

Repaso Parcial 02

Dispositivos en IoT

Los dispositivos IoT son componentes físicos que detectan, interactúan o responden al entorno. Sirven como punto de contacto entre el mundo físico y el sistema digital, captando información o realizando acciones.

- **Sensores (Entrada/ Input):** Recogen datos del entorno físico (temperatura, luz, humedad, movimiento, etc.). Ejemplo: Sensor DHT11 (temperatura y humedad).
- **Actuadores (Salida/output):** Ejecutan acciones físicas, como encender un motor, abrir una válvula o prender una luz. Ejemplo: Relé, Servo motor, LED.
- **Dispositivos Híbridos:** Combinan sensores y actuadores en un solo módulo. Ejemplo: ESP32 puede leer un sensor y controlar una salida digital.

Hardware en IoT

El hardware IoT incluye todos los componentes electrónicos que permiten la operación del sistema, desde el procesamiento hasta la conexión de sensores y la comunicación con la red.

Componentes Principales

- **Microcontroladores y Microprocesadores:** Ejecutan el software, toman decisiones y procesan datos. Como Arduino UNO (microcontrolador), ESP32 (con WiFi/Bluetooth), Raspberry Pi (microprocesador).
- **Placas de Desarrollo:** Integran el microcontrolador con pines de entrada/salida, interfaces de comunicación, alimentación, etc. Facilitando conexiones, programación y prototipado. Como Arduino, NodeMCU, ESP32 DevKit, Raspberry Pi Pico.
- **Periféricos:** Sensores, actuadores, pantallas, botones, etc., que interactúan con el entorno físico.

Consideraciones

- Bajo consumo energético (especialmente en zonas rurales o con baterías).
- Capacidad de procesamiento y almacenamiento.
- Compatibilidad con módulos de comunicación.

Comunicación en IoT

La comunicación IoT se refiere a los medios y protocolos que permiten la transferencia de datos entre dispositivos, gateways (como Raspberry Pi para traducir protocolos de sensores a formatos compatibles con la nube), servidores y plataformas en la nube.

Tipos de Comunicación

- **Corto Alcance:**
 - WiFi: Alta velocidad, adecuado para redes domésticas o industriales con cobertura WiFi.
 - Bluetooth/BLE: Bajo consumo, ideal para dispositivos personales o de corto alcance.
 - Zigbee/Z-Wave: Comunicación en malla, eficiente para domótica y automatización.
- **Largo Alcance:**
 - LoRa/LoRaWAN: Muy bajo consumo, ideal para IoT rural o agrícola (LoRa).
 - Redes Móviles (NB-IoT, LTE-M, 4G, 5G): Uso de infraestructura celular para IoT masivo.
- **Cableada:**
 - **Serial (UART):** Comunicación punto a punto, sencilla (TX, RX).
 - **SPI:** Comunicación rápida con varios esclavos (4 líneas).
 - **I2C:** Comunicación con múltiples dispositivos mediante solo 2 líneas (SDA, SCL).
 - **Ethernet:** Conexión estable, baja latencia, ideal para entornos industriales.

Protocolos de Comunicación Digital

- **UART (Universal Asynchronous Receiver-Transmitter):** Protocolo de comunicación asíncrona entre dos dispositivos. Utiliza dos líneas: TX y RX.
- **SPI (Serial Peripheral Interface):** Protocolo síncrono de alta velocidad para comunicación entre un maestro y varios esclavos. Usa cuatro líneas: MOSI, MISO, SCLK y SS.
- **I2C (Inter-Integrated Circuit):** Protocolo síncrono que permite conectar múltiples dispositivos con dos líneas: SDA (datos) y SCL (reloj).

Protocolo	Tipo	Nº Cables	Velocidad	Dispositivos	Ventajas	Desventajas
UART	Asíncrono	2 (TX, RX)	Media (hasta 1 Mbps)	1 a 1	Simple, ampliamente usado	No apto para múltiples nodos
SPI	Síncrono	4	Alta (hasta 10 Mbps)	1 maestro + esclavos	Muy rápido, full-duplex	Usa muchos cables

Protocolo	Tipo	Nº Cables	Velocidad	Dispositivos	Ventajas	Desventajas
I2C	Síncrono	2	Media (100 kHz – 1 Mbps)	Hasta 127	Solo 2 cables, múltiples nodos	Más lento y complejo que SPI

Protocolos de Internet

- **MQTT (Message Queuing Telemetry Transport):** Ideal para IoT en tiempo real y comunicaciones eficientes entre muchos dispositivos, es estándar en plataformas como AWS IoT, Azure IoT y Node-RED.
- **HTTP/HTTPS:** Útil para integración con servicios web y APIs, pero más pesado.
- **COAP (Constrained Application Protocol):** Perfecto para dispositivos muy limitados y redes con alta pérdida de paquetes.

