

Introducción y uso práctico de Raspberry Pi

Conceptos, modelos, usos y programación

Abel Carnicero
acarnm@unileon.es

7 de noviembre de 2025

- 1 ¿Qué es Raspberry Pi?
- 2 Modelos y diferencias
- 3 Qué se puede hacer con Raspberry Pi
- 4 Instalación y configuración
- 5 Uso básico y software
- 6 Programación y ejemplos

Definición I

¿Qué es?

Un **ordenador completo del tamaño de una tarjeta de crédito** (SBC: Single Board Computer), que puede ser conectada a un monitor o una TV, y usarse con un mouse y teclado estándar.

Características clave:

- ✓ ordenador completo funcional
- ✓ Precio: desde 5€ hasta 100€
- ✓ Bajo consumo energético (3-5W)
- ✓ Puertos GPIO programables
- ✓ Sistema operativo Linux

Origen y propósito:

- Creada en **2012** por la *Raspberry Pi Foundation* (UK)
- Objetivo: **democratizar la informática**
- Más de **50 millones** de unidades vendidas
- Usada en educación, industria y hobbies

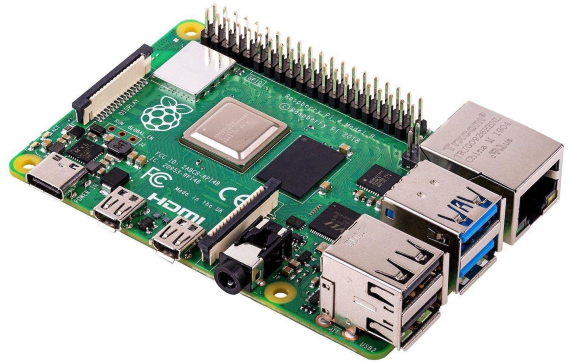
¿Por qué es revolucionaria?

Convierte la programación y electrónica en algo **accesible, económico y práctico** para cualquier persona.

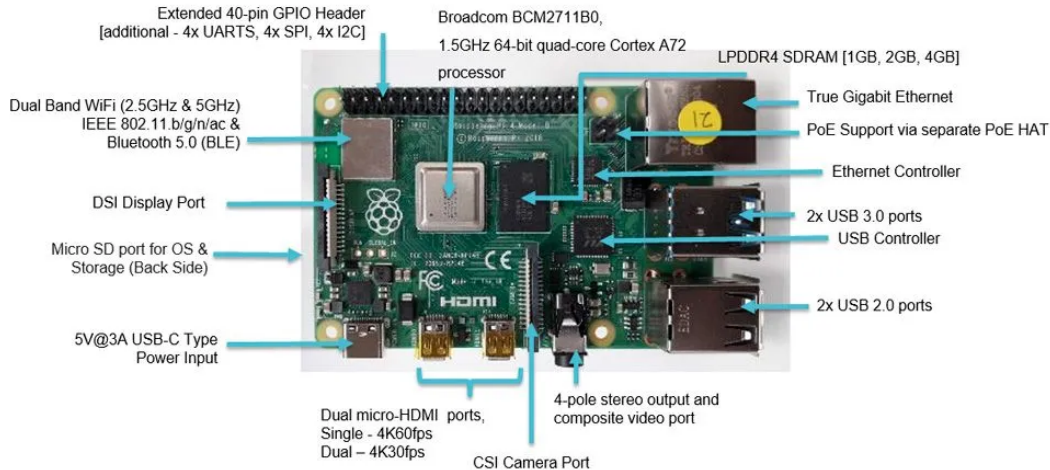
Ha democratizado el acceso a la tecnología permitiendo que estudiantes, aficionados y profesionales puedan crear desde simples proyectos de aprendizaje hasta **sistemas industriales complejos**. Su versatilidad la hace ideal tanto para **enseñar conceptos básicos de programación** a niños como para desarrollar **prototipos IoT avanzados** en empresas.

