Arquitectura y Conectividad

# Cuestionario N°1

**Profesor Ing.** Morales Jorge Elias | https://github.com/JorEl057

**Miembros:**

* Durigutti, Vittorio | GitHub: https://github.com/vittoriodurigutti
* Zalazar, Joaquín | GitHub: https://github.com/breaakerr
* Marquez, José | Github: https://github.com/marquezjose
* Lujan, Luciano | Github: https://github.com/lucianoilujan
* Velez, Nahuel | Github: https://github.com/Lucasmurua19
* Juncos, Lisandro | Github: https://github.com/Lisandro-05
* Garzón, Joaquín | Github: https://github.com/Joacogarzonn
* Guzmán, Maria |Github: https://github.com/lilenguzman01

**Cuestionario** (Link a la respuesta)  
[1. Nombre, describa y grafique las capas OSI.](#_838mmnci6lh9)

[2. ¿Cómo se comunican los dispositivos IoT? Esquematizar y ejemplificar.](#_1i1ynrqq2sj5)

[3. Ejemplifique qué son dispositivos IoT](#_8v5b9jb1c9ih)

[4. ¿Qué tecnologías han hecho posible el IoT? Menciona 5 ejemplos.](#_qj9sw6g4v13b)

[5. ¿Qué es la comunicación de datos en serie? Describa su funcionamiento.](#_geyu5vfbkyjw)

[6. ¿Cuáles son los protocolos serie más usados? Nombre 3 y descríbalos.](#_asj06ksvcrdu)

[7. ¿Qué es la comunicación de datos en paralelo? Describa su funcionamiento.](#_fvkj9m9sxgr9)

[8. ¿Cuáles son los protocolos paralelos más usados? Nombre alguno y descríbalo.](#_lk170t67bxbi)

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

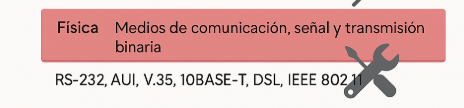
El modelo OSI es fundamental para entender cómo los datos se transfieren entre dispositivos. Sus capas son:

* **Capa Física**: Se encarga de la transmisión de señales eléctricas o electromagnéticas.
* **Capa de Enlace de Datos**: Permite la comunicación entre dispositivos directamente conectados mediante protocolos como MAC y ARP.
* **Capa de Red:** Define cómo se enrutan los datos entre nodos. Ejemplo: IPv4 e IPv6.
* **Capa de Transporte:** Garantiza la transmisión fiable de datos
* **Capa de Sesión**: Maneja las conexiones entre aplicaciones, permitiendo inicio, gestión y finalización de sesiones.
* **Capa de Presentación**: Asegura que los datos sean interpretables, manejando formatos y cifrado.
* **Capa de Aplicación**: Es la más cercana al usuario, donde se ejecutan aplicaciones como navegadores y correos electrónicos.

**Capa Física:**

Es la base del modelo OSI y se encarga de la transmisión de datos en forma de señales eléctricas, ópticas o electromagnéticas a través de medios físicos como cables, fibra óptica o aire

Define especificaciones de hardware como voltajes, niveles de señal y tiempos de transmisión.



**Capa de Enlace de Datos:**

Proporciona una comunicación directa y fiable entre nodos conectados físicamente.

Control de acceso al medio (para evitar colisiones en transmisiones simultáneas).

Detección y corrección de errores.

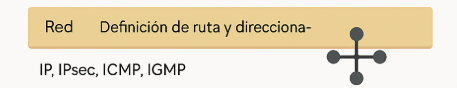


**Capa de Red:**

Gestiona la forma en que los datos se enrutan y entregan entre dispositivos a través de distintas redes.

Asignación de direcciones lógicas a los dispositivos (como direcciones IP).

Enrutamiento de datos a través de múltiples nodos para llegar a su destino final

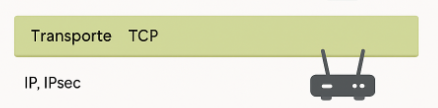


**Capa de Transporte:**

Asegura la transmisión fiable de datos entre dos puntos finales

control de flujo y segmentación de datos en paquetes.

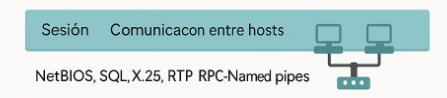
Retransmisión de paquetes en caso de pérdida.



**Capa de Sesión:**

Maneja las sesiones o conexiones entre aplicaciones que intercambian datos.

Establecimiento, mantenimiento y cierre de sesiones.

****

**Capa de Presentación**:

Transforma los datos en un formato comprensible para las aplicaciones.

Compresión de datos para reducir el tamaño de la transmisión.

Encriptación y desencriptación para garantizar la seguridad.



**Capa de Aplicación**

Es la más cercana al usuario y se encarga de proveer servicios que permiten la interacción con la red.

Brinda interfaces para aplicaciones de software

Maneja protocolos de alto nivel para la transferencia de datos y la comunicación.



