un Robot

La **arquitectura de control** de un robot es la estructura interna que organiza cómo fluye la información desde la percepción del entorno hasta la ejecución del movimiento. Entender esta arquitectura es fundamental para poder diseñar, programar y mantener cualquier sistema robótico. A continuación estudiarás sus tres componentes principales y cómo se relacionan entre sí.



Este diagrama representa el flujo de información en un robot: los sensores capturan datos del entorno, el controlador central los procesa y toma decisiones, y los actuadores ejecutan las acciones resultantes. Este ciclo se repite miles de veces por segundo en robots modernos.

Controlador Central

Es el núcleo computacional del robot. Puede ser un microcontrolador (como Arduino o ESP32), un ordenador de placa reducida (Raspberry Pi) o un PLC industrial. Su función es ejecutar el programa de control, recibir datos de los sensores, tomar decisiones según los algoritmos implementados y enviar señales de mando a los actuadores. La capacidad de procesamiento del controlador determina la complejidad de las tareas que puede realizar el robot.

- Ejecuta el programa de control en tiempo real
- Gestiona la comunicación entre subsistemas
- Almacena parámetros y configuraciones
- Implementa algoritmos de control (PID, lógica difusa, redes neuronales)

Entrada de Sensores

Los sensores son los órganos sensoriales del robot. Convierten magnitudes físicas del entorno (temperatura, distancia, luz, fuerza, posición) en señales eléctricas que el controlador puede interpretar. En robótica se utilizan sensores de muy diversa naturaleza: encoders para medir posición angular, sensores ultrasónicos para medir distancias, cámaras para visión artificial, sensores de presión para manipulación de objetos frágiles, etc.

Salida hacia Actuadores

Los actuadores transforman las señales eléctricas del controlador en movimiento físico. Los más comunes en robótica son los motores DC, los servomotores y los motores paso a paso. También existen actuadores neumáticos e hidráulicos en robótica industrial pesada. La calidad del actuador determina la precisión, velocidad y fuerza del movimiento robótico.

Programación Básica de Robots

Concepto de programación robótica

La **programación robótica** es el proceso mediante el cual defines el comportamiento de un robot a través de instrucciones codificadas que el controlador puede interpretar y ejecutar. Programar un robot significa traducir una tarea humana (como "recoge la pieza A y colócala en la posición B") a un lenguaje formal que el sistema computacional del robot puede procesar. Esta traducción requiere pensar de forma estructurada, secuencial y precisa, desarrollando lo que se conoce como pensamiento computacional.

La programación robótica va más allá de escribir código. Implica comprender el hardware del robot (sensores, actuadores, controlador), modelar el entorno en el que operará, definir los objetivos de la tarea y diseñar algoritmos que permitan alcanzarlos de forma segura y eficiente. Como futuro profesional, deberás dominar tanto los aspectos conceptuales como los prácticos de esta disciplina.

Lenguajes utilizados en robótica

Python

Lenguaje de alto nivel, muy usado con ROS (Robot Operating System). Ideal para prototipado rápido, inteligencia artificial y visión por computador. Sintaxis clara y gran comunidad de soporte.

C / C++

Lenguajes de bajo nivel con alta eficiencia computacional. Imprescindibles en sistemas embebidos, microcontroladores y robots industriales donde el tiempo de respuesta es crítico.

Scratch / Blockly

Programación visual por bloques, ideal para iniciación. Permite construir algoritmos de forma gráfica sin escribir código textual. Compatible con robots educativos como LEGO, mBot o Makeblock.

ROS (Robot Operating System)

No es un SO convencional, sino un middleware robótico que proporciona herramientas, librerías y convenciones para simplificar la creación de software robótico complejo y distribuido.

