

# Trabajo Práctico N° 1

## **Módulo 1**

Protocolos de Comunicaciones

## **Profesor**

Jorge Elias Morales

## **Integrantes**

Fernando Gimenez Coria - [FerCbr](#)

Nicolás Barrionuevo - [NicolasB-27](#)

Macarena Aylen Carballo - [MacarenaAC](#)

Raul Jara - [r-j28](#)

Diego Ezequiel Ares - [diegote7](#)

Juan Diego González Antoniazzi - [JDGA1997](#)

## **Fecha de entrega**

Lunes 14 de Abril 2025

## Índice

1. Nombre, describa y grafique las capas OSI.....	5
🌐 Modelo OSI 🌐.....	5
🔄 Comunicación entre Capas del Modelo OSI.....	6
📦 Proceso de Encapsulación (Ejemplo: Envío de un Mensaje HTTP).....	7
🔄 Desencapsulación en el Destino.....	8
🌐 Ejemplo Práctico: Navegación Web (HTTP + TCP/IP).....	8
⚡ Interacción Clave entre Capas.....	9
2. ¿Cómo se comunican los dispositivos IoT?, Esquematizar y ejemplificar.....	9
3. Ejemplifique que son dispositivos IoT.....	16
4. ¿Qué tecnologías han hecho posible el IoT menciona 5 ejemplos?.....	18
5. ¿Que es la comunicación de datos en serie? Describa su funcionamiento.....	19
6. ¿Cuáles son los protocolos serie más usados?, nombre 3 y descríbalos.....	22
1. UART (Universal Asynchronous Receiver Transmitter).....	23
2. I2C (Inter-Integrated Circuit).....	24
3. SPI (Serial Peripheral Interface).....	26
7. ¿Qué es la comunicación de datos en paralelo? Describa su funcionamiento.....	28
8. ¿Cuáles son los protocolos paralelo más usados?, nombre alguno y descríbalos.....	30
1. Parallel ATA (PATA) / IDE.....	30
2. SCSI (Small Computer System Interface).....	31
3. IEEE 1284 (Puerto Paralelo Centronics).....	31
4. PCI (Peripheral Component Interconnect).....	31
5. IEEE-488 (GPIB/HPIB).....	32
Comparativa de Protocolos Paralelos.....	32

## 1. Nombre, describa y grafique las capas OSI

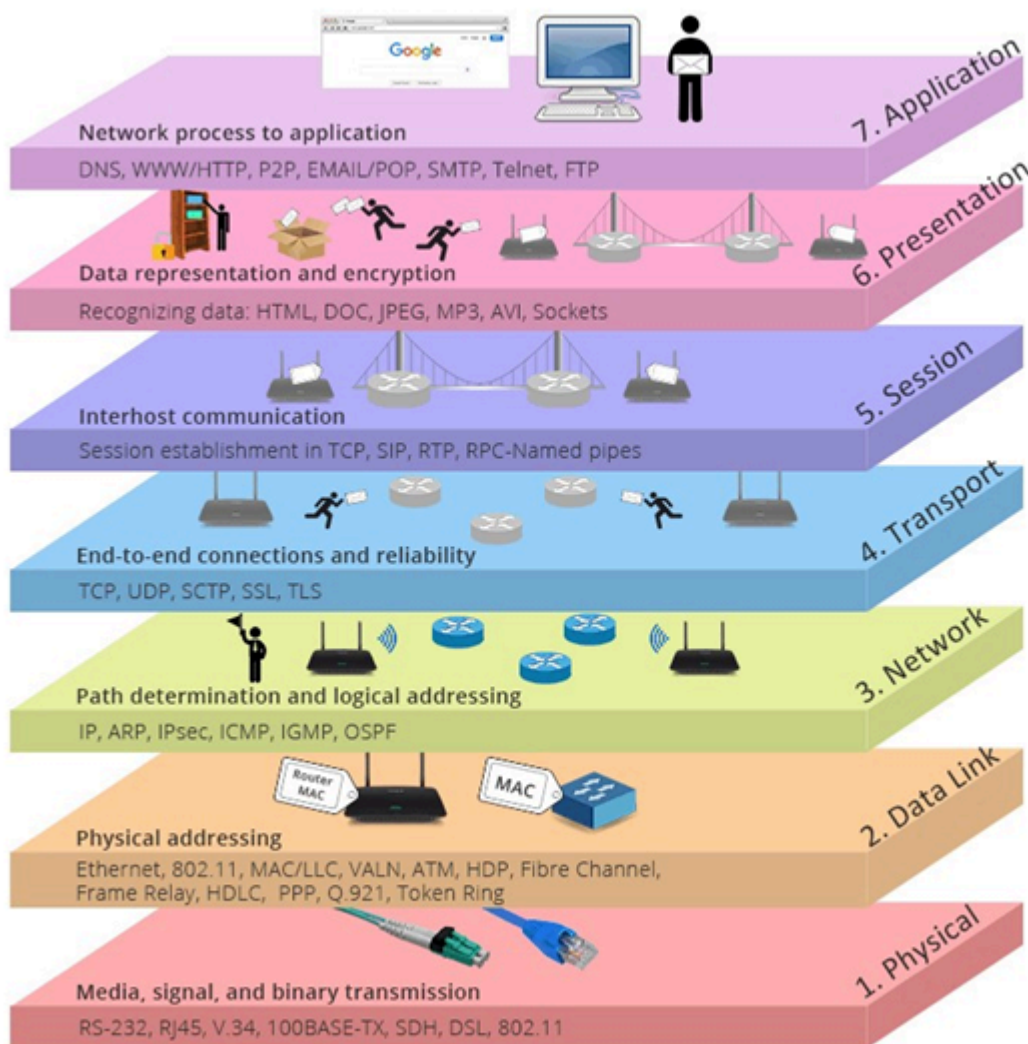
### **Modelo OSI**

El modelo OSI, de siete capas, es un modelo conceptual que caracteriza y estandariza la manera en la que los diferentes componentes de software y hardware involucrados en una comunicación de red deben dividir el trabajo e interactuar entre sí.

- ♦ Capas del modelo OSI.  
(de abajo hacia arriba)

Capa	Nombre	Función Principal	Ejemplo de Protocolos/Dispositivos
Capa 1		Transmisión de bits crudos (señales eléctricas, ópticas o inalámbricas).	Cable UTP, fibra óptica, hubs, repetidores.
Capa 2	Enlace de Datos	Manejo de tramas (frames), detección/corrección de errores y control de flujo.	Ethernet, bridges.
Capa 3	Red	Enrutamiento de paquetes entre redes (direcciones lógicas como IP).	IP, ICMP,
Capa 4	Transporte	Comunicación extremo a extremo (confiable), UDP (no confiable o no). Segmentación de datos. (confiable o no).	TCP
Capa 5	Sesión	Establece, gestiona y termina conexiones entre aplicaciones.	NetBIOS, RPC.
Capa 6	Presentación	Traduce datos (encriptación, compresión, formatos).	SSL/TLS, JPEG, ASCII.