Característica	MQTT	HTTP/HTTPS	CoAP
Tipo	Pub/Sub	Request/Response	Request/Response optimizado
Modelo	Cliente ↔ Broker ↔ Cliente	Cliente ↔ Servidor	Cliente ↔ Servidor
Transporte	TCP	TCP	UDP
Ligereza	Muy ligero	Pesado	Muy ligero
Energía	Bajo consumo	Alto consumo	Muy bajo consumo
Seguridad	TLS/SSL	HTTPS (TLS)	DTLS (TLS sobre UDP)
Ideal para	IoT, redes inestables	Web, APIs RESTful	IoT muy limitado
Datos	Texto/binario (JSON)	Texto (HTML, JSON)	CBOR, JSON
Soporte QoS	Sí (niveles 0, 1, 2)	No	Limitado

Ejemplo de flujo de integración:

1. Sensor de humedad se conecta al ESP32.
 2. El microcontrolador procesa la lectura y la transmite vía WiFi.
 3. La información llega a la nube por MQTT.
 4. El usuario accede a los datos mediante una app o dashboard web.
-

Plataformas Populares para IoT:

¿Qué es una Plataforma IoT?

Un conjunto integrado de servicios, herramientas y tecnologías que permiten desarrollar, conectar, gestionar, monitorear y analizar dispositivos IoT de forma eficiente. Actúa como un puente entre el hardware (sensores, microcontroladores, actuadores) y el software (aplicaciones, dashboards, bases de datos). Funciones clave:

- Conectividad (MQTT, HTTP, CoAP).
- Gestión de dispositivos (registro, actualización, control remoto).
- Recolección y análisis de datos (procesamiento, alertas, visualización).
- Seguridad (autenticación, cifrado, control de acceso).

La elección de plataforma depende del costo, facilidad de uso, conectividad, capacidad de procesamiento y escalabilidad.

Arduino

Plataforma de hardware y software de código abierto para crear proyectos electrónicos interactivos.

- **Características:**
 - Basado en microcontroladores AVR, IDE Arduino con lenguaje simplificado basado en C/C++.
 - Interacción directa con sensores/actuadores.
 - Pines digitales, analógicos, PWM.
- **Modelos:** Uno, Nano, Mega, Leonardo. Se diferencian por número de pines, memoria y velocidad.
- **Ventajas:** Fácil de aprender, gran comunidad, bajo costo.
- **Desventajas:** Limitada capacidad de procesamiento, no tiene conectividad inalámbrica por defecto, no soporta multitarea
- **Aplicaciones:** Automatización básica del hogar, robótica educativa, lectura de sensores simples.

ESP32

Microcontrolador de alto rendimiento con Wi-Fi y Bluetooth integrados.

- **Características:**
 - CPU dual-core, RAM, conectividad, bajo consumo.

- Conectividad Wi-Fi, BT, BLE.
- Soporte para IDE Arduino, PlatformIO, ESP-IDF, MicroPython.
- GPIOs multifunción.
- PWM, ADC, DAC, SPI, I2C, UART.
- Deep Sleep (bajo consumo).
- Seguridad: cifrado de flash, arranque seguro.
- **Ventajas:** Potente, conectividad integrada, económico, multitarea básico.
- **Desventajas:** Curva de aprendizaje algo mayor, requiere cuidado en el manejo de voltajes.
- **Aplicaciones:** Proyectos de IoT con conexión a la nube, control domótico, dataloggers remotos.

Raspberry Pi

Computadora de placa reducida (SBC). desarrollada para educación y proyectos avanzados. Funciona con sistema operativo y puede ejecutar aplicaciones completas. Y puede actuar como **gateway Edge Computing**, evitando la dependencia de la nube.