Algoritmos de movimiento

Un **algoritmo de movimiento** es una secuencia finita y ordenada de instrucciones que especifica cómo debe moverse el robot para completar una tarea. Los algoritmos de movimiento pueden ser simples (avanza 50 cm, gira 90°, avanza 30 cm) o complejos (planificación de trayectorias con evitación de obstáculos en tiempo real). El diseño de buenos algoritmos de movimiento requiere conocer las capacidades cinemáticas del robot, las restricciones del entorno y los criterios de optimización (velocidad, consumo energético, seguridad).

Lógica de Programación Aplicada a Robots

La lógica de programación es la base sobre la que se construyen todos los programas robóticos, independientemente del lenguaje utilizado. Dominar los tres elementos fundamentales —condicionales, bucles y secuencias— te permitirá resolver prácticamente cualquier problema de programación robótica básica. A continuación se desarrolla cada uno con ejemplos aplicados directamente a robots.



Secuencias de Instrucciones

Una secuencia es el conjunto de instrucciones que el robot ejecuta en orden estrictamente lineal, una tras otra. Es el bloque más básico de cualquier programa. Ejemplo: *encender motor izquierdo → esperar 2 segundos → apagar motor izquierdo → encender motor derecho*. Las secuencias son perfectas para tareas predecibles y estructuradas donde no hay variabilidad en el entorno.



Condicionales (SI...ENTONCES)

Los condicionales permiten al robot tomar decisiones en función del estado de sus sensores o variables internas. La estructura básica es: *SI (condición) ENTONCES (acción A) SINO (acción B)*. Ejemplo: SI el sensor de distancia detecta un obstáculo a menos de 20 cm, ENTONCES detener los motores y girar 180°; SINO, continuar avanzando. Los condicionales dotan al robot de capacidad de adaptación al entorno.



Bucles (Repetición)

Los bucles permiten repetir un bloque de instrucciones un número determinado de veces o mientras se cumpla una condición. Son fundamentales en robótica porque muchos comportamientos son cíclicos: el robot seguidor de línea no ejecuta la corrección una sola vez, sino que la repite indefinidamente. Tipos principales: bucle *mientras* (WHILE), bucle *para* (FOR) y bucle infinito (LOOP), muy usado en robots autónomos.



Ejemplo práctico desarrollado — Robot Seguidor de Línea: Imagina un robot seguidor de línea con dos sensores infrarrojos (izquierdo y derecho) que detectan el contraste entre la línea negra y el suelo blanco. El programa utiliza: (1) un **bucle infinito** que mantiene el robot en funcionamiento continuo; (2) un **condicional** que comprueba si el sensor izquierdo detecta negro → si es así, el robot gira a la izquierda ajustando la velocidad del motor derecho; (3) otro **condicional** para el sensor derecho → si detecta negro, el robot gira a la derecha; (4) si ambos sensores detectan blanco, la **secuencia** indica avanzar recto. Este programa, aparentemente simple, contiene los tres elementos fundamentales de la lógica de programación.

Rol Profesional en la Programación Robótica

Como profesional de robótica, tu rol no se limitará a escribir código. Tu función abarcará un conjunto amplio de competencias técnicas y organizativas que son esenciales para garantizar que los sistemas robóticos funcionen de manera correcta, segura y eficiente en entornos reales de producción o investigación. Esta sección detalla las tres competencias nucleares que deberás desarrollar.

Interpretación de Algoritmos

Un profesional de robótica debe ser capaz de leer, comprender y analizar algoritmos escritos por otros, identificando la lógica de control, los parámetros configurables y los posibles puntos de fallo. Esta competencia es crítica cuando se trabaja con código heredado (legacy code) de sistemas en producción, o cuando se colabora en equipos multidisciplinares donde distintos ingenieros contribuyen al mismo proyecto. La interpretación algorítmica también implica verificar que el algoritmo cumple los requisitos de la tarea y detectar posibles ineficiencias o errores lógicos antes de la implementación en hardware real.