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

Los dispositivos de la Internet de las Cosas (IoT) utilizan diversas tecnologías de comunicación. La elección correcta depende siempre de las necesidades específicas de la aplicación, del tipo de dispositivo, su uso y el entorno en el que opera.

**Tipos de conectividad: física o inalámbrica**

Los dispositivos IoT se conectan utilizando distintos tipos de redes, tanto inalámbricas como por cable:

**Redes inalámbricas**

* Wi-Fi: Muy común en entornos domésticos y comerciales, ofrece alta velocidad pero con consumo energético elevado. Opera en las bandas de frecuencia de 2,4 GHz y 5 GHz, y la más reciente, la banda de 6 GHz.
* Bluetooth: Ideal para comunicaciones de corto alcance y bajo consumo, usado en dispositivos personales y wearables.
* Z-Wave: Tecnología inalámbrica propietaria de corto alcance y bajo consumo, orientada a la automatización del hogar. Utiliza su propio protocolo de comunicación y opera en bandas sub-GHz, ideal para redes malladas (mesh) de sensores y actuadores.
* LoRaWAN: Excelente para comunicaciones de largo alcance y bajo consumo, ideal para zonas rurales o aplicaciones de monitoreo ambiental.
* Sigfox: Es una red de IoT pensada para tener un bajo consumo y ser independiente de los despliegues de telefonía, usada en sensores remotos donde se transmiten pocos datos.

**Redes celulares**

* 3G, 4G y 5G: Tecnologías celulares tradicionales que permiten conectividad móvil de alta cobertura y velocidad.
* NB-IoT (Narrowband IoT): Diseñada específicamente para dispositivos IoT que requieren poca energía y buena penetración en interiores.
* LTE-M (Long Term Evolution for Machines): Similar al NB-IoT, pero con mayor velocidad y soporte de movilidad, ideal para dispositivos en movimiento.

**Comunicación por cable**

* Ethernet: Proporciona conexión estable y de alta velocidad, usada en entornos industriales o cuando no se requiere movilidad.
* RS-485: Estándar usado en entornos industriales para comunicación en serie entre dispositivos.

**Protocolos de comunicación**

Además de las tecnologías de conectividad, los dispositivos IoT emplean diversos protocolos de transmisión de datos, que definen cómo los dispositivos intercambian información de manera eficiente y segura:

* MQTT (Message Queuing Telemetry Transport): Ligero y eficiente, ideal para IoT con conexiones inestables o limitadas.
* CoAP (Constrained Application Protocol): Similar a HTTP, pero optimizado para dispositivos con recursos limitados.
* AMQP (Advanced Message Queuing Protocol): Protocolo confiable y orientado a mensajes, útil en aplicaciones críticas que requieren entrega garantizada.
* HTTP/HTTPS: Utilizado en dispositivos que requieren comunicación con servidores web, aunque con mayor carga de datos.
* LoRaWAN: Es un protocolo de red que usa la tecnología LoRa, para redes de baja potencia y área amplia, LPWAN (Low Power Wide Area Network) empleado para comunicar y administrar dispositivos LoRa. El protocolo LoRaWAN se compone de gateways y nodos.
* Bluetooth Low Energy (BLE): Protocolo optimizado para dispositivos con batería, como relojes inteligentes o sensores de salud.
* Zigbee y Z-Wave: Muy usados en domótica, permiten la creación de redes de dispositivos interconectados.

**Arquitectura de comunicación IoT**

La comunicación en IoT sigue una arquitectura que involucra varios componentes:

* Dispositivos IoT: Sensores, actuadores y dispositivos inteligentes que capturan datos o ejecutan acciones.
* Pasarelas IoT (Gateways): Traducen y retransmiten información entre redes locales y la nube.
* Servidores en la nube: Gestionan la transmisión y almacenamiento de datos.
* Aplicaciones en la nube: Procesan los datos y los presentan a los usuarios finales.
* Interfaz de usuario: Medio por el cual los usuarios visualizan la información y controlan los dispositivos (apps móviles, web, etc.).

**Modelos de comunicación en IoT**

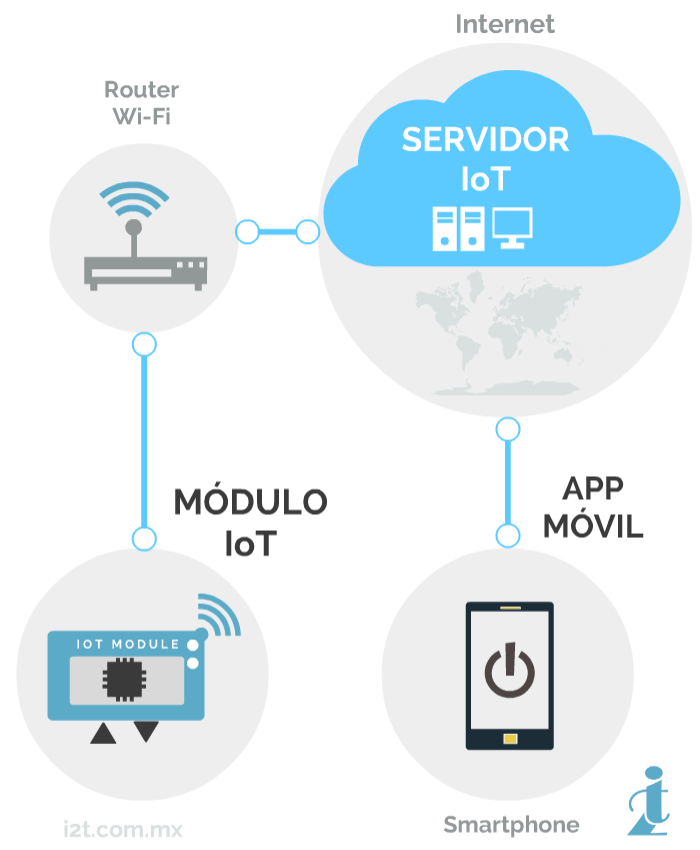
Los dispositivos IoT pueden comunicarse siguiendo distintos modelos:

