

NOVA RP2350 – Guía para Niños: ¡Construye Robots Increíbles!

| ¡Bienvenido al Mundo de los Robots!

¿Te gustan los robots? ¿Quieres hacer que parpadeen unos ojos de robot, construir un dinosaurio lanzallamas o crear un robot gigante luchador? ¡Con NOVA RP2350 puedes hacerlo!

Esta es tu guía para aprender a crear robots súper cool con tecnología real.

| ¿Qué es NOVA RP2350?

NOVA es como el **cerebro de un robot**. Es una placa pequeña pero súper poderosa que puedes programar para hacer cosas increíbles:

- Encender luces de colores
- Hacer que servos muevan brazos de robot
- Leer sensores para que tu robot "vea" y "sienta"
- Controlar motores para que tu robot se mueva

¿Por qué es Especial?

NOVA tiene el **RP2350A**, el cerebro más nuevo de la familia Raspberry Pi Pico. Es como tener dos cerebros trabajando juntos (dual-core), ¡súper rápido!

Lo que puedes hacer:

- ⚡ 30 conexiones diferentes (pines) para sensores y luces
- 🕹️ Controlar hasta 24 servos al mismo tiempo

NOVA

- Hacer efectos de luz súper rápidos
 - Programar con Python (¡fácil de aprender!)
-

Proyecto 1: Ojos de Robot que Parpadean

¿Qué Necesitas?

- NOVA RP2350
- 2 LEDs redondos grandes (rojos, azules o verdes)
- 2 resistencias de 220Ω
- Cables jumper
- Cable USB

Diagrama de Conexión

| | | |
|--------|---|--|
| NOVA | → | LED Ojo Izquierdo |
| Pin 15 | → | Pata larga (+) del LED → Resistencia 220Ω → GND |
| Pin 14 | → | Pata larga (+) del LED → Resistencia 220Ω → GND |

NOVA

El Código Mágico (MicroPython)

NOVA

```

from machine import Pin
import time
import random

# Configurar los "ojos" del robot
ojos_izquierdos = Pin(15, Pin.OUT)
ojos_derechos = Pin(14, Pin.OUT)

def parpadeo_normal():
    """Parpadeo simple como una persona."""
    # Apagar ojos
    ojos_izquierdos.off()
    ojos_derechos.off()
    time.sleep(0.15) # Cerrado 150ms

    # Encender ojos
    ojos_izquierdos.on()
    ojos_derechos.on()

def parpadeo_robot():
    """Parpadeo como robot (alternado)."""
    # Parpadear ojo izquierdo
    ojos_izquierdos.off()
    time.sleep(0.1)
    ojos_izquierdos.on()
    time.sleep(0.1)

    # Parpadear ojo derecho
    ojos_derechos.off()
    time.sleep(0.1)
    ojos_derechos.on()

def escaneo():
    """Efecto de escaneo de robot."""
    for _ in range(3):
        ojos_izquierdos.on()
        ojos_derechos.off()
        time.sleep(0.1)

        ojos_izquierdos.off()
        ojos_derechos.on()
        time.sleep(0.1)

    # ¡El programa principal!
    print("¡Robot activado! 🤖")

while True:
    # Encender ojos
    ojos_izquierdos.on()
    ojos_derechos.on()

```

NOVA

```
time.sleep(random.randint(2, 5)) # Esperar 2-5 segundos

# Elegir efecto random
efecto = random.randint(1, 3)

if efecto == 1:
    parpadeo_normal()
elif efecto == 2:
    parpadeo_robot()
else:
    escaneo()
```

¡Mejóralo!**Ideas Cool:**

1. Agrega más LEDs alrededor para hacer "cejas" de robot
2. Usa LEDs RGB para cambiar el color de los ojos según el modo
3. Conecta un sensor de distancia para que parpadee cuando alguien se acerque