Configuración Básica de Programas

La configuración de programas robóticos implica adaptar los parámetros del software al hardware específico del robot y al entorno de operación. Esto incluye: calibrar los valores de los sensores (umbrales de detección, rangos de medición), ajustar los parámetros de los controladores (ganancias PID, velocidades máximas), configurar las interfaces de comunicación (protocolos I2C, UART, CAN) y establecer las condiciones de parada de emergencia. Un error en la configuración puede traducirse en comportamientos peligrosos o en daños al equipo, por lo que la precisión y el método sistemático son imprescindibles.

Supervisión de Funcionamiento Robótico

Una vez que el robot está en operación, el profesional debe supervisar su funcionamiento para garantizar que el sistema se comporta según las especificaciones. Esto implica monitorizar los datos de telemetría en tiempo real, interpretar los registros de eventos (logs), detectar anomalías en el comportamiento (movimientos bruscos, desviaciones de trayectoria, activaciones inesperadas de sensores) y aplicar procedimientos de diagnóstico y corrección. La supervisión también incluye el mantenimiento preventivo: actualizar el firmware del controlador, verificar la integridad del cableado y documentar el estado del sistema.

Caso Práctico 1: Programación Básica de Movimiento de Robot

En este caso práctico realizarás la programación básica del movimiento de un robot móvil de tracción diferencial (como el mBot o el robot de LEGO Mindstorms). El objetivo es que el robot realice un recorrido en forma de cuadrado, avanzando recto y girando 90° en cada esquina. Este ejercicio te permitirá aplicar secuencias de instrucciones y comprender la relación entre tiempo de activación de motores y distancia recorrida.

1

Paso 1: Definir el Movimiento Deseado

Antes de escribir una sola línea de código, debes analizar la tarea. El robot debe: avanzar 50 cm → girar 90° a la derecha → repetir 4 veces. Anota las características del robot: diámetro de las ruedas, distancia entre ruedas, velocidad máxima de los motores. Calcula el tiempo necesario para avanzar 50 cm a la velocidad seleccionada y el tiempo de giro para 90°. Esta fase de análisis previo es fundamental en la ingeniería robótica real.

2

Paso 2: Programar la Secuencia de Instrucciones

Con los parámetros calculados, redacta el pseudocódigo: *REPETIR 4 VECES: [activar ambos motores adelante durante T_{avance} segundos; detener motores; activar motor derecho atrás y motor izquierdo adelante durante T_{giro} segundos; detener motores]*. Implementa este pseudocódigo en el lenguaje del robot (Scratch, mBlock, EV3-G o Python). Verifica que los valores de tiempo están correctamente calculados antes de proceder.

3

Paso 3: Ejecutar el Programa

Carga el programa en el robot y ejecútalo en un espacio despejado. Observa si el recorrido es realmente un cuadrado o si hay desviaciones. Las desviaciones son normales y se deben a imprecisiones en el cálculo de tiempos o a diferencias entre los dos motores. Ajusta los parámetros de tiempo iterativamente hasta obtener un cuadrado preciso. Este proceso de ajuste iterativo es representativo del trabajo real de un programador robótico.



Consejo técnico: Si el robot no gira exactamente 90°, no cambies la velocidad de los motores, ajusta el tiempo de giro. Un incremento de 0,05 segundos puede suponer una diferencia de varios grados en el ángulo final. Utiliza un transportador para medir el ángulo real y compararlo con el objetivo de 90°.

Caso Práctico 2: Programación de Respuesta a Sensor

En este caso práctico programarás un robot para que reaccione de forma autónoma ante la señal de un sensor de distancia ultrasónico. El objetivo es implementar un comportamiento de evasión de obstáculos: el robot avanza libremente y, cuando detecta un obstáculo a menos de 25 cm, se detiene, gira 90° y continúa su marcha. Este caso te introduce en el paradigma del control en lazo cerrado, donde el sensor proporciona la realimentación que modifica el comportamiento del robot.