<b>Capa 7</b>	Interfaz directa con el usuario o	
<b>Aplicación</b>	SMTP, DNS.	HTTP, FTP,
	aplicaciones (servicios de red).	



## Comunicación entre Capas del Modelo OSI

El modelo OSI sigue un principio de encapsulación/desencapsulación: cada capa añade información específica (encabezado o header) a los datos originales,

formando una PDU (Unidad de Datos de Protocolo) con nombres distintos en cada capa.

## Proceso de Encapsulación (Ejemplo: Envío de un Mensaje HTTP)

### 1. Capa 7 (Aplicación):

- El usuario escribe **www.ejemplo.com** en el navegador.
- Datos originales: **GET / HTTP/1.1** (solicitud HTTP).
- PDU: Mensaje.

### 2. Capa 6 (Presentación):

- Encripta (SSL/TLS) o comprime los datos si es necesario.
- PDU: Mensaje + header de Presentación.

### 3. Capa 5 (Sesión):

- Establece una sesión con el servidor (ej. autenticación).
- PDU: Mensaje + header de Sesión.

### 4. Capa 4 (Transporte):

- Divide los datos en segmentos (TCP) o los envía tal cual (UDP).
- Añade puertos de origen/destino (ej. **Puerto 80** para HTTP).
- PDU: Segmento (TCP/UDP).

### 5. Capa 3 (Red):

- Añade direcciones IP (ej. **192.168.1.1 → 8.8.8.8**).
- PDU: Paquete (IP + segmento).

### 6. Capa 2 (Enlace de Datos):



- Convierte el paquete en una trama (frame).
- Añade direcciones MAC (ej. **00:1A:2B:3C:4D:5E**).
- PDU: Trama (Frame).

## 7. Capa 1 (Física):

- Convierte la trama en bits (señales eléctricas/ópticas).
- PDU: Bits (transmitidos por cable, WiFi, fibra, etc.).



## Desencapsulación en el Destino

El proceso inverso ocurre en el receptor:

1. Capa 1 (Física): Recibe bits y los convierte en una trama.
2. Capa 2 (Enlace): Verifica la MAC y pasa el paquete a la capa 3.
3. Capa 3 (Red): Revisa la IP y lo envía al protocolo correcto (TCP/UDP).
4. Capa 4 (Transporte): Reensambla los segmentos (si es TCP).
5. Capas 5-7 (Sesión, Presentación, Aplicación): Desencrpta, interpreta y muestra la página web.



## Ejemplo Práctico: Navegación Web (HTTP + TCP/IP)

1. Usuario → Escribe **http://www.ejemplo.com**.
2. Capa 7 (HTTP): Solicitud GET.
3. Capa 4 (TCP): Establece conexión (3-way handshake).
4. Capa 3 (IP): Enruta el paquete hacia el servidor.
5. Capa 2 (Ethernet): Switches usan MAC para dirigir la trama.

6. Capa 1: Bits viajan por fibra óptica/cobre.

→ Respuesta: El servidor repite el proceso inverso y envía la página web.

## Interacción Clave entre Capas

Capa	¿Qué usa?	¿Qué entrega?
<b>Aplicación</b>	DNS (resuelve nombres)	Datos para Presentación.
<b>Transporte</b>	Puertos (ej. 80, 443)	Segmentos a Red.
<b>Red</b>	IP (enrutamiento)	Paquetes a Enlace.
<b>Enlace</b>	MAC (switches)	Tramas a Física.

### Errores Comunes por Capa

- Capa 1: Cable desconectado, señal débil.
- Capa 2: MAC incorrecta, switch mal configurado.
- Capa 3: IP errónea, falla en el router.
- Capa 4: Puerto bloqueado (firewall).
- Capa 7: DNS no responde, HTTP 404.

## 2. ¿Cómo se comunican los dispositivos IoT?, Esquematizar y ejemplificar.

La Internet de las Cosas (IoT) se caracteriza por la interconexión de dispositivos físicos –tales como sensores, actuadores, cámaras, medidores, entre otros, que se comunican para recolectar, transmitir y procesar información. La finalidad principal es capturar datos en tiempo real y, a partir de éstos, automatizar procesos, optimizar recursos y tomar decisiones informadas.

En esencia, la comunicación en IoT se apoya en la integración de hardware y software. Esto implica que los dispositivos no sólo capturan datos sino que también deben transmitirlos de forma eficiente y segura a sistemas de procesamiento (locales o en la nube) mediante protocolos de comunicación específicos.

Antes de entrar en detalles técnicos, es útil considerar las tres capas fundamentales de la arquitectura de IoT:

- **Dispositivos/End Points:** Son los nodos que capturan la información del entorno (mediante sensores) y pueden, a su vez, ejecutar acciones (actuadores).
- **Gateways/Concentradores:** Actúan como intermediarios; recogen datos de varios dispositivos y los transmiten, muchas veces realizando procesamiento local (edge computing) para reducir la latencia o filtrar datos antes de enviarlos a la nube.
- **Nube/Servidores:** Son sistemas donde se centraliza el almacenamiento y análisis de datos, permitiendo además la implementación de aplicaciones de control, visualización y toma de decisiones.

## Arquitectura de Comunicación en IoT: Esquema General

Para comprender cómo se comunican los dispositivos IoT, se puede esquematizar la comunicación en tres niveles:

### 1. Comunicación Directa entre Dispositivos:

Algunos dispositivos pueden comunicarse directamente entre sí mediante tecnologías de corto alcance. Por ejemplo, en un entorno de domótica, una bombilla inteligente puede sincronizarse con un sensor de movimiento usando protocolos como Zigbee o Bluetooth Low Energy (BLE).

Esta comunicación generalmente forma una red en malla (mesh network), lo que incrementa la resiliencia y el alcance de la red sin depender exclusivamente de un concentrador central.

### 2. Comunicación a través de un Gateway:

En la mayoría de las implementaciones IoT, los dispositivos se comunican a través de un gateway.

#### o Función del Gateway:

- § Recibe datos de múltiples dispositivos mediante protocolos como Zigbee, LoRaWAN, NB-IoT, Wi-Fi o Bluetooth.
- § Realiza tareas de pre-procesamiento o “edge computing”, reduciendo la carga de datos que deben enviarse a la nube.
- § Transmite la información recopilada hacia un servidor (en la nube o local) usando protocolos de red, como MQTT, HTTP o CoAP.

#### o Ejemplo Práctico:

En una ciudad inteligente, los sensores instalados en postes de luz o señales de tráfico envían datos al gateway local, el cual los envía en lotes o



en tiempo real a un centro de control para gestionar la iluminación y el flujo vehicular.