Proyecto 2: Robot con Brazos Móviles **¿Qué Necesitas?**

- NOVA RP2350
- 2 servos SG90 (para los brazos)
- Fuente de alimentación 5V (o batería)
- Palitos de helado o cartón (para hacer los brazos)
- Cinta adhesiva y pegamento

Cómo Conectar los Servos**Servo Brazo Izquierdo:**

- Cable Naranja (señal) → Pin 16 de NOVA
 Cable Rojo (5V) → 5V (fuente externa)
 Cable Café (GND) → GND (común con NOVA)

Servo Brazo Derecho:

- Cable Naranja (señal) → Pin 17 de NOVA
 Cable Rojo (5V) → 5V (fuente externa)
 Cable Café (GND) → GND (común con NOVA)

NOVA

 **IMPORTANTE:** Los servos necesitan su propia fuente de 5V porque consumen mucha energía. No los conectes directo a NOVA o podría apagarse.

NOVA

El Código para Controlar los Brazos

NOVA

```

from machine import Pin, PWM
import time

class ServoMotor:
    """Clase para controlar un servo fácilmente."""

    def __init__(self, pin_numero):
        self.servo = PWM(Pin(pin_numero))
        self.servo.freq(50) # 50Hz para servos

    def mover(self, angulo):
        """Mueve el servo a un ángulo (0-180 grados)."""
        # Convertir ángulo a duty cycle
        # 0° = 1ms (duty ~1600), 180° = 2ms (duty ~8000)
        duty = int(1600 + (angulo / 180) * 6400)
        self.servo.duty_u16(duty)

    def apagar(self):
        """Apaga el servo para ahorrar energía."""
        self.servo.duty_u16(0)

# Crear los brazos del robot
brazo_izquierdo = ServoMotor(16)
brazo_derecho = ServoMotor(17)

def saludar():
    """El robot saluda moviendo un brazo."""
    print("¡Hola! 🙋")
    for _ in range(3):
        brazo_derecho.mover(90) # Subir brazo
        time.sleep(0.3)
        brazo_derecho.mover(180) # Bajar brazo
        time.sleep(0.3)

def pose_fuerza():
    """Pose de robot fuerte con ambos brazos arriba."""
    print("¡SOY FUERTE! 💪")
    brazo_izquierdo.mover(45)
    brazo_derecho.mover(135)
    time.sleep(2)

def bailar():
    """El robot baila moviendo los brazos."""
    print("¡A bailar! 💃")
    for _ in range(4):
        brazo_izquierdo.mover(0)
        brazo_derecho.mover(180)
        time.sleep(0.5)

        brazo_izquierdo.mover(180)

```

NOVA

```

brazo_derecho.mover(0)
time.sleep(0.5)

def posicion_reposo():
    """Brazos en posición neutral."""
    brazo_izquierdo.mover(90)
    brazo_derecho.mover(90)

# ¡Programa principal!
print("🤖 Robot listo para acción!")

posicion_reposo()
time.sleep(1)

while True:
    saludar()
    time.sleep(2)

    pose_fuerza()
    time.sleep(2)

    bailar()
    time.sleep(2)

    posicion_reposo()
    time.sleep(3)

```

Proyecto 3: Robot Luchador con Sensores

¿Qué Necesitas?

- NOVA RP2350
- 2 sensores ultrasónicos HC-SR04 (para "ver")
- 2 servos grandes (para brazos/pinzas)
- Buzzer (para sonidos de batalla)
- LEDs rojos (ojos de guerra)
- Estructura de cartón o impresa en 3D