01	02	03
Paso 1: Detectar la Señal del Sensor	Paso 2: Programar la Respuesta	Paso 3: Ejecutar la Acción del Robot
El primer paso es verificar que el sensor ultrasónico funciona correctamente y que el controlador puede leer sus datos. Conecta el sensor al puerto digital correspondiente del controlador y escribe un programa de prueba simple que muestre en pantalla (o envíe por puerto serie) la distancia medida en tiempo real. Coloca objetos a diferentes distancias y verifica que las lecturas son coherentes. Es fundamental validar el sensor antes de integrarlo en el programa de control, ya que un sensor defectuoso o mal conectado puede generar comportamientos imprevisibles e incluso peligrosos en el robot.	Una vez validado el sensor, diseña el algoritmo de respuesta utilizando la estructura condicional: <i>MIENTRAS (verdadero): LEER distancia del sensor; SI (distancia menor que 25 cm) ENTONCES: detener motores, esperar 0,5 s, activar giro derecha durante T_giro s, reanudar marcha hacia adelante; SINO: continuar avanzando.</i> Implementa este algoritmo en el lenguaje de programación del robot. Presta especial atención al umbral de 25 cm: si es demasiado pequeño, el robot puede chocar antes de completar el giro; si es demasiado grande, el robot girará con demasiada antelación y su movimiento será errático.	Carga el programa y prueba el robot en un pasillo o sala con varios obstáculos distribuidos. Observa cómo responde ante diferentes tipos de obstáculos (paredes planas, objetos cilíndricos, esquinas) y evalúa si el comportamiento de evasión es efectivo. Documenta los casos en que el robot falla en detectar o evadir correctamente el obstáculo y analiza las causas: ¿es un problema de umbral? ¿El sensor tiene ángulo muerto? ¿El tiempo de giro es insuficiente? Esta fase de evaluación y diagnóstico es tan importante como la programación en sí.

