

Arquitectura y componentes de IoT

Breve descripción:

El Internet de las cosas propone una tendencia de conexión a Internet con más cosas u objetos que con personas, se refiere a la red colectiva de dispositivos conectados y a la tecnología que facilita la comunicación entre los dispositivos y la nube, así como entre los propios dispositivos. Las tecnologías como IoT se utilizan en los proyectos de ciudades inteligentes.

Julio 2024

Tabla de contenido

Introducción	4
1. ¿Qué es IoT?	6
1.1. Origen e importancia	6
1.2. Arquitectura IoT	7
1.3. Componentes IoT	9
2. Raspberry Pi y Arduino	13
3. Conectividad IoT	19
3.1. Protocolos de conexión	20
3.2. Protocolo de comunicación MQTT	24
3.3. Computación en la nube	26
3.4. Pasarelas de cloud para IoT	27
3.5. Adquisición de datos en la nube	30
4. Protocolos de capa de aplicación	33
4.1. HTTP	34
4.2. WEBSOCKET	35
4.3. JSON	36
5. Sensores para IoT	38
6. Almacenamiento para IoT	39

6.1. Tipos.....	39
6.2. Tecnologías iscsi, fiber channel, sas y arreglos raid.....	40
7. Puertos de comunicación	42
7.1. Analógicos y digitales	42
7.2. Programación de bloques	43
8. Criterios de configuración, interconexión y administración para IoT.....	45
Síntesis	46
Material complementario.....	47
Glosario	49
Referencias bibliográficas	51
Créditos	52

Introducción

Estimado aprendiz, bienvenido a este componente formativo llamado Arquitectura y componentes de IoT, revise el siguiente video de introducción a la temática y alcance propios del componente.

Video 1. Arquitectura y componentes de IoT



[Enlace de reproducción del video](#)

Síntesis del video: Arquitectura y componentes de IoT

Los dispositivos IoT prometen crear objetos más inteligentes y personificar nuestra vida haciéndola más eficiente, este conjunto de técnicas puede aportar

inteligencia a sectores, como la salud, el transporte, la agricultura, la educación, la seguridad ciudadana y la fabricación.

En el desarrollo de este escenario formativo se abordarán contenidos relacionados con la arquitectura y los componentes requeridos en el Internet de las cosas. Las principales características se basan en la automatización de procesos y el uso de los datos aplicando tecnología de bajos recursos por medio de sensores o dispositivos electrónicos.

Con el estudio realizado de analítica de datos, desde la representación personal, el IoT contribuye a una mayor comodidad, personalización y eficacia. Se espera que la información presentada en este componente formativo resulte útil para quienes estén interesados en acercarse a proyectos de IoT.

1. ¿Qué es IoT?

El internet de las cosas hace referencia a la interconexión de los dispositivos y objetos con otros mediante redes como internet, empleando sensores y actuadores integrados y variedad de elementos electrónicos. La interconexión de estos dispositivos con microprocesadores y la conexión de la red, permite obtener, tratar y transferir datos en forma automática; todo objeto en el mundo puede ser una fuente de datos.

A grandes rasgos, podemos decir que el internet de las cosas se conforma por tres sectores: conectividad, hardware y servicios, y el sector de servicios es un campo fértil por sembrar: desarrollo web en la nube, ingeniería de datos, machine learning, inteligencia artificial, big data.

1.1. Origen e importancia

Entre los años 1991 y 1999, Mark Weiser y Naoto Fukusawa se centraron en el área del diseño de dispositivos que tengan una sensibilidad a la naturaleza humana, y estos dispositivos se volverán una parte integral de los seres humanos ejecutando algunos patrones de uso. Kevin Ashton, cofundador de AUTO ID center del MIT en 1999 en una presentación de frecuencias de radio (RFID) llamó a su presentación “Internet de las cosas”, pero el verdadero Internet de las cosas sucedería cuando se conecten a internet más dispositivos que personas y esto sucedió en el 2009 con el nacimiento de los celulares.

En estos términos, para el 2022 se espera que estén conectados 18 mil millones de dispositivos, no solo celulares, tablets, pc’s, sino también dispositivos que nunca

nos imaginaríamos, tales como refrigeradores, cepillos, autos, zapatillas, collares de mascotas, iluminación, cerraduras, ropa, etc.