NOVA

Sistema de Detección de Enemigo

NOVA

```

from machine import Pin, PWM
import time

class SensorUltrasonico:
    """Para que el robot vea obstáculos."""

    def __init__(self, trigger_pin, echo_pin):
        self.trigger = Pin(trigger_pin, Pin.OUT)
        self.echo = Pin(echo_pin, Pin.IN)

    def medir_distancia(self):
        """Mide distancia en centímetros."""
        # Enviar pulso
        self.trigger.low()
        time.sleep_us(2)
        self.trigger.high()
        time.sleep_us(10)
        self.trigger.low()

        # Medir eco
        while self.echo.value() == 0:
            inicio = time.ticks_us()

        while self.echo.value() == 1:
            fin = time.ticks_us()

        # Calcular distancia
        duracion = time.ticks_diff(fin, inicio)
        distancia = duracion * 0.034 / 2

        return distancia

class RobotLuchador:
    """¡El robot luchador completo!"""

    def __init__(self):
        # Sensores
        self.sensor_izq = SensorUltrasonico(trigger_pin=18, echo_pin=19)
        self.sensor_der = SensorUltrasonico(trigger_pin=20, echo_pin=21)

        # Brazos/pinzas
        self.brazo_izq = PWM(Pin(16))
        self.brazo_der = PWM(Pin(17))
        self.brazo_izq.freq(50)
        self.brazo_der.freq(50)

        # Ojos de guerra (LEDs rojos)
        self.ojos = Pin(15, Pin.OUT)

        # Buzzer para sonidos

```

NOVA

```

        self.buzzer = PWM(Pin(14))

        self.en_combate = False

    def sonido_alarma(self):
        """¡PELIGRO!"""
        for _ in range(5):
            self.buzzer.freq(800)
            self.buzzer.duty_u16(30000)
            time.sleep(0.1)
            self.buzzer.duty_u16(0)
            time.sleep(0.1)

    def sonido_ataque(self):
        """Sonido de golpe."""
        self.buzzer.freq(200)
        self.buzzer.duty_u16(40000)
        time.sleep(0.15)
        self.buzzer.duty_u16(0)

    def modo_patrulla(self):
        """Vigilando..."""
        self.ojos.off()
        time.sleep(0.5)
        self.ojos.on()
        time.sleep(0.5)

    def detectar_enemigo(self):
        """Revisa si hay algo cerca."""
        dist_izq = self.sensor_izq.medir_distancia()
        dist_der = self.sensor_der.medir_distancia()

        # Si algo está a menos de 30cm
        if dist_izq < 30 or dist_der < 30:
            return True, min(dist_izq, dist_der)
        return False, 0

    def atacar(self):
        """¡ATAQUE!"""
        print("🔴 ¡ATACANDO!")
        self.ojos.on() # Ojos rojos encendidos
        self.sonido_alarma()

        # Mover brazos en ataque
        for _ in range(3):
            # Brazo derecho golpea
            self.brazo_der.duty_u16(8000) # 180°
            time.sleep(0.2)
            self.sonido_ataque()
            self.brazo_der.duty_u16(1600) # 0°
            time.sleep(0.2)

```

NOVA

```

# Brazo izquierdo golpea
self.brazo_izq.duty_u16(1600)  # 0°
time.sleep(0.2)
self.sonido_ataque()
self.brazo_izq.duty_u16(8000)  # 180°
time.sleep(0.2)

def iniciar(self):
    """Loop principal del robot."""
    print("🤖 Robot Luchador ACTIVADO")

    while True:
        enemigo_detectado, distancia = self.detectar_enemigo()

        if enemigo_detectado:
            print(f"⚠️ ENEMIGO A {distancia:.1f}cm!")
            self.atacar()
        else:
            self.modo_patrulla()

# ¡Crear y activar el robot!
robot = RobotLuchador()
robot.iniciar()

```

Ideas para Mejorar tu Robot Luchador

- Escudo:** Agrega un servo que levante un escudo cuando detecte enemigo
- Más sensores:** Pon sensores atrás para detectar ataques sorpresa
- Control remoto:** Usa Bluetooth para controlarlo desde tu teléfono
- Contador de victorias:** Usa un display para mostrar cuántas batallas ganó

Proyecto 4: Dinosaurio Robot con Efectos

¡IMPORTANTE DE SEGURIDAD! 🔥

NO hagas un dinosaurio con lanzallamas real. Es MUY peligroso y puede causar incendios. En lugar de eso, ¡vamos a hacer efectos increíbles con LEDs y humo artificial (máquina de niebla)!