Cuadro Comparativo: Control Manual vs. Control Automático

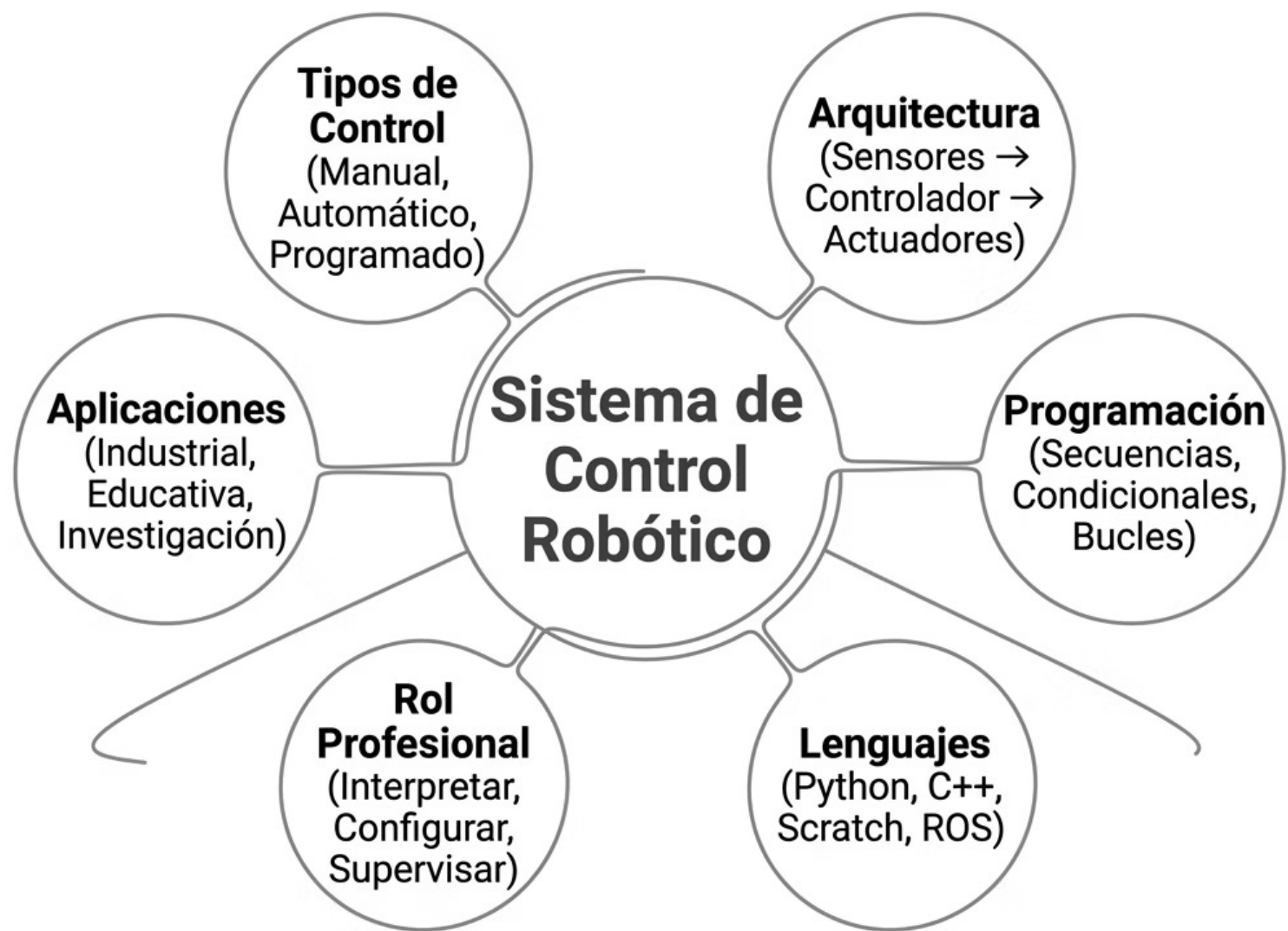
Una de las decisiones más relevantes en el diseño de un sistema robótico es determinar qué tipo de control utilizará. La siguiente comparativa te ayudará a comprender en profundidad las diferencias entre el control manual y el control automático, sus ventajas, limitaciones y contextos de aplicación más adecuados.

Criterio	Control Manual	Control Automático
Intervención humana	Directa y continua. El operador decide cada acción en tiempo real mediante mandos o interfaces.	Mínima o nula durante la operación. El sistema actúa según el programa establecido.
Velocidad de respuesta	Limitada por los reflejos y capacidades del operador humano (0,2-0,5 s de latencia típica).	Muy alta. Los sistemas electrónicos responden en milisegundos o microsegundos.
Precisión y repetibilidad	Variable. Depende de la fatiga, concentración y experiencia del operador.	Alta y constante. El robot repite la tarea con la misma precisión indefinidamente.
Flexibilidad ante imprevistos	Muy alta. El operador puede adaptarse a situaciones no previstas usando su criterio.	Limitada a los escenarios contemplados en el programa. Requiere reprogramación ante nuevas situaciones.
Coste operativo	Alto a largo plazo: requiere personal cualificado permanente.	Bajo a largo plazo: la inversión inicial se amortiza con la reducción de mano de obra.
Seguridad	El operador puede evaluar el riesgo en tiempo real y detener la operación instintivamente.	Depende de los sistemas de seguridad programados. Requiere diseño cuidadoso de paradas de emergencia.
Aplicaciones típicas	Cirugía robótica asistida, desactivación de explosivos, exploración espacial teledirigida.	Líneas de ensamblaje automotriz, pick-and-place, robots de soldadura, logística de almacenes.

Como puedes apreciar, ninguno de los dos tipos de control es superior en términos absolutos. La elección depende del contexto: cuando la tarea es repetitiva, predecible y de alto volumen, el control automático es claramente la mejor opción. Cuando el entorno es impredecible, la tarea es delicada o se requiere criterio humano, el control manual o la teleoperación son más apropiados. En la práctica, los sistemas robóticos modernos tienden hacia la **autonomía supervisada**, que combina la eficiencia del control automático con la capacidad de intervención humana cuando es necesario.