* Dispositivo a Dispositivo: Comunicación directa entre dispositivos sin intermediarios, usando protocolos como Bluetooth o Zigbee.
* Dispositivo a Nube (Device-to-Cloud): Los dispositivos envían datos directamente a servicios en la nube para su procesamiento y almacenamiento, común en aplicaciones de monitoreo remoto.
* Dispositivo a Puerta de Enlace (Gateway): Los dispositivos se conectan a través de una puerta de enlace que facilita la comunicación con la nube o con otros dispositivos.
* Intercambio de Datos entre Nubes: Los usuarios pueden exportar y analizar datos de objetos inteligentes desde un servicio en la nube, en combinación con datos de otras fuentes, y enviarlos a otros servicios para su agregación y análisis.

**Esquema de comunicación en IoT**

La comunicación entre dispositivos IoT se puede representar en varias etapas:

* Dispositivos y Sensores: Capturan datos del entorno (temperatura, humedad, movimiento, etc.).
* Conectividad: Transmiten esos datos mediante tecnologías como Wi-Fi, Bluetooth, Zigbee, LoRaWAN o redes móviles (4G/5G).
* Puerta de Enlace (Gateway): Recoge los datos y los envía a la nube o a servidores locales.
* Plataforma en la Nube o Servidor: Almacena y procesa los datos recibidos.
* Aplicaciones y Usuarios: Los usuarios acceden a la información procesada a través de aplicaciones móviles o web, pudiendo controlar dispositivos y tomar decisiones.



**Ejemplo: Lámparas inteligentes en un hogar IoT**

* La lámpara inteligente se conecta a la red Wi-Fi del hogar o a un hub central mediante Zigbee o Bluetooth.
* Este hub (o directamente la lámpara, si tiene Wi-Fi) se conecta a la nube del fabricante.
* El usuario interactúa desde su smartphone o asistente de voz para dar órdenes como “apagar todas las luces del dormitorio”.
* La aplicación móvil envía la orden a través de Internet, que es redirigida por los servidores del fabricante hacia la lámpara.
* La lámpara ejecuta la orden y puede enviar una confirmación del estado (por ejemplo, "apagada").

### 3. Ejemplifique qué son los dispositivos IoT

Los dispositivos IoT son objetos físicos que están conectados a Internet y pueden recolectar, enviar y recibir datos. Estos dispositivos suelen tener sensores, actuadores y conectividad de red, permitiendo que interactúen con su entorno y se comuniquen con otros dispositivos o sistemas a través de Internet.

**Algunos ejemplos de estos dispositivos son:  
1 - Termostato inteligente:** Detecta la temperatura ambiente por medio de sensores y se conecta al WiFi para que el usuario pueda controlarlo desde su celular, incluso cuando no está en casa.

  
**2 - Básculas inteligentes:** Mientras miden peso, calculan automáticamente el IMC, grasa corporal y envían los datos a una app.

  
**3 - Sensores de humedad del suelo:** Miden si la planta necesita agua y activan el riego automático.



**4 - Semáforos inteligentes:** Ajustan los tiempos según la cantidad de autos, peatones y datos del tráfico.



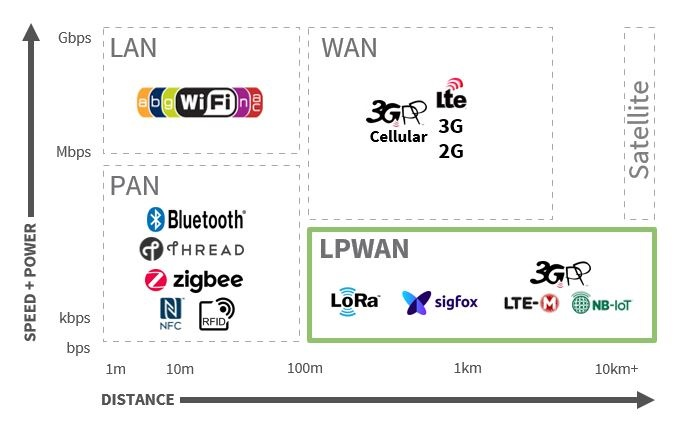
### 4. ¿Qué tecnologías han hecho posible el IoT? Menciona 5 ejemplos.