NOVA

El Dinosaurio T-Rex Robot

¿Qué Necesitas?

- NOVA RP2350
- Tira de LEDs NeoPixel (WS2812B) rojos y naranjas
- Servo para la mandíbula
- Servos para las patas
- Sensor de movimiento PIR
- Mini máquina de niebla (5V, segura)
- Buzzer para rugidos

NOVA

El Código del T-Rex

NOVA

```

from machine import Pin, PWM
import neopixel
import time
import random

class DinosaurioRobot:
    """¡Un T-Rex robot increíble!"""

    def __init__(self):
        # LEDs para "llamas" en la boca
        self.num_leds = 12
        self.leds = neopixel.NeoPixel(Pin(16), self.num_leds)

        # Servo para mandíbula
        self.mandibula = PWM(Pin(17))
        self.mandibula.freq(50)

        # Servos para patas (caminar)
        self.pata_izq = PWM(Pin(18))
        self.pata_der = PWM(Pin(19))
        self.pata_izq.freq(50)
        self.pata_der.freq(50)

        # Sensor de movimiento
        self.sensor_pir = Pin(20, Pin.IN)

        # Buzzer para rugidos
        self.buzzer = PWM(Pin(14))

        # Control de máquina de niebla
        self.niebla = Pin(15, Pin.OUT)

    def efecto_fuego(self):
        """Efecto de fuego con LEDs (¡parece real!)"""
        for i in range(self.num_leds):
            # Colores de fuego: rojo a naranja con variación
            rojo = random.randint(200, 255)
            verde = random.randint(0, 100)
            azul = 0
            self.leds[i] = (rojo, verde, azul)
        self.leds.write()

    def rugido(self):
        """¡ROAAAAR!"""
        print("🦖 ROAAAAAAAR!")

        # Sonido de rugido (frecuencias bajas)
        for freq in [80, 60, 40, 60, 80, 60]:
            self.buzzer.freq(freq)
            self.buzzer.duty_u16(40000)

```

NOVA

```

        time.sleep(0.15)
        self.buzzer.duty_u16(0)

def abrir_mandibula(self):
    """Abre la boca del dinosaurio."""
    self.mandibula.duty_u16(8000) # 180°

def cerrar_mandibula(self):
    """Cierra la boca."""
    self.mandibula.duty_u16(1600) # 0°

def escupir_fuego(self):
    """¡Efecto de escupir fuego!"""
    print("🔥 ¡ESCUPIENDO FUEGO!")

# Abrir boca
self.abrir_mandibula()

# Activar niebla
self.niebla.on()

# Efecto de fuego con LEDs (animación rápida)
for _ in range(20):
    self.efecto_fuego()
    time.sleep(0.05)

# Cerrar boca y apagar niebla
self.cerrar_mandibula()
self.niebla.off()

# Apagar LEDs gradualmente
for brillo in range(10, 0, -1):
    for i in range(self.num_leds):
        self.leds[i] = (brillo * 20, brillo * 8, 0)
    self.leds.write()
    time.sleep(0.1)

def caminar(self):
    """Movimiento de caminar (patas alternadas)."""
    for _ in range(4):
        # Pata izquierda adelante
        self.pata_izq.duty_u16(6000)
        self.pata_der.duty_u16(3000)
        time.sleep(0.3)

        # Pata derecha adelante
        self.pata_izq.duty_u16(3000)
        self.pata_der.duty_u16(6000)
        time.sleep(0.3)

def modo_cazador(self):