Resumen Visual de Conceptos Clave

A continuación encontrarás una síntesis visual de los conceptos fundamentales desarrollados en este tema. Utiliza este esquema como referencia rápida para consolidar tu aprendizaje y como punto de partida para la autoevaluación.



3

4

3

3

Tipos de Control
Manual, automático y programado: los tres paradigmas que rigen el comportamiento robótico en función del nivel de autonomía requerido.

Lenguajes Principales
Python, C/C++, Scratch y ROS son los entornos de programación más utilizados por los profesionales de robótica a nivel mundial.

Elementos Lógicos
Secuencias, condicionales y bucles: los tres bloques fundamentales con los que se construye cualquier programa robótico.

Competencias Profesionales
Interpretar algoritmos, configurar programas y supervisar el funcionamiento: las tres competencias nucleares del programador robótico.

Recomendaciones y Recursos de Ampliación

Lectura Recomendada

Para profundizar en los contenidos de este tema, se recomienda la lectura del artículo técnico sobre ROS (Robot Operating System) publicado por la Asociación Española de Robótica. Este documento te introducirá en el sistema operativo robótico más utilizado a nivel mundial, que es fundamental en el mercado laboral de la robótica profesional.

**Artículo recomendado:** [ROS — Robot Operating System. Artículo técnico de la AER \(Asociación Española de Robótica\)](#). Presta especial atención a los conceptos de nodos, tópicos y mensajes, que son la base de la comunicación en sistemas robóticos distribuidos.

Ejercicios de Verdadero o Falso sobre la Lectura

Una vez hayas leído el artículo, responde a los siguientes enunciados para verificar tu comprensión. Las respuestas correctas se proporcionan para que puedas autoevaluar tu aprendizaje:

1

Enunciado 1
«ROS es un sistema operativo convencional como Windows o Linux que gestiona el hardware del ordenador.»
→ **FALSO**. ROS no es un sistema operativo en el sentido convencional, sino un middleware o entorno de trabajo (framework) que se ejecuta sobre un sistema operativo estándar (generalmente Linux/Ubuntu) y proporciona herramientas, librerías y convenciones para el desarrollo de software robótico.

2

Enunciado 2
«En ROS, los nodos son procesos independientes que pueden comunicarse entre sí mediante el envío de mensajes a través de tópicos.»
→ **VERDADERO**. Los nodos son la unidad básica de ejecución en ROS. Cada nodo realiza una función específica (leer un sensor, controlar un motor, procesar imágenes) y se comunica con otros nodos publicando o suscribiéndose a tópicos que transportan mensajes con los datos necesarios.

3

Enunciado 3
«ROS solo puede utilizarse con robots industriales de alta gama y no es adecuado para entornos educativos o de investigación.»
→ **FALSO**. ROS está ampliamente adoptado en entornos académicos e investigadores gracias a su naturaleza de código abierto y su comunidad global. Existen paquetes específicos para robots educativos y plataformas de bajo coste, como TurtleBot, que hacen de ROS una herramienta accesible para el aprendizaje y la investigación.

Herramientas Digitales Recomendadas



Scratch — Programación por Bloques
Scratch es una herramienta de programación visual desarrollada por el MIT que permite crear programas mediante bloques gráficos que se encajan como piezas de puzle. Para la robótica, Scratch dispone de extensiones que conectan con robots educativos populares como LEGO WeDo, Micro:bit y mBot. Es el punto de entrada ideal para comprender la lógica de programación antes de migrar a lenguajes textuales. Accede en: [scratch.mit.edu](#)