Resumidamente las 5 tecnologías que hacen posible al IoT son los sensores inteligentes, la conectividad inalámbrica, la nube o cloud computing, microcontroladores/SoCs, big data e inteligencia artificial. Cada una de estas aporta una de las características que hacen al Internet de las Cosas.

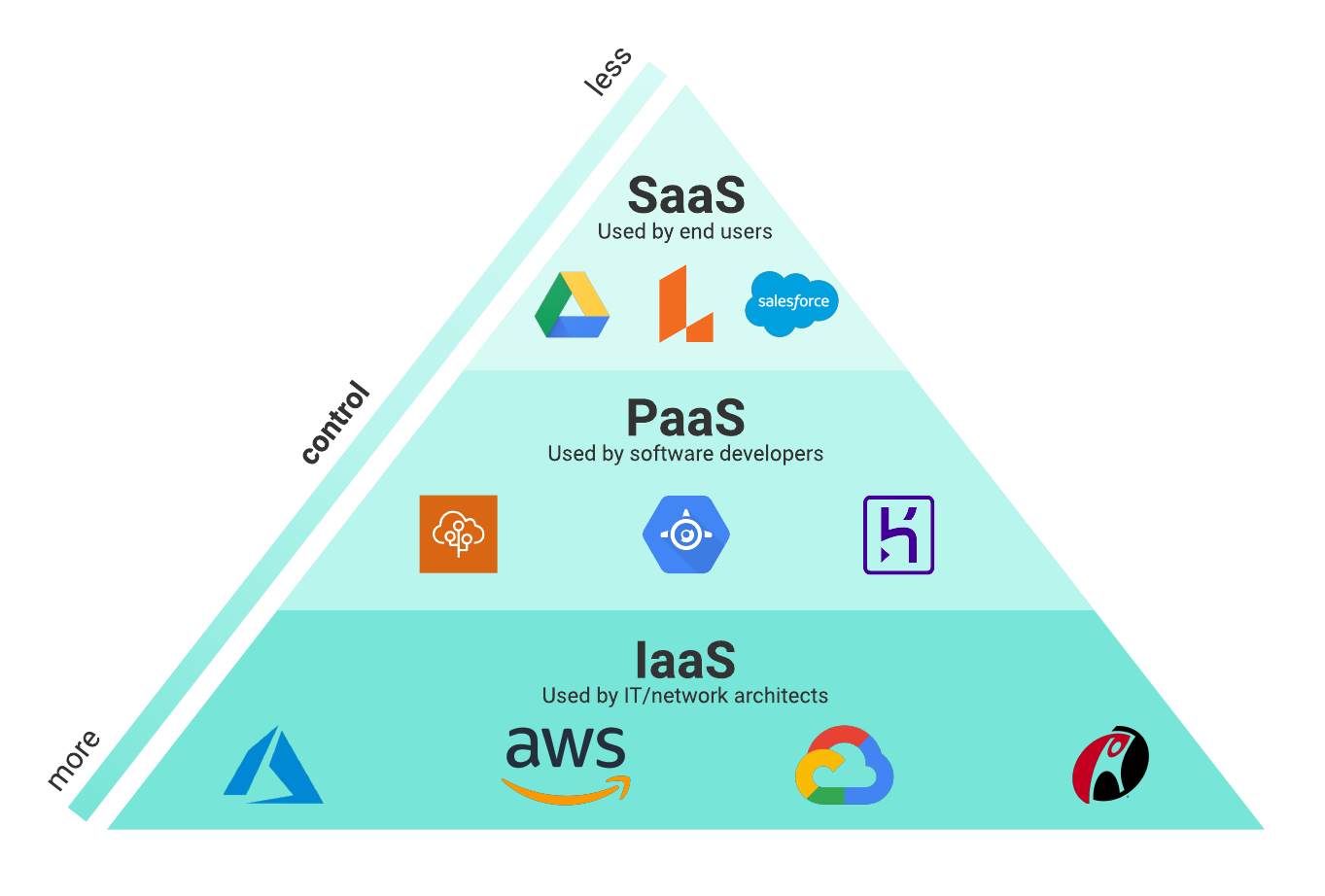
**1. Sensores Inteligente.** Funcionan como los ojos y oídos de los dispositivos IoT. Sin esos, no hay datos que recolectar. Además, a medida que el tiempo avanza, la variedad de sensores aumenta. Y con esto aumentan su precisión, reduce sus costos, se hacen más pequeños y se optimiza su rendimiento.