### 3. Comunicación con la Nube:

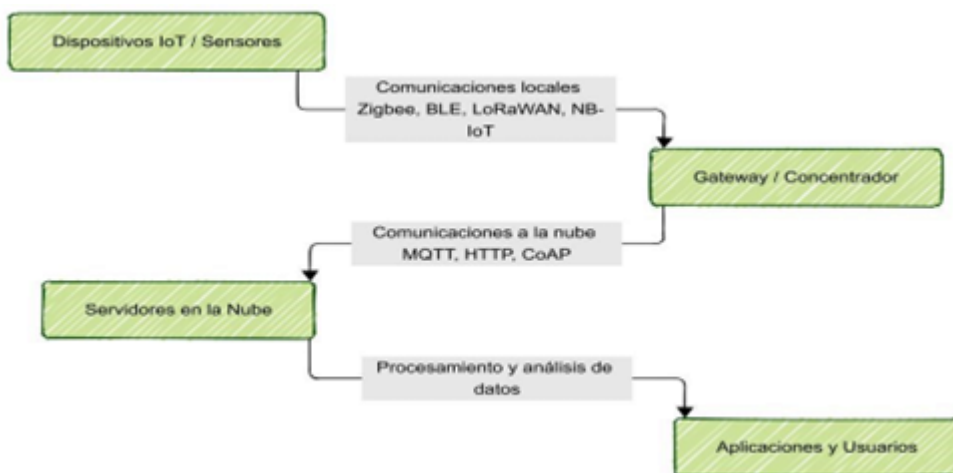
La comunicación con la nube permite el almacenamiento y procesamiento masivo de datos. Esto es esencial para:

- o Analizar grandes volúmenes de información en tiempo real. o Aplicar algoritmos de inteligencia artificial o machine learning.
- o Proveer interfaces de usuario (dashboards) para la monitorización de la red.

#### o Ejemplo Práctico:

Sistemas como AWS IoT Core o Azure IoT Hub permiten la conexión y gestión de miles de dispositivos, proporcionando además funciones de seguridad y administración de dispositivos.

A continuación se muestra un diagrama esquemático que ilustra este flujo de comunicación:



## Protocolos y Tecnologías de Comunicación en IoT

Existen múltiples protocolos y tecnologías que permiten la comunicación entre dispositivos IoT. A continuación, se detallan algunos de los más comunes:

### a) Protocolos de Comunicación a Nivel de Aplicación

#### • MQTT (Message Queuing Telemetry Transport):

Es un protocolo ligero diseñado para conexiones de baja latencia y con poco consumo de ancho de banda. Ideal para redes IoT, MQTT funciona con un modelo “publica/suscribe”, lo que permite que dispositivos se comuniquen de forma asíncrona.

- o Ejemplo: Sensores ambientales que envían lecturas periódicas a un broker MQTT, el cual distribuye los mensajes a aplicaciones de monitorización.

- **CoAP (Constrained Application Protocol):**

Diseñado para dispositivos con recursos limitados, CoAP utiliza el modelo cliente/servidor similar a HTTP, pero de manera más eficiente en términos de comunicación y consumo de energía.

- o Ejemplo: Dispositivos de medición de consumo energético en edificios, que notifican cambios significativos a través de CoAP.

**b) Protocolos y Tecnologías de Comunicación a Nivel Físico y de Enlace Wi-Fi y Ethernet:**

Usados principalmente en entornos donde la alta velocidad y la gran cantidad de datos son importantes, como en aplicaciones industriales o en oficinas inteligentes.

- **Bluetooth / BLE:**

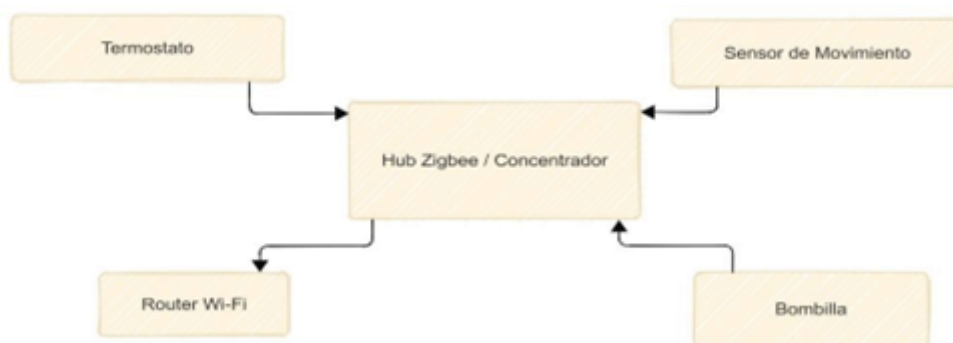
Ideal para comunicaciones de corto alcance y aplicaciones que requieren bajo consumo de energía.

- o Ejemplo: Wearables o dispositivos de asistencia en el hogar.

- **Zigbee:**

Diseñado para aplicaciones de domótica y control de iluminación, Zigbee utiliza una red en malla que permite interconectar cientos o incluso miles de dispositivos.

- o Ejemplo: El sistema de iluminación Philips Hue, donde las bombillas y sensores se comunican a través de un hub central mediante Zigbee.
- o Diagrama Simplificado de una red Zigbee:



- **LoRa y LoRaWAN:**

LoRa es el protocolo a nivel físico que permite comunicaciones a largas distancias con bajo consumo de energía, mientras que LoRaWAN (que opera en niveles superiores) se encarga de la gestión de direcciones y la coordinación en la red.

- o Ejemplo: Sensores en una red agrícola que monitorean la humedad del suelo; cada sensor transmite la información a un gateway mediante LoRa y este la remite a un sistema central para análisis y acciones de riego.

- **NB-IoT (Narrowband IoT):**

Esta tecnología utiliza bandas estrechas del espectro celular para ofrecer comunicaciones de largo alcance y muy bajo consumo de energía, ideal para

dispositivos que se comunican de forma infrecuente (por ejemplo, medidores inteligentes en zonas de difícil acceso).

- o Ejemplo: Medidores de agua instalados en sótanos o áreas remotas que, gracias a NB-IoT, transmiten lecturas periódicas a una central de datos.

Cada una de estas tecnologías tiene sus ventajas y limitaciones; la elección adecuada depende del caso de uso (por ejemplo, alcance, consumo de energía, costo y densidad de dispositivos).

## Ejemplos Prácticos y Casos de Uso

Para ilustrar de forma concreta cómo se comunican los dispositivos IoT, presentamos dos ejemplos:

### Ejemplo 1 – Smart Home (Hogar Inteligente):

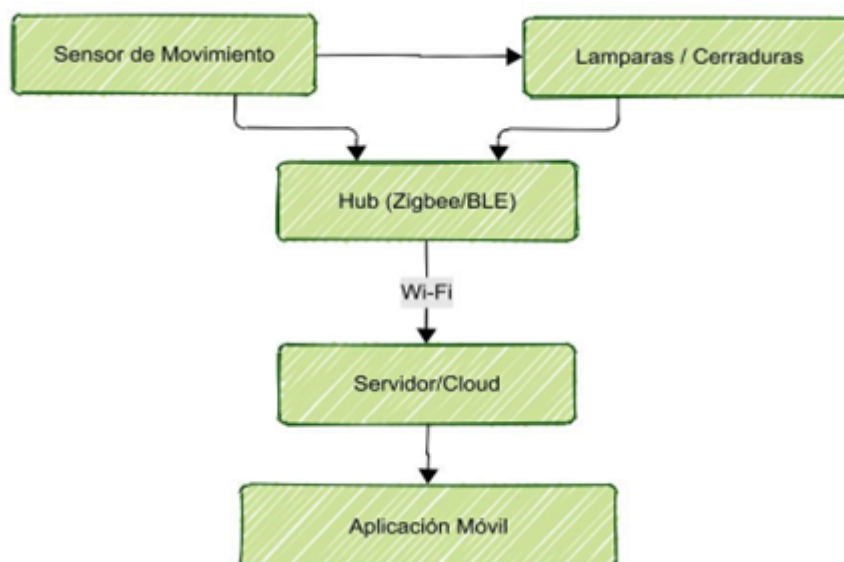
Componentes involucrados:

- **Dispositivos:** Bombillas inteligentes, sensores de temperatura, sensores de movimiento, cerraduras inteligentes.
- **Comunicación Local:** Muchos de estos dispositivos utilizan tecnologías como Zigbee o BLE para comunicarse entre sí.
- **Gateway:** Un concentrador (hub) que integra estos dispositivos y los conecta a la red doméstica vía Wi-Fi.
- **Nube y Aplicaciones:** Los datos son enviados a servidores en la nube donde se integran con aplicaciones móviles que permiten el control y monitoreo en tiempo real.

Flujo de Comunicación:

1. Un sensor de movimiento detecta actividad y envía una señal mediante Zigbee al hub central.
2. El hub procesa la señal y, a través de MQTT, la envía a una aplicación en la nube.
3. La aplicación analiza la información y, si es necesario, envía una orden al sistema de iluminación para encender las luces.

Diagrama Esquemático:



## Ejemplo 2 – Agricultura Inteligente:

Componentes involucrados:

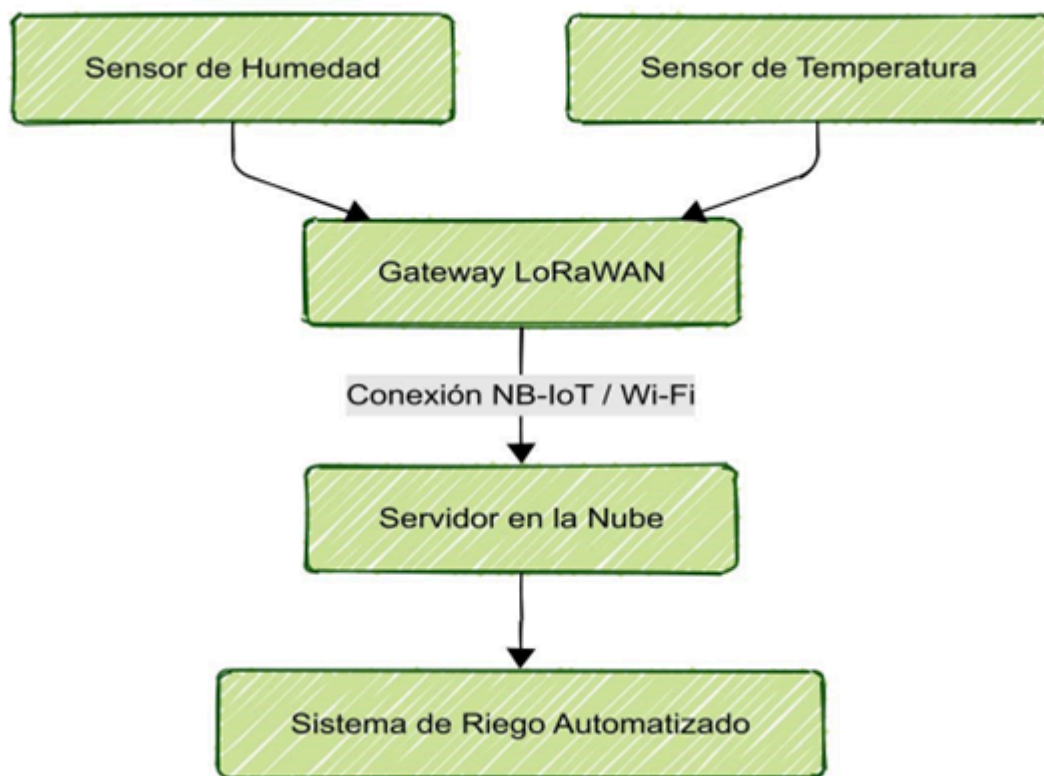
- **Dispositivos:** Sensores de humedad y temperatura en el suelo, estaciones meteorológicas.
- **Comunicación:** Se utilizan tecnologías de largo alcance como LoRa o NB-IoT para conectar sensores dispersos en grandes extensiones agrícolas.
- **Gateway:** Un concentrador ubicado estratégicamente en la granja que recoge datos de los sensores.
- **Nube:** Los datos se transmiten a la nube para permitir análisis predictivos y activar sistemas de riego de manera automática.

Flujo de Comunicación:

1. Los sensores instalados en distintos puntos de un campo miden la humedad y temperatura del suelo.
2. Mediante LoRaWAN, los datos se transmiten a un gateway que actúa como concentrador.
3. El gateway envía la información a una plataforma en la nube, donde se analizan las condiciones.
4. De acuerdo con el análisis, el sistema puede activar un riego automático en zonas específicas.

Diagrama Esquemático:





## Consideraciones de Seguridad y Gestión

La comunicación en IoT no solo se trata de transmitir datos; la **seguridad** es un aspecto crítico. Algunas de las medidas implementadas son:

- **Cifrado en Tránsito y en Reposo:** Utilizando protocolos de seguridad (por ejemplo, TLS/SSL para MQTT o HTTPS) se asegura que la información transmitida esté protegida.
- **Autenticación y Autorización:** Los dispositivos y gateways deben estar autenticados para evitar accesos no autorizados.
- **Actualizaciones Remotas y Gestión de Firmware:** Plataformas de gestión de dispositivos permiten mantener la seguridad de toda la red a través de actualizaciones remotas, sin necesidad de intervención manual.

Además, una adecuada **gestión de la red** se logra con plataformas de administración que permiten:

- Registrar y monitorear cada dispositivo.
- Configurar alertas en caso de mal funcionamiento o intentos de acceso no autorizados.
- Visualizar el estado de la red y efectuar diagnósticos de manera centralizada.



---

## 6. Conclusiones

---

La comunicación entre dispositivos IoT es un proceso complejo y escalable que se sostiene sobre una arquitectura de múltiples niveles. Desde la captura y transmisión inicial de datos en el dispositivo hasta el procesamiento avanzado en servidores en la nube, cada etapa –que incluye la intermediación por gateways y la utilización de diversos protocolos y tecnologías (como Zigbee, LoRaWAN, NB-IoT, MQTT, entre otros)– está diseñada para optimizar la eficiencia, reducir el consumo energético y garantizar la seguridad.

Los ejemplos prácticos, como la implementación de hogares inteligentes o soluciones en agricultura, demuestran la aplicabilidad de estas tecnologías en entornos reales. La elección de la tecnología de comunicación depende de numerosos factores: distancia, consumo, costo, densidad de nodos y el tipo de datos transmitidos.