Partes principales I

- CPU y GPU (procesador ARM)
- RAM
- Puertos USB y HDMI
- Conector GPIO
- Conector de cámara (CSI)
- Lector microSD



Partes principales II



Componentes explicados:

- **CPU/GPU (SoC):** Procesador ARM Broadcom que ejecuta todas las operaciones. Incluye la GPU para gráficos y video.
- **RAM:** Memoria de acceso aleatorio (1-8GB según modelo) para ejecutar programas y sistema operativo.
- **GPIO (40 pines):** Pines de entrada/salida programables para conectar sensores, LEDs, motores y otros dispositivos electrónicos.
- **Puertos USB:** Para conectar teclado, ratón, memorias USB y otros periféricos. USB 3.0 en Pi 4/5.

Partes principales IV

- **HDMI:** Salida de video y audio digital para conectar monitores o TVs (hasta 4K en Pi 4/5).
- **Puerto Ethernet (RJ45):** Conexión de red cableada (Gigabit en Pi 4/5).
- **Conector microSD:** Almacenamiento principal donde se instala el sistema operativo y archivos.
- **Puerto CSI/DSI:** Para conectar la cámara oficial (CSI) o pantallas táctiles (DSI).
- **USB-C/microUSB:** Alimentación eléctrica (5V, 3A recomendado para Pi 4/5).

Seguridad física

- Manipularla en superficies **no conductoras** (evitar metal, aluminio).
- **No tocar** los pines GPIO mientras está encendida.
- Evitar electricidad estática - descargarse antes de manipular.
- Considerar usar una **carcasa protectora** para evitar cortocircuitos.

Alimentación eléctrica

- Usar una fuente de alimentación **oficial o de calidad** (5V, 3A para Pi 4/5).
- Alimentación insuficiente causa **reinicios aleatorios** y corrupción de datos.
- Síntoma común: icono de rayo amarillo en pantalla = **voltaje bajo**.
- No conectar/desconectar periféricos con la Pi encendida.

Precauciones y buen uso II

Gestión del sistema

- **NUNCA** desconectar la energía sin apagar el sistema operativo.
- Usar siempre: `sudo shutdown -h now` o menú de apagado.
- Apagado incorrecto puede **corromper la tarjeta SD**.
- Hacer **copias de seguridad** regulares de proyectos importantes.

Temperatura y ventilación

- Temperatura óptima: **< 70°C** (revisar con `vcgencmd measure_temp`).
- Usar **disipadores de calor** o ventiladores activos en proyectos intensivos.
- No cubrir la placa durante operación - necesita **ventilación**.
- El throttling térmico reduce rendimiento automáticamente.

Uso de GPIO - Precauciones críticas

- Los pines GPIO son **3.3V** - NO conectar componentes de 5V directamente.
- Corriente máxima por pin: **16mA** - usar transistores para cargas mayores.
- Algunos pines tienen funciones especiales (I2C, SPI, UART) - documentar antes de usar.
- **Conexión incorrecta puede dañar permanentemente** la Raspberry Pi.

Buenas prácticas recomendadas

- Actualizar regularmente: `sudo apt update && sudo apt upgrade`
- Usar tarjetas SD de **calidad** (Clase 10, A1 o superior).
- Documentar conexiones y proyectos (esquemas, fotos, código comentado).
- Unirse a la **comunidad** - foros, Reddit ([r/raspberry_pi](https://www.reddit.com/r/raspberry_pi)).

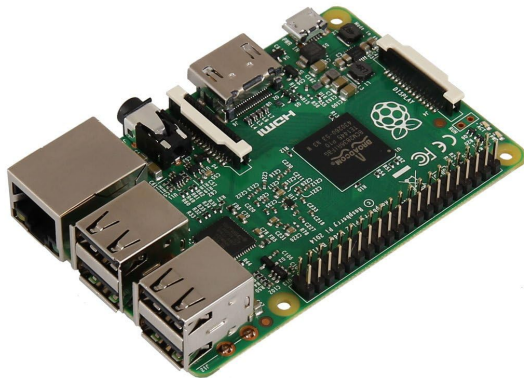
Raspberry Pi 1 (A, B)

- Primer modelo (2012)
- CPU: 700 MHz ARM11
- RAM: 256-512 MB
- Revolucionó la educación