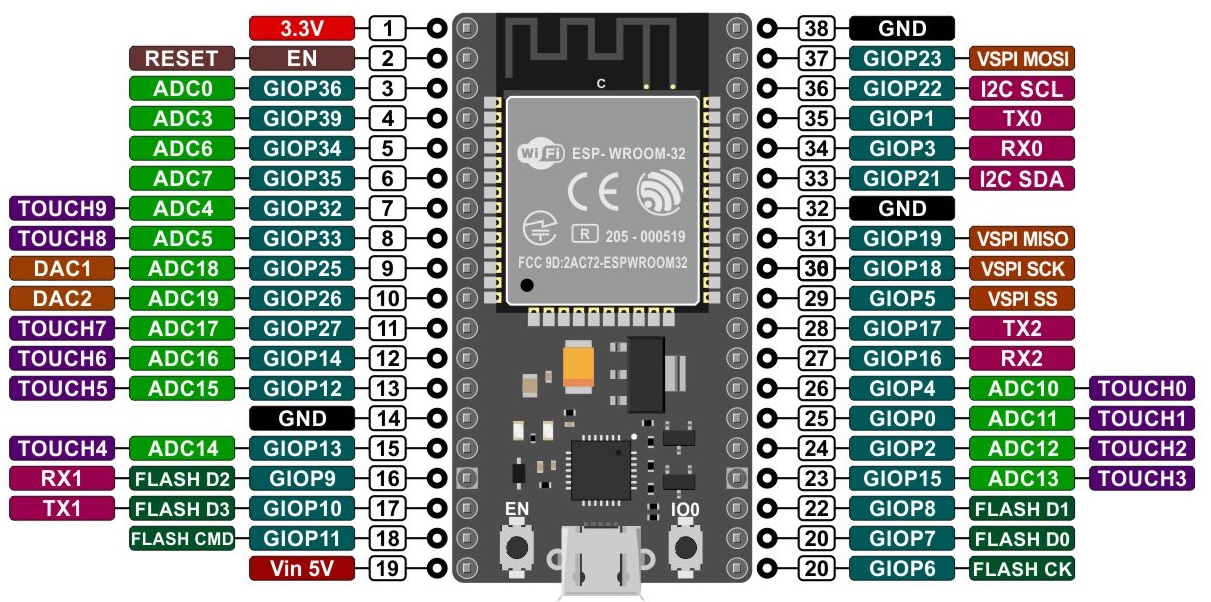
**2. Conectividad Inalámbrica.** Es el habilitante que permite a los dispositivos IoT comunicarse entre sí, el entorno y con internet. Y no es solo la existencia de la comunicación inalámbrica, sino la variedad. La amplia disposición de diferentes protocolos que varían en capacidades, ancho de banda, rendimiento, distancia y seguridad permite encarar dispositivos iot con un medio de comunicación idóneo para el propósito para el cual este planteado. Y el futuro se ve brillante, con el horizonte de tecnologías de comunicación accesibles a latencias próximas al cero, lo que probablemente sea provoque otro salto en el rubro



**3. Cloud Computing.** La existencia de la nube, y la posibilidad de acceder a servicios de estas características según las necesidades y presupuestos hace posible almacenar, gestionar y acceder desde múltiples dispositivos y direcciones a los mismos. Esto saca al IoT de ambientes locales y retira la limitación de las lecturas, aumentando la tasa de datos, y con esto la calidad y presión de los análisis posteriores.



**4. Microcontroladores SoCs (System-on-Chip).** Son el cerebro de los dispositivos. Controlan los sensores y actuadores, manejan la comunicación y ejecutan el software. Cada vez son más compactos, potentes, baratos y consumen menos. Se suma a estos estándares empresariales y de producción que hacen cada vez más simple la modulación y compatibilidad para con sensores, protocolos de comunicación, otros microcontroladores, etc.



**5. Big Data e Inteligencia Artificial (IA).** Almacenar y poder estudiar/analizar todo los datos obtenidos, da lugar luego a la automatización de procesos, análisis predictivo o administración de esos datos con un fin determinado. La reciente llegada de la inteligencia artificial toma esto y lo lleva al siguiente nivel. A medida que el acceso a motores de IA cada vez más compacto y accesibles permitirá, en un futuro cercano, mejorar todos los aspectos relacionados al IoT.

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

La **comunicación de datos en serie** es un método de transmisión en el que los **bits de información se envían uno tras otro** a través de una sola línea de comunicación o canal. Es ampliamente utilizada en sistemas embebidos, electrónica digital e IoT debido a su simplicidad y eficiencia en el uso de pines y cables.

El funcionamiento de la comunicación en serie se basa en el principio de **transmitir un bit de datos a la vez** a través de una línea física (como un cable o pista en un circuito impreso). El transmisor convierte un conjunto de datos (por ejemplo, un byte: 8 bits) en una secuencia de bits y los envía uno a uno, mientras que el receptor los va capturando en el orden correcto para reconstruir el mensaje.

### 

### 

### **Paso a paso del proceso (modo general):**

1. **Inicio de transmisión:**
   * En comunicaciones **asíncronas**, se usa un **bit de inicio** (start bit) para indicar que va a comenzar la transmisión.
   * En comunicaciones **síncronas**, el **reloj (clock)** marca el ritmo y sincroniza ambos extremos.
2. **Transmisión de datos:**
   * Los bits se transmiten uno tras otro a través de la línea de datos (**TX → RX**) en un orden definido (normalmente **LSB primero**).
   * El receptor "lee" el estado de la línea de datos en cada pulso de reloj (síncrona) o en intervalos de tiempo específicos (asíncrona).
3. **Finalización de transmisión:**
   * Puede incluir un **bit de parada** (stop bit) o simplemente cesar la transmisión según el protocolo.
   * En algunos casos se incluyen **bits de paridad o verificación** para detectar errores.