- **Modelos:** Pi 3B+, 4, 400, Zero 2 W. Varían en RAM, conectividad, puertos y rendimiento.
- **Características:**
 - CPU ARM de 64 bits, RAM, almacenamiento
 - Puertos USB, HDMI, Ethernet, GPIOs.
 - Almacenamiento mediante microSD o SSD.
- **Sistema Operativo:**
 - Raspberry Pi OS (Debian).
 - También compatible con Ubuntu, Windows IoT, LibreELEC.
 - Soporte para múltiples lenguajes: Python (**MicroPython**, ideal para scripts rápidos), Node.js, Java, etc.
- **Ventajas:** Computadora completa (multitarea, navegación, acceso remoto), muy versátil, ideal para gateways IoT, procesamiento local.
- **Desventajas:** Mayor consumo energético, requiere sistema de archivos y administración. Requiere conocimientos de sistema operativo. No apta para tareas de tiempo real estricto.
- **Aplicaciones:** Gateways para dispositivos IoT, servidores web locales, monitorización de cámaras.

Comparación Final

Criterio	Arduino	ESP32	Raspberry Pi
Complejidad	Bajo	Medio	Alto

Criterio	Arduino	ESP32	Raspberry Pi
Costo	Bajo	Bajo	Medio/Alto
Conectividad	No (por defecto)	Wi-Fi, BT	Wi-Fi, BT, Ethernet
Sistema operativo	No	No	Sí (Linux)
Uso típico	Sensores simples	IoT conectados	Gateways, procesamiento

¿Cuándo usar cuál?

- Arduino: sensores simples, bajo consumo, proyectos educativos.
- ESP32: IoT con conectividad, sensado distribuido, autonomía media, control inalámbrico.
- Raspberry Pi: procesamiento intensivo, almacenamiento, comunicación avanzada, - integración con la nube o visión artificial.

Desarrollo de Proyectos IoT

¿Qué es un Proyecto IoT?

Un sistema tecnológico que integra dispositivos físicos con plataformas digitales para capturar, procesar y transmitir datos del entorno físico a través de Internet, permitiendo la toma de decisiones automatizada o informada.

- Herramientas como Blynk, Cayenne o TagoIO facilitan la creación rápida de interfaces.
- Normativas como ISO/IEC 30141 proporcionan marcos de referencia para arquitectura IoT.

Características de los sistemas IoT:

- Interconectividad de objetos.
- Automatización de procesos.
- Acceso remoto y tiempo real.
- Escalabilidad desde prototipos hasta sistemas masivos.

Componentes de un Proyecto IoT

- Sensores y actuadores
- Plataformas de procesamiento (microcontroladores, microcomputadoras)
- Protocolos de comunicación
- Servicios en la nube o aplicaciones móviles/web

Etapas del Ciclo de Vida de un Proyecto IoT

1. Fase 1: Identificación de la Necesidad

- ¿Qué necesidad se quiere resolver?
- ¿A quién está dirigido el proyecto?
- ¿Cuál es el impacto?

2. Fase 2: Requerimientos y Diseño

- A. Lista de funcionalidades deseadas: Como Monitoreo de variables, Control de dispositivos, Notificaciones ante eventos, Visualización de datos en tiempo real, Automatización de acciones, etc.
- B. Diseño lógico del sistema (diagrama de bloques): Representa cómo fluyen los datos entre componentes:
Sensores → Microcontrolador → Actuadores/Red → Plataforma/Usuario.
- C. Definición de componentes (IoT): Sensores, actuadores, microcontroladores, fuentes de energía, tecnologías de red.

3. Fase 3: Desarrollo e Integración

- A. Programación del microcontrolador
- B. Configuración de red
- C. Integración con plataforma en la nube o local

4. Fase 4: Pruebas y Validación

- A. Pruebas funcionales
- B. Pruebas de conectividad
- C. Pruebas de respuesta ante fallos

5. Fase 5: Despliegue y Mantenimiento

- A. Instalación física: Ubicación y protección de sensores, Alimentación confiable (con respaldo), Montaje estructural seguro, Evitar interferencias físicas/electromagnéticas.
- B. Monitoreo remoto
- C. Actualizaciones (firmware OTA)
- D. Seguridad

Flujo General de un Proyecto IoT

Sensores → Microcontrolador (con código) → Red (WiFi, LoRa, etc.) → Plataforma (nube/local)
→ Visualización/acción