```

NOVA

```
"""El dinosaurio caza cuando detecta movimiento."""
print("👀 Modo cazador activado...")

while True:
    if self.sensor_pir.value() == 1:
        # ¡Detectó algo!
        print("⚠️ PRESA DETECTADA!")

        # Rugir
        self.rugido()
        time.sleep(0.5)

        # Caminar hacia la presa
        self.caminar()

        # ¡Atacar con fuego!
        self.escupir_fuego()

        # Esperar antes de volver a patrullar
        time.sleep(5)
    else:
        # Patrullar: parpadeando ojos (LEDs suaves)
        for i in range(2):
            self.leds[i] = (10, 0, 0)
        self.leds.write()
        time.sleep(0.5)

        for i in range(self.num_leds):
            self.leds[i] = (0, 0, 0)
        self.leds.write()
        time.sleep(0.5)

# ¡Crear el dinosaurio!
dino = DinosaurioRobot()

print("🦖 T-Rex Robot Iniciado")
print("Esperando detectar movimiento...")

# ¡Activar modo cazador!
dino.modo_cazador()
```

Mejoras Épicas para tu Dinosaurio

1. **Cola móvil:** Agrega un servo en la cola para balance
2. **Sonido mejorado:** Usa un módulo MP3 para rugidos reales
3. **Piel texturizada:** Usa espuma modelada pintada
4. **Control de altura:** Haz que se agache antes de atacar

NOVA

Proyecto 5: Robot Gigante Luchador

(¡Estilo Mecha!)

El Concepto

Un robot grande (30-50cm) con:

- Brazos articulados (múltiples servos)
- Luces LED en el pecho
- Sistema de sonido
- Sensores de toque para detectar golpes

Lista Completa de Materiales

Electrónica:

- 1× NOVA RP2350
- 6× Servos grandes (MG996R o similar)
- 1× Tira NeoPixel (20 LEDs)
- 1× Buzzer
- 4× Sensores de toque (botones)
- 1× Fuente 5V 3A
- Cables y conectores

Estructura:

- Cartón grueso o madera MDF
- Tornillos y tuercas
- Pegamento caliente
- Pintura (colores metálicos)

NOVA

Sistema de Movimiento

NOVA

```

from machine import Pin, PWM
import time

class MechaGigante:
    """Robot gigante de batalla."""

    def __init__(self):
        # Servos del cuerpo
        self.servos = {
            'hombro_izq': PWM(Pin(16)),
            'codo_izq': PWM(Pin(17)),
            'hombro_der': PWM(Pin(18)),
            'codo_der': PWM(Pin(19)),
            'cintura': PWM(Pin(20)),
            'cabeza': PWM(Pin(21))
        }

        # Configurar frecuencia
        for servo in self.servos.values():
            servo.freq(50)

        # LEDs del pecho (reactor arc)
        import neopixel
        self.reactor = neopixel.NeoPixel(Pin(22), 20)

        # Sensores de impacto
        self.sensor_pecho = Pin(10, Pin.IN, Pin.PULL_UP)
        self.sensor_brazo_izq = Pin(11, Pin.IN, Pin.PULL_UP)
        self.sensor_brazo_der = Pin(12, Pin.IN, Pin.PULL_UP)
        self.sensor_cabeza = Pin(13, Pin.IN, Pin.PULL_UP)

        # Sistema de vida
        self.vida = 100

        # Buzzer
        self.sonido = PWM(Pin(14))

    def mover_servo(self, nombre, angulo):
        """Mueve un servo específico."""
        duty = int(1600 + (angulo / 180) * 6400)
        self.servos[nombre].duty_u16(duty)

    def efecto_reactor(self, intensidad):
        """Efecto de reactor en el pecho."""
        # Azul brillante que pulsa
        for i in range(20):
            brillo = int(intensidad * 2.55)
            self.reactor[i] = (0, brillo // 3, brillo)
        self.reactor.write()