Finalmente, es fundamental considerar que en IoT los dispositivos, al estar distribuidos y ser numerosos, requieren de sistemas de gestión centralizados que permitan supervisar, actualizar y asegurar de forma continua toda la infraestructura. Con ello se garantiza la robustez y eficiencia de los procesos automatizados, ofreciendo soluciones que transforman diversos sectores –desde la domótica hasta la industria– y abren la puerta a un mundo cada vez más conectado.

## 3. Ejemplifique que son dispositivos IoT

Los dispositivos IoT (Internet de las Cosas) son objetos físicos equipados con sensores y conectividad a internet, lo que les permite recopilar, enviar e incluso procesar datos de forma automática. Estos dispositivos están diseñados para interactuar entre sí y con plataformas digitales, lo que facilita una amplia gama de aplicaciones en diversos sectores.

- 1- En la salud, se usan relojes inteligentes (Apple Watch, Fitbit) que monitorizan la frecuencia cardíaca, el sueño y la actividad física, además de sensores médicos conectados que ayudan a seguir parámetros vitales en pacientes crónicos.



- 2- En la industria, sensores IoT detectan temperatura, humedad o vibración en equipos para prevenir fallos mediante mantenimiento predictivo.



- 3- En agricultura, hay sensores de humedad del suelo que permiten activar automáticamente sistemas de riego, optimizando el uso del agua.



Dado a su capacidad de conectividad y recopilación de datos, los dispositivos IoT esto permite mejorar la toma de decisiones, automatizar procesos, aumentar la seguridad y crear entornos más inteligentes y eficientes. Tecnologías como el 5G, la inteligencia artificial y el edge computing potencian aún más sus capacidades y aplicaciones.

#### 4. ¿Qué tecnologías han hecho posible el IoT menciona 5 ejemplos?

El Internet de las Cosas (IoT) se ha hecho posible gracias a tecnologías de conectividad, sensores, actuadores, computación en la nube y otras tecnologías.

Conectividad: Wifi, Bluetooth, Telefonía móvil, Zigbee, LoRaWAN.

Sensores y actuadores: Captan información de los objetos conectados. Los sensores detectan cambios ambientales como la temperatura, la humedad, la luz, el movimiento o la presión.

Computación en la nube: La nube es donde se almacenan, procesan y analizan las grandes cantidades de datos generados por los dispositivos IoT.

Computación de borde: Aumenta la potencia de computación en los bordes de una red de IoT, lo que reduce la latencia de las comunicaciones y mejora el tiempo de respuesta.

Realidad aumentada: Se puede usar para visualizar datos de cientos de sensores simultáneamente.

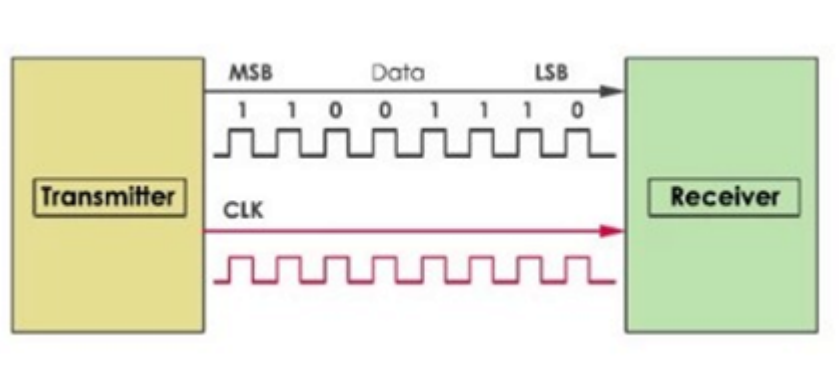
Inteligencia artificial (IA): Se combinan con IoT para ayudar a los consumidores a mejorar el racionamiento y comprender su uso de energía.

El IoT se ha aplicado en: Hogares inteligentes, Ciudades inteligentes, Autos conectados, Comercio minorista, Telemedicina.



## 5. ¿Que es la comunicación de datos en serie? Describa su funcionamiento.

La comunicación en serie es un método de transmisión de datos que envía y recibe información bit a bit, a través de un único canal de comunicación. Se usa para intercambiar datos entre computadoras y dispositivos periféricos. La transmisión serie entre el emisor y el receptor está sujeta a protocolos estrictos que proporcionan seguridad y fiabilidad y han llevado a su longevidad. Muchos dispositivos, desde ordenadores personales hasta dispositivos móviles, utilizan la comunicación en serie.



### Funcionamiento:

- Utiliza una o dos líneas de transmisión, una para enviar datos (Tx) y otra para recibirlos (Rx).

- Los datos se envían y reciben continuamente, un bit a la vez.
- Los protocolos de comunicaciones serie incluyen señales de sincronización y control, como bits de inicio y parada.
- El dispositivo receptor vuelve a ensamblar todos los datos transmitidos bit a bit. Usos:
- Se usa en muchos dispositivos, desde ordenadores personales hasta dispositivos móviles.
- El protocolo RS232 se utiliza en PC de generaciones anteriores para conectar dispositivos periféricos como mouse, impresoras, módems, etc.
- También se utiliza en máquinas PLC, máquinas CNC y servocontroladores.

### Consideraciones:

- La comunicación en serie está diseñada para permitir que solo dos dispositivos se comuniquen a través de un bus en serie.
- Si más de un dispositivo está intentando transmitir en la misma línea serie, podría encontrarse con una contención de bus.

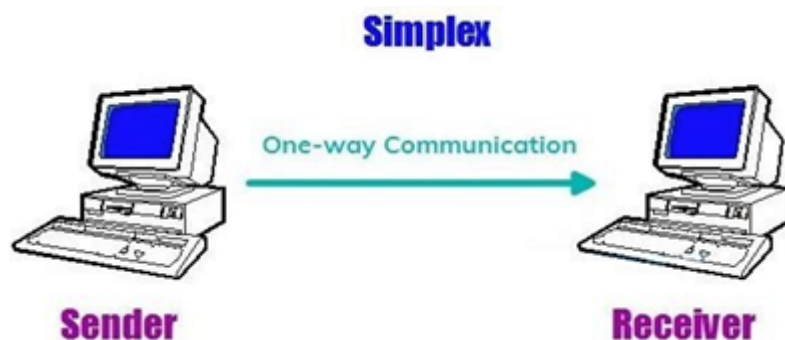
### Modos Básicos de Transmisión de Datos Serie

En la transmisión de datos en serie se utilizan pulsos binarios para transmitir los datos. El dígito binario uno está representado por cinco voltios o una lógica ALTA. Por el contrario, el cero binario se denota con una lógica BAJA o cero voltios. Para implementar la comunicación en serie, se requieren un origen y un destino. También se les conoce como emisor y receptor. Se pueden emplear varios tipos de comunicación serie y se designan como Simplex, Half Duplex y Full Duplex.