* **S:** Bit de inicio
* **D0-D7:** Bits de datos
* **P:** Bit de paridad (opcional)
* **STOP:** Bit de parada

**A continuación se describen los tres protocolos en serie más utilizados:**

***UART*** *(Universal Asynchronous Receiver/Transmitter)*

* **Tipo**: Asíncrono (no requiere señal de reloj)
* **Uso común**: Comunicación entre microcontroladores, módulos Bluetooth, GPS, etc.
* **Características**:
* Solo requiere dos líneas: TX (transmisión) y RX (recepción)
* La velocidad (baud rate) debe estar configurada igual en ambos dispositivos
* No es adecuado para largas distancias
* No tiene un estándar físico (se implementa dentro del microcontrolador)

***I2C*** *(Inter-Integrated Circuit)*

* **Tipo**: Síncrono (requiere señal de reloj)
* **Uso común**: Sensores, memorias EEPROM, pantallas, RTC, etc.
* **Características**:
* Usa solo dos líneas: SDA (datos) y SCL (reloj)
* Soporta múltiples dispositivos en el mismo bus (comunicación multidispositivo)
* Cada dispositivo tiene una dirección única
* Velocidades típicas: 100 kHz, 400 kHz, hasta 3.4 MHz (High Speed Mode)

***SPI*** *(Serial Peripheral Interface)*

**Tipo**: Síncrono

**Uso común**: Comunicación rápida entre microcontroladores y periféricos como memorias flash, sensores, pantallas, etc.

**Características**:

* Usa al menos 4 líneas: MISO, MOSI, SCK, y SS/CS (chip select)
* Comunicación full-dúplex (puede enviar y recibir al mismo tiempo)
* Más rápido que I2C pero requiere más pines
* No tiene un protocolo de direccionamiento como I2C, por lo que necesita una línea de selección por dispositivo esclavo.

### 6. ¿Cuáles son los protocolos serie más usados? Nombre 3 y descríbalos.

## ***Protocolos serie más utilizados***

### 1. *UART (Universal Asynchronous Receiver/Transmitter)*

* **Tipo**: Comunicación asíncrona (no usa reloj compartido).
* **Uso común**: Comunicación entre microcontroladores, módulos Bluetooth, GPS, Wi-Fi (como ESP8266/ESP32), etc.  
  Conexión: Sólo necesita 2 líneas: TX (transmisión) y RX (recepción).
* **Velocidad**: Configurable (baud rate), típicamente 9600, 115200 bps.
* ***Ventajas***:  
  + Simple de implementar.
  + No requiere señal de reloj adicional.
* ***Desventajas***:  
  + Solo comunicación punto a punto.
  + Menor velocidad que SPI o USB.

### 2. *SPI (Serial Peripheral Interface)*

* **Tipo**: Comunicación sincrónica (usa reloj).
* **Uso común**: Comunicación rápida entre microcontroladores y periféricos como memorias, pantallas, sensores.
* **Conexión**: 4 líneas:  
  + MOSI (Master Out Slave In)
  + MISO (Master In Slave Out)
  + SCK (Serial Clock)
  + SS/CS (Slave Select / Chip Select)
* **Velocidad**: Muy alta, puede superar los 10 Mbps.
* ***Ventajas***:  
  + Alta velocidad.
  + Full-duplex (envío y recepción simultáneos).
* ***Desventajas***:  
  + Necesita más pines.
  + No estándar para múltiples esclavos (requiere múltiples líneas CS).

### 3. *I²C (Inter-Integrated Circuit)*

* **Tipo**: Comunicación sincrónica, de tipo bus.
  + Uso **común**: Conectar múltiples dispositivos (sensores, pantallas, memorias) con solo 2 líneas.
* **Conexión**:  
  + SDA (Serial Data Line)
  + SCL (Serial Clock Line)
* **Velocidad**: Hasta 3.4 Mbps (en modo High-Speed).
* ***Ventajas***:  
  + Solo usa 2 pines, ideal para muchos dispositivos.
  + Cada dispositivo tiene una dirección única.
* ***Desventajas***:  
  + Más lento que SPI.
  + Requiere resistencias pull-up externas.

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

### **¿Qué es la comunicación de datos en paralelo?**

La comunicación de datos en paralelo es un método de transmisión de información en el que varios bits de datos se envían simultáneamente a través de múltiples líneas o canales físicos, uno por cada bit. A diferencia de la comunicación serial (que transmite un bit por vez), en paralelo se envían varios bits al mismo tiempo, lo que puede aumentar significativamente la velocidad de transmisión, aunque a costa de complejidad en el hardware.