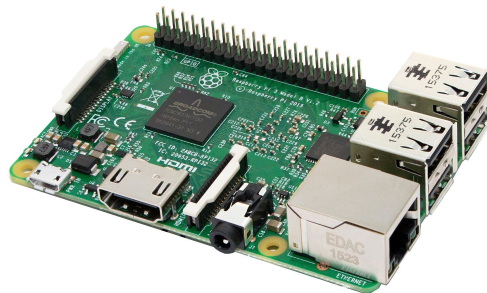
Raspberry Pi 2

- Lanzamiento: 2015
- CPU: 900 MHz quad-core ARM Cortex-A7
- RAM: 1 GB
- 6× más rápida que Pi 1



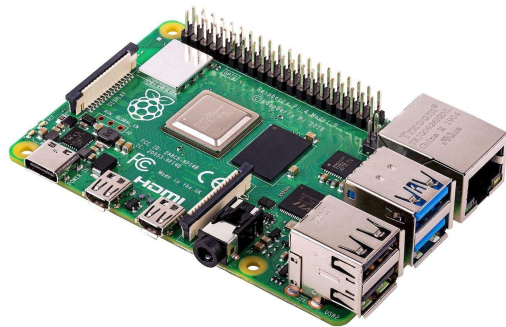
Raspberry Pi 3 (B, B+)

- Lanzamiento: 2016-2018
- CPU: 1.2-1.4 GHz quad-core ARM Cortex-A53
- RAM: 1 GB
- **Wi-Fi y Bluetooth integrados**
- Ideal para proyectos IoT



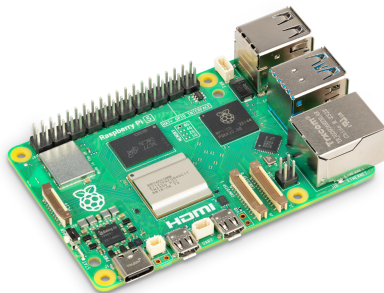
Raspberry Pi 4

- Lanzamiento: 2019
- CPU: 1.5 GHz quad-core ARM Cortex-A72
- RAM: 2/4/8 GB LPDDR4
- USB 3.0, dual micro-HDMI 4K
- Gigabit Ethernet real



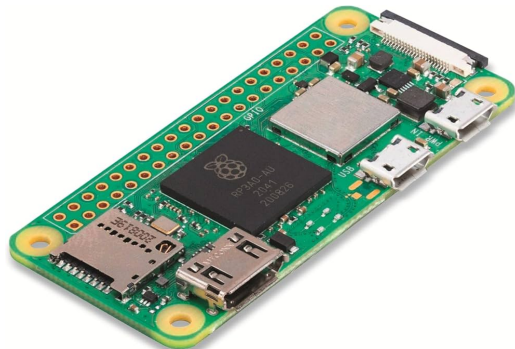
Raspberry Pi 5

- Lanzamiento: 2023
- CPU: 2.4 GHz quad-core ARM Cortex-A76
- RAM: 4/8 GB LPDDR4X
- GPU VideoCore VII mejorada
- PCIe 2.0, botón de encendido
- 2-3× más rápida que Pi 4



Raspberry Pi Zero / Zero 2W

- Tamaño ultra compacto (65×30 mm)
- Zero: 1 GHz single-core (desde \$5)
- Zero 2W: 1 GHz quad-core + Wi-Fi/BT
- Ideal para proyectos embebidos
- Menor consumo energético



Comparativa rápida I

Característica	Pi 3B+	Pi 4B	Pi 5	Zero 2W
CPU	1.4 GHz 4-core	1.5 GHz 4-core	2.4 GHz 4-core	1.0 GHz 4-core
RAM	1 GB	2-8 GB	4-8 GB	512 MB
Wi-Fi/BT	✓ 4.2	✓ 5.0	✓ 5.3	✓ 4.2
USB	4× 2.0	2× 3.0 2× 2.0	2× 3.0 2× 2.0	1× 2.0 (OTG)
Video	1× HDMI 1080p	2× micro 4K@60	2× micro 4K@60	1× mini HDMI
Ethernet	Gigabit	Gigabit	Gigabit	×
Precio aprox.	35€	45-75€	60-80€	15€

¿Cuál elegir?