```

NOVA

```

def pose_neutral(self):
    """Posición de reposo."""
    self.mover_servo('hombro_izq', 90)
    self.mover_servo('codo_izq', 90)
    self.mover_servo('hombro_der', 90)
    self.mover_servo('codo_der', 90)
    self.mover_servo('cintura', 90)
    self.mover_servo('cabeza', 90)

def puñetazo_derecho(self):
    """;Golpe con brazo derecho!"""
    print("👉 ;PUÑETAZO DERECHO!")

    # Preparar golpe
    self.mover_servo('hombro_der', 45)
    self.mover_servo('codo_der', 45)
    time.sleep(0.3)

    # ¡GOLPE!
    self.sonido.freq(300)
    self.sonido.duty_u16(40000)

    self.mover_servo('hombro_der', 180)
    self.mover_servo('codo_der', 180)
    time.sleep(0.2)

    self.sonido.duty_u16(0)

    # Regresar
    time.sleep(0.3)
    self.pose_neutral()

def puñetazo_izquierdo(self):
    """;Golpe con brazo izquierdo!"""
    print("👉 ;PUÑETAZO IZQUIERDO!")

    self.mover_servo('hombro_izq', 135)
    self.mover_servo('codo_izq', 135)
    time.sleep(0.3)

    self.sonido.freq(300)
    self.sonido.duty_u16(40000)

    self.mover_servo('hombro_izq', 0)
    self.mover_servo('codo_izq', 0)
    time.sleep(0.2)

    self.sonido.duty_u16(0)

    time.sleep(0.3)
    self.pose_neutral()

```

NOVA

```

def combo_mortal(self):
    """¡Combo devastador!"""
    print("💥 ¡COMBO MORTAL!")

    # Golpe 1-2
    self.puñetazo_derecho()
    time.sleep(0.1)
    self.puñetazo_izquierdo()
    time.sleep(0.1)

    # Giro de cintura + gancho
    self.mover_servo('cintura', 45)
    time.sleep(0.2)
    self.puñetazo_derecho()

    # Uppercut
    self.mover_servo('cintura', 135)
    self.mover_servo('hombro_izq', 180)
    self.mover_servo('codo_izq', 45)

    # Sonido de impacto final
    for _ in range(3):
        self.sonido.freq(200)
        self.sonido.duty_u16(50000)
        time.sleep(0.1)
        self.sonido.duty_u16(0)
        time.sleep(0.05)

    self.pose_neutral()

def recibir_daño(self, zona):
    """Reacción al recibir golpe."""
    self.vida -= 10
    print(f"💔 ¡Golpe en {zona}! Vida: {self.vida}%")

    # Efecto visual (reactor parpadea rojo)
    for _ in range(3):
        for i in range(20):
            self.reactor[i] = (255, 0, 0)
        self.reactor.write()
        time.sleep(0.1)

        for i in range(20):
            self.reactor[i] = (0, 0, 0)
        self.reactor.write()
        time.sleep(0.1)

    # Sonido de dolor
    self.sonido.freq(100)
    self.sonido.duty_u16(30000)

```

NOVA

```

time.sleep(0.3)
self.sonido.duty_u16(0)

# Reaccionar según zona
if zona == "cabeza":
    # Mover cabeza
    self.mover_servo('cabeza', 45)
    time.sleep(0.5)
    self.mover_servo('cabeza', 90)

if self.vida <= 0:
    self.derrota()

def derrota(self):
    """Animación de derrota."""
    print("😢 ;DERROTADO!")

    # Caer al suelo (todos los servos abajo)
    for nombre in ['hombro_izq', 'hombro_der', 'cintura']:
        self.mover_servo(nombre, 0)

    # Apagar reactor
    for i in range(20):
        self.reactor[i] = (0, 0, 0)
    self.reactor.write()

    # Sonido de derrota
    for freq in [300, 250, 200, 150, 100, 50]:
        self.sonido.freq(freq)
        self.sonido.duty_u16(20000)
        time.sleep(0.2)
    self.sonido.duty_u16(0)

def victoria(self):
    """;Pose de victoria!"""
    print("🏆 ¡VICTORIA!")

    # Brazos arriba
    self.mover_servo('hombro_izq', 0)
    self.mover_servo('hombro_der', 180)
    self.mover_servo('codo_izq', 0)
    self.mover_servo('codo_der', 180)

    # Reactor brillante
    for ciclo in range(5):
        for brillo in range(0, 100, 10):
            self.efecto_reactor(brillo)
            time.sleep(0.05)

    # Música de victoria
    melodía = [523, 587, 659, 784, 880, 784, 659, 523]