### **Componentes clave:**

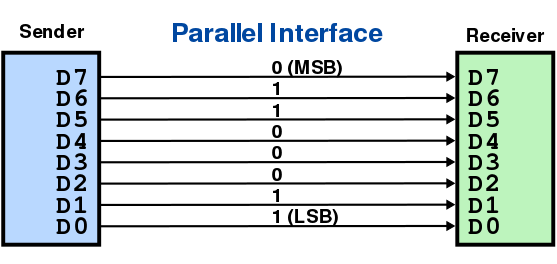
1. Líneas de datos (*Data lines*):  
   Cada línea transporta un bit. Por ejemplo, para transmitir 8 bits (1 byte), se necesitan 8 líneas de datos: D0 a D7.
2. Línea de reloj (*Clock line*) [opcional]:  
   Sincroniza el envío y la lectura de datos entre el emisor y el receptor. Si no se usa, puede haber otra señal de control como strobe.
3. Señales de control:
   * *STROBE*: Indica al receptor que los datos están listos para ser leídos.
   * *ACK* (Acknowledgement): El receptor responde cuando los datos se han recibido correctamente.
   * *BUSY*: Indica que el receptor aún no está listo para recibir nuevos datos.

### **¿Cómo es su funcionamiento?**

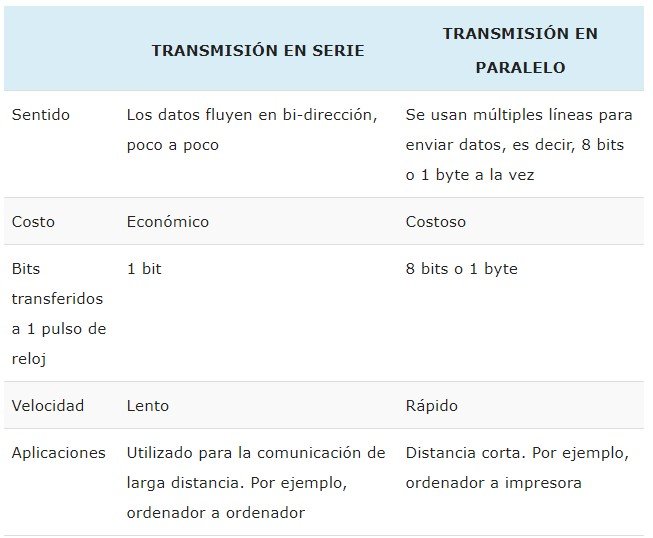
Supongamos que queremos enviar un byte (8 bits) desde un dispositivo A a un dispositivo B:

* En comunicación paralela, se usan 8 líneas de datos, una para cada bit.
* Todos los bits del byte se transmiten al mismo tiempo, sincronizados por una señal de reloj o una señal de control adicional (como strobe o handshake).
* Además de las líneas de datos, puede haber otras líneas auxiliares para control de flujo, confirmación de recepción o sincronización.

[Click para ampliar información.](https://ceptelematica.blogspot.com/2010/10/transmision-serie-y-paralela.html)



## Tabla comparativa SERIE/PARALELO



## Ventajas y desventajas de la comunicación en paralelo:

### 🧠 Ventajas

* ✅ Mayor velocidad de transmisión para distancias cortas.
* ✅ Simple de implementar en entornos donde el espacio físico y el ruido no son un problema (como dentro de un microcontrolador o entre componentes en una placa).

### ⚠️ Desventajas

* ❌ Requiere más cables o pistas físicas (más costoso y complejo en hardware).
* ❌ Problemas de desincronización o interferencias cuando las líneas son largas (efecto skew).
* ❌ No es ideal para distancias largas, como entre nodos IoT distribuidos.

## 📡 Aplicación en Internet de las cosas (IoT):

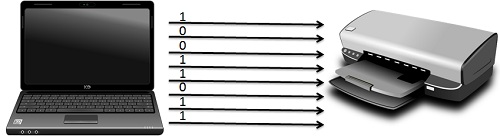
En el contexto de IoT (Internet of things) por sus siglas en inglés, la comunicación paralela no es común entre nodos IoT distribuidos, ya que estos suelen usar comunicación serial (UART, SPI, I2C) o protocolos inalámbricos (Wi-Fi, BLE, Zigbee, LoRa, etc.) para minimizar el cableado y ahorrar energía.

Sin embargo, sí se puede usar internamente en una PCB IoT, por ejemplo:

* Entre un microcontrolador y una pantalla LCD.
* Entre un microcontrolador y una memoria paralela.
* En buses de expansión internos.