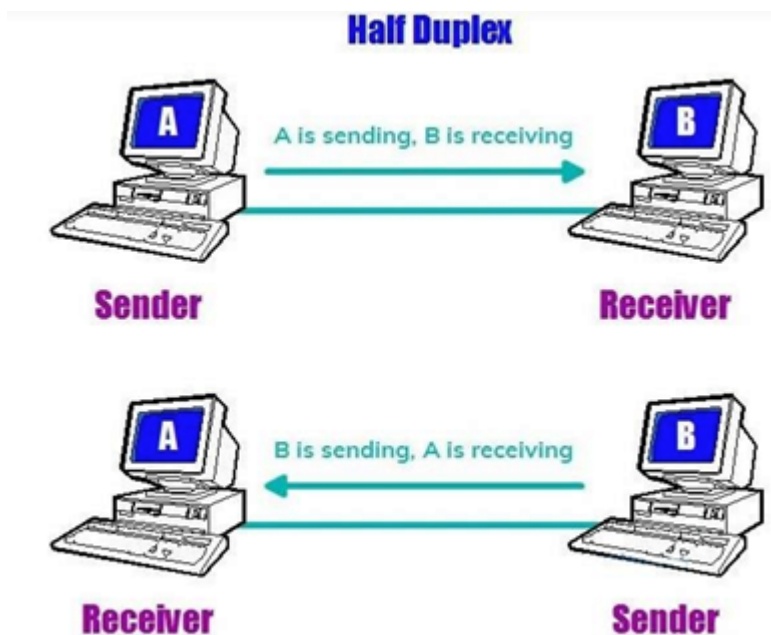
### Modos de transmisión

El **método Simplex** implementa la transmisión de datos unidireccional. En este esquema, solo el origen o el destino están activos en un momento dado. Si la fuente está enviando datos, el receptor no tiene más remedio que aceptar la transmisión. El modo Simplex se usa para transmitir señales de televisión o radio.





**Modo Half Duplex** permite que el origen y el destino estén activos, pero no simultáneamente. La transmisión solo ocurre en una dirección a la vez. Un caso en cuestión se puede ver al usar Internet. Cuando usted realiza una solicitud desde su ordenador para una página web, el servidor procesa la solicitud y luego se convierte en el remitente cuando devuelve la información a su ordenador, que ahora es el receptor.



**Modo Full Duplex** es la forma de comunicación serie más utilizada en el mundo. El origen y el destino están activos y pueden enviar y recibir datos simultáneamente. Su teléfono inteligente es un excelente ejemplo del modo full duplex en acción.



Otra consideración al analizar la comunicación en serie es el protocolo y la resistencia de los dos ordenadores host.

### ¿Cómo Funciona la Comunicación Serie? Ejemplo.

La comunicación serie es utilizada por microcontroladores y microprocesadores modernos para la transferencia de datos internos y externos. Un ejemplo simple e ilustrativo de enviar un archivo desde su ordenador portátil a un teléfono inteligente. Probablemente lo envíe utilizando el protocolo WiFi o Bluetooth.

Establecer una comunicación serie requiere estos pasos:

- Crear una conexión.
- El portátil realizará una búsqueda de dispositivos cercanos y proporcionará una lista de los que ha descubierto.
- Seleccione el dispositivo con el que desea comunicarse.

Su teléfono inteligente debe estar emparejado para completar la conexión. El software funciona con los valores predeterminados, por lo que no necesita configurar los parámetros manualmente. Hay cuatro parámetros que afectan la comunicación: velocidad de transmisión, selección de bits de datos (trama), bit de inicio-parada y paridad.

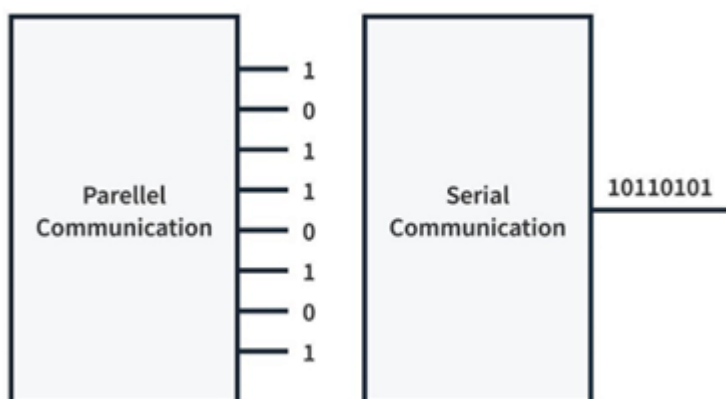
6. ¿Cuáles son los protocolos serie más usados?, nombre 3 y descríbalos.

En el contexto de las telecomunicaciones e IoT, los protocolos de comunicación serie permiten la transmisión de datos entre dispositivos utilizando una secuencia de bits a través de una línea de comunicación. A continuación, se describen los tres protocolos serie más utilizados:

## 1. UART (Universal Asynchronous Receiver Transmitter)

Es uno de los protocolos de comunicación más utilizados en electrónica embebida. Es un protocolo de comunicación serie, full-duplex, asíncrono y de placa a placa. Antes de comprender el protocolo, analicemos algunas palabras clave importantes.

- *Comunicación serial:* Para la comunicación serial, solo se requieren una o dos conexiones para enviar o recibir datos. La recepción y transmisión de datos se realiza bit a bit o secuencialmente.
- *Comunicación paralela:* En la comunicación paralela, se utilizan múltiples líneas para el intercambio de datos. El número de líneas utilizadas para la transmisión o recepción de datos se denomina ancho de bus de un dispositivo. En la figura a continuación, se puede observar que el ancho de bus es de 8 bits. Los 8 bits se transfieren simultáneamente.

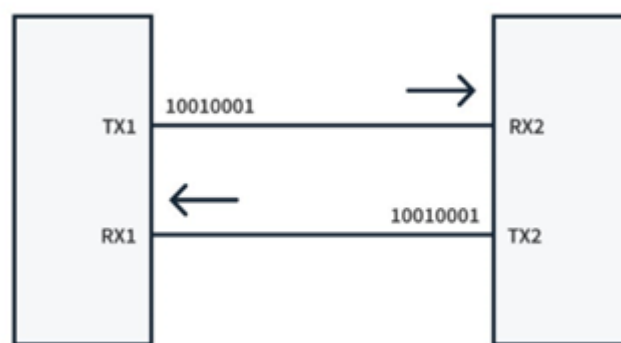


- *Comunicación asíncrona:* En la comunicación asíncrona, los dispositivos no comparten un reloj común para sincronizar la transferencia de datos. En su lugar,

acuerdan una velocidad de transferencia idéntica y configuran su reloj interno para capturar o transferir los datos.

- **Comunicación full-duplex:** En este tipo de comunicación, se utilizan líneas separadas para transmitir y recibir simultáneamente.

En la figura a continuación, se puede observar que el dispositivo envía y recibe datos en paralelo.



Características:

- Tipo: Asíncrono (no necesita señal de reloj compartida).