```

NOVA

```

for nota in melodía:
    self.sonido.freq(nota)
    self.sonido.duty_u16(30000)
    time.sleep(0.2)
    self.sonido.duty_u16(0)

def modo_batalla(self):
    """Loop principal de batalla."""
    self.pose_neutral()
    self.efecto_reactor(50)

    print("⚔️ MECHA GIGANTE LISTO PARA BATALLA")

    while self.vida > 0:
        # Revisar sensores de impacto
        if self.sensor_cabeza.value() == 0:
            self.recibir_daño("cabeza")
        elif self.sensor_pecho.value() == 0:
            self.recibir_daño("pecho")
        elif self.sensor_brazo_izq.value() == 0:
            self.recibir_daño("brazo izquierdo")
        elif self.sensor_brazo_der.value() == 0:
            self.recibir_daño("brazo derecho")

        # Rutina de ataque automática
        import random
        acción = random.randint(1, 10)

        if acción <= 3:
            self.puñetazo_derecho()
        elif acción <= 6:
            self.puñetazo_izquierdo()
        elif acción <= 8:
            self.combo_mortal()

        # Actualizar reactor
        self.efecto_reactor(self.vida)

        time.sleep(0.5)

# ¡Crear el mecha!
mecha = MechaGigante()
mecha.modo_batalla()

```

Consejos de Construcción



Para Hacer tu Robot Fuerte

1. **Usa cartón grueso** (de cajas grandes) o madera MDF
2. **Refuerza las juntas** con pegamento caliente + cinta adhesiva
3. **Monta los servos firmes** con tornillos, no solo pegamento
4. **Balancea el peso** para que no se caiga
5. **Protege los cables** con cinta o tubos

Consejos de Diseño

1. **Dibuja primero** tu robot en papel
 2. **Haz partes separadas** (cabeza, cuerpo, brazos, piernas)
 3. **Deja espacio** para la electrónica dentro
 4. **Decora al final** con pintura y stickers
-

¿Qué Más Puedes Hacer?

Ideas de Proyectos

1. **Robot Perro** que camina en 4 patas y ladra
2. **Mano Robótica** que atrapa objetos
3. **Robot Araña** con 8 patas
4. **Transformador** que cambia de forma (carro a robot)
5. **Exoesqueleto** para tu brazo (¡como Iron Man!)

Aprende Más

- **YouTube:** Busca "raspberry pi pico robot" para tutoriales
 - **Libros:** "Python para niños" (aprende a programar)
 - **Clubes:** Únete a un club de robótica en tu escuela
-

NOVA

Palabras Finales

¡Felicitaciones! Ahora sabes cómo crear robots increíbles con NOVA RP2350. Recuerda:

💡 **Empieza simple** - Primero haz parpadear un LED, luego proyectos más grandes

🧪 **Experimenta** - Cambia el código, prueba cosas nuevas

🔧 **Repara** - Si algo no funciona, revisa las conexiones

🎨 **Sé creativo** - ¡Tu robot puede ser único!

🤝 **Comparte** - Muestra tus creaciones a amigos y familia

¡EL FUTURO ES TUYO! Construye, programa, ¡y conquista el mundo con robots! 

¿Tienes dudas? Pregunta a tus papás, maestros o busca en internet. ¡La comunidad maker está lista para ayudarte!