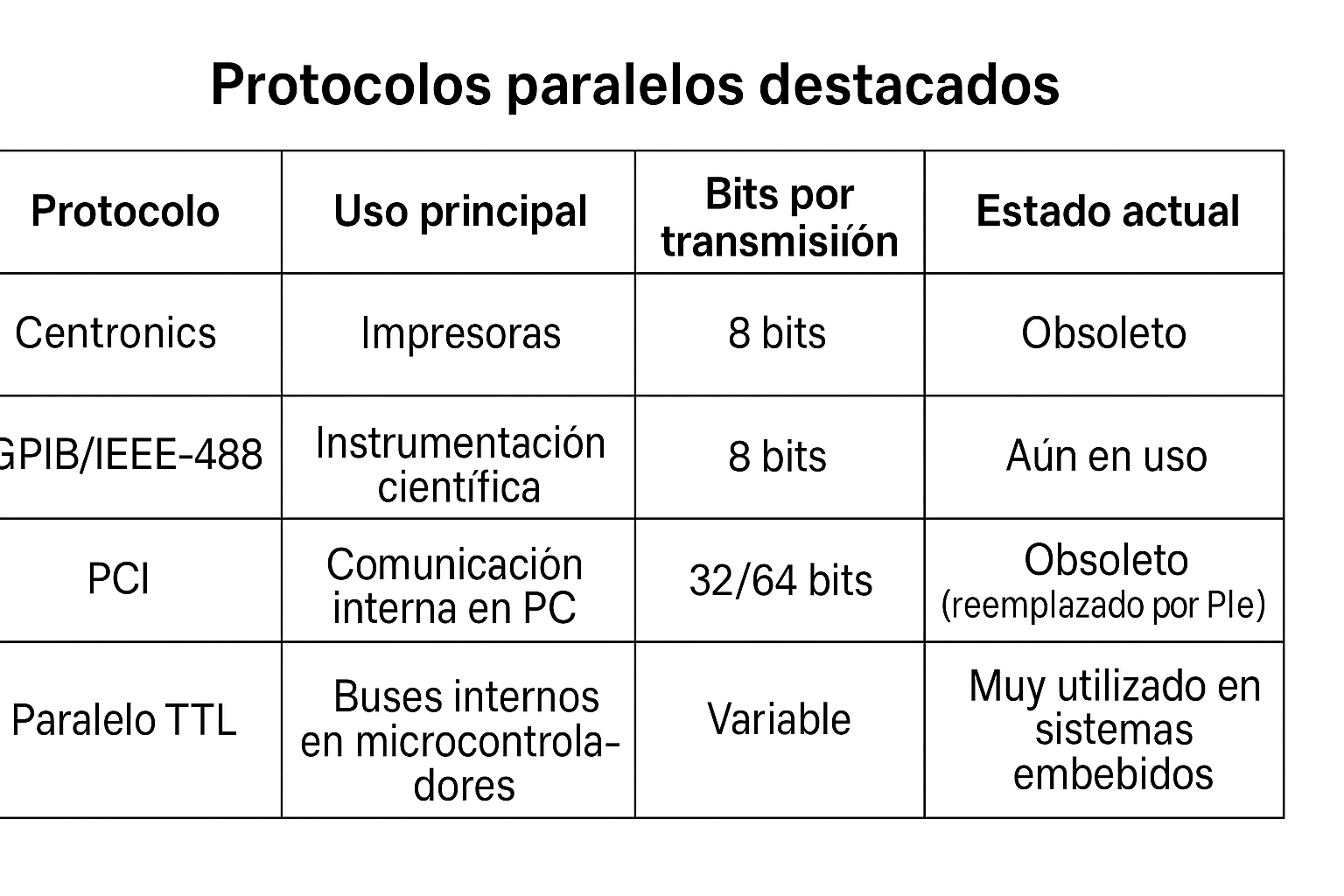
### **¿Dónde se usa esta lógica?**

Este modelo se ve, por ejemplo, en la interfaz de puerto paralelo tradicional (como las impresoras antiguas con conector DB-25), y también en dispositivos embebidos que usan LCDs de 8 bits, memorias RAM antiguas, buses de expansión, etc.



### 8. ¿Cuáles son los protocolos paralelos más usados? Nombre alguno y descríbalo.

La comunicación paralela se caracteriza por transmitir múltiples bits simultáneamente, cada uno por una línea de datos diferente. Esto permite una alta velocidad de transferencia, pero con el costo de mayor complejidad y necesidad de sincronización precisa.



## ***Protocolo Centronics (IEEE 1284)***

### **Descripción:**

Fue uno de los primeros estándares de comunicación paralela, diseñado en los años 70 por la empresa Centronics para impresoras.



Utiliza un conector DB-25 en la computadora y un conector de 36 pines en la impresora.

Transmite datos en bloques de 8 bits en paralelo.

Acompañado de señales de control como STROBE, ACK, BUSY, ERROR y SELECT.

### **Características:**

◍ Originalmente unidireccional: solo enviaba datos desde la computadora a la impresora.

### ***Posteriormente evolucionó a un estándar bidireccional (IEEE 1284) con cinco modos:***

◍ Compatible (modo Centronics)

◍ Nibble (lectura desde la impresora)

◍ Byte (bidireccionalidad)

◍ ECP (Extended Capability Port)

◍ EPP (Enhanced Parallel Port)

### **Uso:**

->Conexión de impresoras paralelas (impresoras matriciales, láser antiguas).

->Fue reemplazado por USB y otras tecnologías más modernas.