Características:

- ✓ Utiliza dos líneas principales: TX (transmisión) y RX (recepción).
- ✓ Permite la comunicación punto a punto (1 a 1).
- ✓ Es común en microcontroladores y módulos como ESP32, Arduino, etc.

- Ventajas: Simple, económico y ampliamente soportado.
- Aplicaciones: Comunicación entre microcontroladores, sensores, módulos GPS, Bluetooth, etc.

## 2. I2C (Inter-Integrated Circuit)

I2C <sup>es</sup> un protocolo síncrono y es el que tiene algo de “intelecto” en él, mientras que otros protocolos cambian tontamente bits hacia adentro y hacia afuera.

- I2C usa solo dos cables para la señal: uno para el reloj (SCL) y otro para los datos (SDA). Tres si incluye tierra y cuatro si también incluye alimentación. Sin embargo, están disponibles en la placa de circuito impreso (PCB), por lo que no cuentan y permiten conectar (idealmente) 128 dispositivos (prácticamente unos 112, mientras que otras direcciones se reservan para otros fines). Esto ahorra mucho cableado en la placa de circuito impreso.
- Esto significa que el maestro y el esclavo envían datos a través del mismo cable, controlado por el maestro que crea la señal de reloj.
- I2C no utiliza selecciones de esclavos independientes para seleccionar un dispositivo en particular, pero tiene direcciones para identificar el dispositivo esclavo.



### Características:

- Tipo: Síncrono (usa una señal de reloj).

### Características:

- Utiliza dos líneas: SDA (datos) y SCL (reloj).
- Permite múltiples dispositivos en el mismo bus (1 maestro y varios esclavos).
- Cada dispositivo tiene una dirección única.
- Ventajas: Ideal para integrar varios sensores o periféricos en un mismo bus.
- Aplicaciones: Comunicación con pantallas LCD, sensores ambientales, memorias EEPROM, etc.

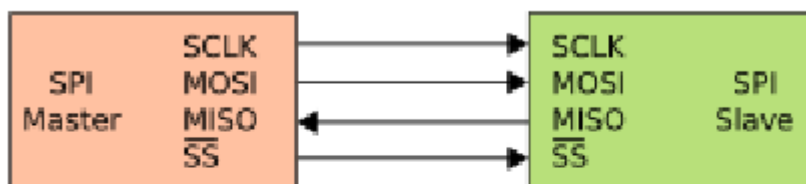


### 3. SPI (Serial Peripheral Interface)

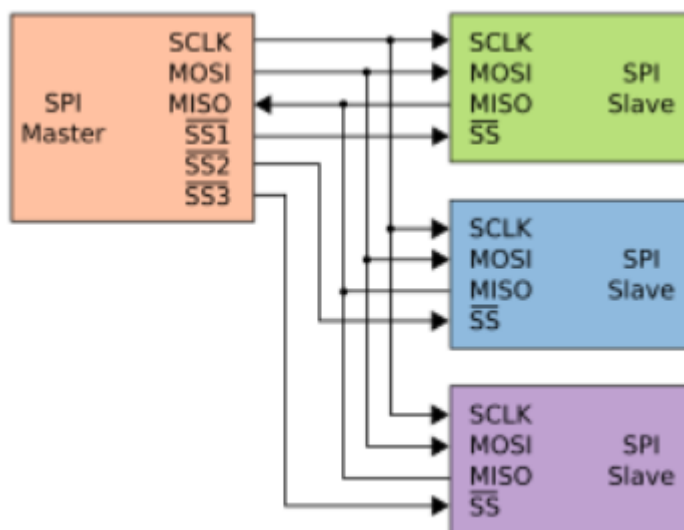
Es un estándar de comunicaciones, usado principalmente para la transferencia de información entre circuitos integrados en equipos electrónicos. El bus de interfaz de periféricos serie o bus SPI es un estándar para controlar casi cualquier dispositivo electrónico digital que acepte un flujo de bits serie regulado por un reloj (comunicación sincrónica).

Incluye una línea de reloj, dato entrante, dato saliente y un pin de *chip select*, que conecta o desconecta la operación del dispositivo con el que uno desea comunicarse. De esta forma, este estándar permite multiplexar las líneas de reloj.

Bus SPI: un maestro y un esclavo.



SPI bus: un maestro y tres esclavos.



Muchos sistemas digitales necesitan una conexión rápida con sus periféricos. Las ventajas de un bus serie es que minimiza el número de conductores, pines y el tamaño del circuito integrado. Esto reduce el coste de fabricar, montar y probar la electrónica. Un bus de periféricos serie es la opción más flexible cuando se tiene tipos diferentes de periféricos serie. El hardware consiste en señales de reloj, data in, data out y chip select para cada circuito integrado que tiene que ser controlado. Casi cualquier dispositivo digital puede ser controlado con esta combinación de señales. Los dispositivos se diferencian en un número predecible de formas. Unos leen el dato cuando el reloj sube, otros cuando el reloj baja. Algunos lo leen en el flanco de subida del reloj y otros en el flanco de bajada. Escribir es casi siempre en la dirección opuesta de la dirección de movimiento del reloj. Algunos dispositivos tienen dos relojes. Uno para capturar o mostrar los datos y el otro para el dispositivo interno.

### Operación

El SPI es un protocolo síncrono. La sincronización y la transmisión de datos se realiza por medio de 4 señales:

- **SCLK (Clock):** Es el pulso que marca la sincronización. Con cada pulso de este reloj, se lee o se envía un bit. También llamado TAKT (en alemán).
- **MOSI (Master Output Slave Input):** Salida de datos del Master y entrada de datos al Esclavo. También llamada SIMO.
- **MISO (Master Input Slave Output):** Salida de datos del Esclavo y entrada al Master. También conocida por SOMI.
- **SS/Select:** Para seleccionar un Esclavo, o para que el Master le diga al Esclavo que se active. También llamada SSTE.

La Cadena de bits es enviada de manera síncrona con los pulsos del reloj, es decir con cada pulso, el Master envía un bit. Para que empiece la transmisión el Master baja la señal SSTE o SS/Select a cero, con esto el Esclavo se activa y empieza la transmisión, con un pulso de reloj al mismo tiempo que el primer bit es leído. Nótese que los pulsos de reloj pueden estar programados de manera que la transmisión del bit se realice en 4 modos diferentes, a esto se llama polaridad y fase de la transmisión:

- 1. Con el flanco de subida sin retraso.
- 2. Con el flanco de subida con retraso.
- 3. Con el flanco de bajada sin retraso.
- 4. Con el flanco de bajada con retraso.