- **Educación/básico:** Pi 3B+ o Zero 2W
- **Uso general:** Pi 4B (4GB)
- **Alto rendimiento:** Pi 5 (8GB)
- **Embebido:** Zero 2W

Rendimiento relativo

- Pi 3 → Pi 4: **+100 %**
- Pi 4 → Pi 5: **+200 %**
- Zero 2W \approx Pi 3 en CPU

Entretenimiento y Multimedia

- ▶ Centro multimedia (Kodi, Plex)
- ▶ Retroconsola (RetroPie)
- ▶ Consola de juegos retro

IoT y Automatización

- ▶ Domótica e IoT
- ▶ Estación meteorológica
- ▶ Cámaras de seguridad

Servidor y Red

- ▶ Servidor web/archivos
- ▶ Servidor de impresión
- ▶ NAS (almacenamiento)

Educación y Desarrollo

- ▶ Robótica y control
- ▶ Aprendizaje de programación
- ▶ Prototipado electrónico

Programación

Python:

- Control de hardware con GPIO Zero
- Automatización de tareas
- Análisis de datos con pandas

Scratch:

- Programación visual para niños
- Control de LEDs y sensores
- Primeros pasos en lógica

Electrónica Básica

Hardware:

- LEDs: control de brillo (PWM)
- Sensores: temperatura, distancia, luz
- Servomotores: control de posición
- Cámara Pi: visión artificial

Conceptos:

- Circuitos digitales (GPIO)
- Señales PWM y analógicas
- Protocolos: I2C, SPI, UART

Ejemplo 1: Sistema de alarma

Descripción:

- Alarma que detecta movimiento con sensor PIR y activa un buzzer y LED RGB.

Componentes:

- Sensor PIR (movimiento)
- Buzzer
- LED RGB
- Python + GPIO Zero

Aprendizaje:

- Entrada digital (sensor)
- Salida digital (LED/buzzer)
- Lógica condicional
- Manejo de eventos

Ejemplo 2: Robot evita obstáculos

Descripción:

- Robot móvil que usa sensor ultrasónico para detectar obstáculos y cambiar de dirección automáticamente.

Componentes:

- Sensor ultrasónico HC-SR04
- Motores DC + driver L298N
- Chasis con ruedas
- Python para control

Aprendizaje:

- Medición de distancia
- Control de motores (PWM)
- Algoritmos de decisión
- Integración de sistemas

Linux y Redes

Administración:

- Línea de comandos (bash)
- Gestión de archivos y permisos
- Instalación de software (apt)
- Scripts de automatización

Networking:

- Configuración de red (Wi-Fi/Ethernet)
- Servidor web (Apache/Nginx)
- SSH y acceso remoto
- Protocolos: HTTP, MQTT, TCP/IP

Sistema operativo

- Usar **Raspberry Pi Imager** → <https://www.raspberrypi.com/software/>.
- Sistema recomendado: **Raspberry Pi OS (basado en Debian)**.
- Alternativas: Ubuntu, RetroPie, LibreELEC, etc.



Software recomendado

Desarrollo

- ★ **VS Code** - Editor profesional
- ★ **Thonny** - IDE para principiantes

Programación

- ◇ **Python 3** - Lenguaje principal
- ◇ **GPIO Zero** - Control hardware

Stack tecnológico recomendado

Thonny + **Python 3** + **GPIO Zero** = ¡Éxito!

Ejemplo básico en Python

```
from gpiozero import LED
from time import sleep

led = LED(17)

while True:
    led.on()
    sleep(1)
    led.off()
    sleep(1)
```

Ejemplo: parpadeo de LED conectado al pin GPIO17.

```
from gpiozero import MotionSensor

pir = MotionSensor(4)

while True:
    pir.wait_for_motion()
    print("Movimiento detectado!")
    pir.wait_for_no_motion()
```