Webots — Simulador Robótico
Webots es un simulador de robots de código abierto y uso profesional, utilizado tanto en universidades como en empresas de robótica. Permite diseñar y probar robots en entornos virtuales 3D realistas antes de implementarlos en hardware real, lo que reduce costes y riesgos. Compatible con ROS y múltiples lenguajes de programación. Accede en: [cyberbotics.com](#)



Podcast de Robótica
Complementa tu formación con contenido audiovisual de actualidad sobre robótica, inteligencia artificial y automatización. Los podcasts son un recurso excelente para mantenerte al día con las últimas tendencias del sector mientras realizas otras actividades. Accede en: [Spotify — Canal de Robótica](#)

Ejercicio de Autoevaluación

El siguiente ejercicio de desarrollo te permitirá verificar que has asimilado los conceptos fundamentales del tema. Redacta tu respuesta de forma autónoma antes de consultar la orientación proporcionada. Este tipo de ejercicio de síntesis es representativo de las preguntas que encontrarás en evaluaciones profesionales y académicas.

Pregunta de Desarrollo

«Explica cómo funcionan los sistemas de control en los robots, describiendo sus componentes principales, los tipos de control existentes y la relación entre el control y la programación robótica. Incluye al menos un ejemplo práctico que ilustre tu respuesta.»

Respuesta Sintetizada Orientativa para Autoevaluación

A continuación se desarrolla una respuesta completa que puedes usar como referencia para evaluar tu propia respuesta. No la consultes hasta haber redactado la tuya propia.

1. Definición y función del sistema de control

Un sistema de control robótico es el conjunto de componentes hardware y software que regulan el comportamiento de un robot para que ejecute tareas de forma precisa y segura. Su función principal es cerrar el bucle entre la percepción del entorno (sensores) y la acción sobre él (actuadores), tomando decisiones basadas en el estado actual del sistema y los objetivos programados. Sin un sistema de control adecuado, el robot no puede adaptarse a variaciones del entorno ni garantizar la precisión necesaria para tareas industriales o colaborativas.

2. Componentes principales

Todo sistema de control robótico se estructura en torno a tres componentes fundamentales: los **sensores**, que miden variables del entorno (distancia, temperatura, posición angular, fuerza); el **controlador central** (microcontrolador, PLC u ordenador), que procesa los datos y ejecuta los algoritmos de decisión; y los **actuadores** (motores, servos, cilindros), que convierten las órdenes del controlador en movimiento físico. La calidad y coordinación de estos tres elementos determina el rendimiento global del robot.

3. Tipos de control y su relación con la programación

Existen tres tipos principales: el **control manual**, donde el operador dirige el robot directamente; el **control automático**, donde el robot actúa de forma autónoma basándose en sensores y algoritmos de control en lazo cerrado; y el **control programado**, donde el comportamiento del robot queda definido por un programa creado previamente. La programación robótica es el puente entre el diseño del sistema de control y su implementación real: a través de algoritmos que combinan secuencias, condicionales y bucles, el programador define exactamente qué debe hacer el robot en cada situación.

4. Ejemplo práctico

Un robot seguidor de línea ilustra perfectamente el funcionamiento del sistema de control: los sensores infrarrojos (sensores) miden el contraste del suelo, el bloque programable EV3 (controlador) ejecuta el algoritmo de control que compara la lectura con el valor de referencia y calcula la corrección necesaria, y los motores (actuadores) ajustan su velocidad para mantener al robot sobre la línea. El bucle se repite continuamente a alta frecuencia, manteniendo al robot en la trayectoria correcta incluso ante curvas o cambios de iluminación ambiental.

☒ **Criterios de autoevaluación:** Tu respuesta será correcta si incluye: (1) una definición clara del sistema de control, (2) la mención de los tres componentes (sensor, controlador, actuador), (3) la diferencia entre al menos dos tipos de control, (4) la relación entre control y programación, y (5) un ejemplo concreto y bien desarrollado. Si tu respuesta cubre estos cinco puntos, has asimilado correctamente los contenidos del Tema 4.