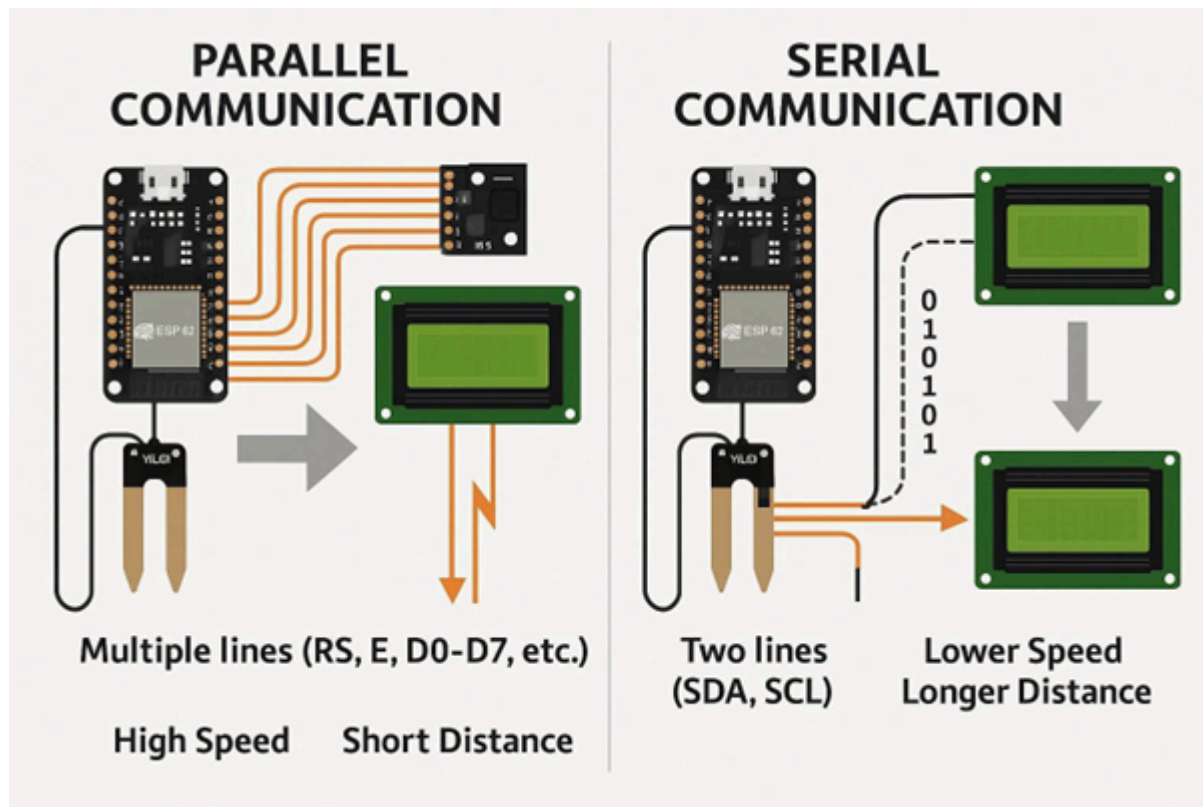
Características:

- Tipo: Síncrono. • Características:
  - Utiliza cuatro líneas: MOSI (Master Out Slave In), MISO (Master In Slave Out), SCK (reloj) y SS (selección de esclavo).
  - Comunicación full dúplex (envía y recibe al mismo tiempo).
  - Más rápida que I2C, pero requiere más pines.
- Ventajas: Alta velocidad de transferencia.
- Aplicaciones: Comunicación con memorias flash, pantallas TFT, sensores de alta velocidad, etc.

Estos protocolos serie permiten una comunicación eficiente y flexible entre dispositivos en sistemas embebidos y proyectos IoT. La elección de uno u otro dependerá de factores como la velocidad requerida, la cantidad de dispositivos conectados y la disponibilidad de pines en el hardware.

## 7. ¿Qué es la comunicación de datos en paralelo? Describa su funcionamiento

La comunicación de datos en paralelo es un método de transmisión de información en el que varios bits se envían simultáneamente utilizando múltiples líneas físicas (una línea por bit). A diferencia de la comunicación serial, que transmite los bits uno detrás del otro a través de una única línea, la comunicación paralela permite que un conjunto completo de bits (por ejemplo, un byte) viaje en un solo ciclo de reloj, lo que puede aumentar la velocidad de transferencia en distancias cortas.



**Funcionamiento:**

- **Estructura:** Cada bit del conjunto de datos tiene su propia línea de transmisión. Por ejemplo, para transmitir un byte (8 bits), se requieren al menos 8 líneas de datos más señales de control adicionales (como reloj y habilitación).
- **Sincronización:** A menudo se utiliza una señal de reloj o control para asegurar que el emisor y el receptor estén sincronizados al momento de enviar y recibir los datos.
- **Transmisión:** El emisor coloca los bits simultáneamente en las líneas de datos, y el receptor los lee todos a la vez en un instante determinado.
- **Aplicación:** Este tipo de comunicación se utiliza comúnmente en sistemas embebidos, especialmente para conectar dispositivos como pantallas LCD,





## 2. SCSI (Small Computer System Interface)

**Descripción:** Estándar paralelo para conectar periféricos, especialmente en entornos profesionales y servidores.

**Características:**

- Mayor velocidad y flexibilidad que PATA/IDE
- Permite conectar hasta 15 dispositivos en una cadena
- Versiones paralelas incluyen SCSI-1, SCSI-2 y Ultra SCSI
- Usado en discos duros, escáneres y unidades de cinta

**Evolución:** Las versiones modernas han migrado a interfaces seriales como SAS.

## 3. IEEE 1284 (Puerto Paralelo Centronics)

**Descripción:** Estándar para comunicación con impresoras y otros periféricos, evolucionado desde la interfaz Centronics original.

**Características:**

- Versiones: SPP (unidireccional), EPP (bidireccional de 2 MB/s) y ECP (con compresión)
- Usa conector DB-25 en la computadora y Centronics de 36 pines en la impresora
- Distancia máxima de 1.8 metros
- Ancho de banda de hasta 3 Mbps

**Estado actual:** Reemplazado por USB en la mayoría de aplicaciones.

## 4. PCI (Peripheral Component Interconnect)

**Descripción:** Bus paralelo para conectar tarjetas de expansión a la placa base.

**Características:**

- Ancho de bus de 32 o 64 bits
- Frecuencias de reloj de 33 MHz o 66 MHz
- Ancho de banda de hasta 533 MB/s (en versión de 64 bits)

- Usado para tarjetas gráficas, de sonido y red

**Evolución:** Las versiones modernas (PCI Express) utilizan comunicación serial.

## 5. IEEE-488 (GPIB/HPIB)

**Descripción:** Bus paralelo desarrollado por HP para instrumentación de laboratorio.

**Características:**

- 8 líneas de datos y 8 líneas de control
- Velocidad de hasta 1 MB/s
- Permite conectar hasta 14 dispositivos
- Usado en equipos de medición y prueba

## Comparativa de Protocolos Paralelos

Protocolo	Año Introducción	Velocidad Máxima	Dispositivos por Canal	Principal Uso
PATA/IDE	1986	133 MB/s	2	Discos duros, unidades ópticas
SCSI	1986	320 MB/s (Ultra-320)	15	Almacenamiento de alto rendimiento
IEEE 1284	1994	3 Mbps	1	Impresoras, escáneres
PCI	1993	533 MB/s	Varios	Tarjetas de expansión
IEEE-488	1965	1 MB/s	14	Instrumentación de laboratorio