El internet de las cosas tiene un millar de aplicaciones, algunas de ellas son:

Fuera del hogar. Automóviles en red, prendas tecnológicas y monitores de salud en la ropa, aplicaciones para ciclistas para controlar su actividad física. Gafas inteligentes.

Ropa tecnológica (wearables). Dispositivos médicos, productos de salud, marcapasos y bombas de insulina. Estos wearables monitorizan la geolocalización, así como la frecuencia cardiaca, el pulso, calorías, patrones del sueño, etc.

Transporte. Supervisión de vehículos conectados, sensores de proximidad, parqueo de vehículos, vehículos autónomos, control remoto de peajes.

Contadores inteligentes. Dispositivos electrónicos de medición de energía, calefacción, climatización, agua.

Hogar inteligente. Termostatos domésticos, control de televisores, detectores de humo, cámaras de seguridad, cafeteras y otros electrodomésticos.

1.2. Arquitectura IoT

Según un artículo en Hindawi Journal of Electrical and Computer Engineering, no existe una arquitectura acordada para el Internet de las Cosas. Una arquitectura simple puede estar compuesta de tres o cinco capas.

Conozcamos la arquitectura compuesta por tres capas:

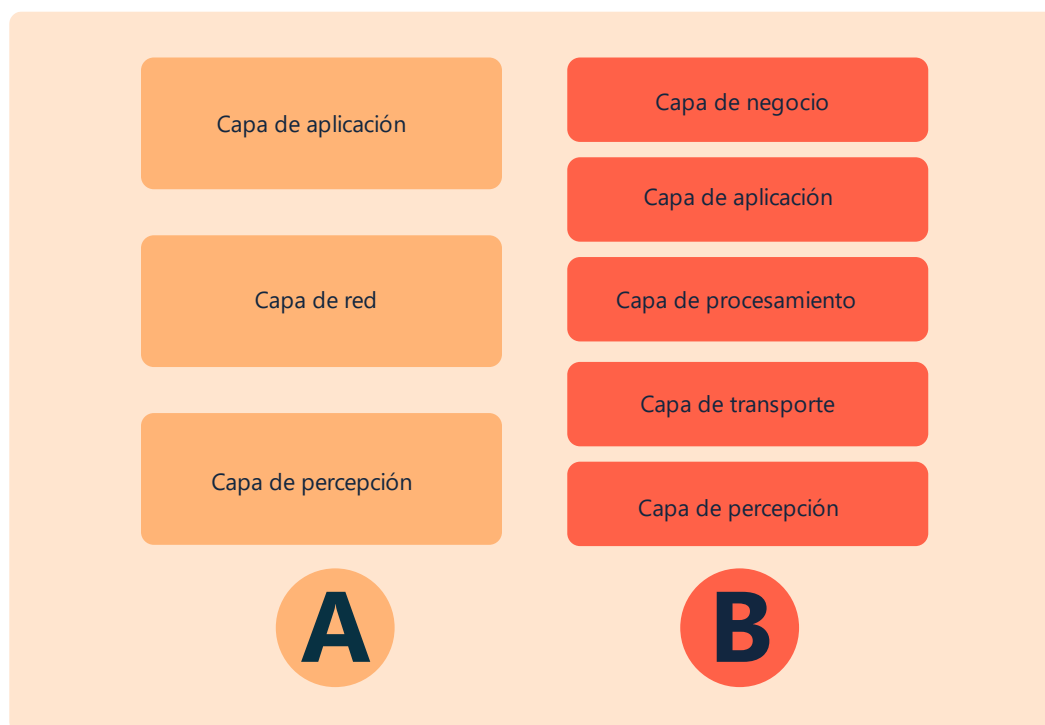
- **Capa de percepción.** Es la capa física, la cual tiene sensores para detectar y recopilar información sobre el medio ambiente.
- **Capa de red.** Tiene la función de conectarse a otras cosas inteligentes, dispositivos de red y servidores, se transmiten y procesan datos de los sensores.
- **Capa de aplicación.** Es responsable de proporcionar servicios de aplicación al usuario, por ejemplo: hogares inteligentes, ciudades inteligentes, salud inteligente, etc.

En la arquitectura de cinco capas, encontramos adicionalmente:

- **Capa de transporte.** Encargada de transferir los datos de sensores desde la capa de percepción a la capa de procesamiento y viceversa a través de redes tales como Wireless, 3G, LAN, Bluetooth, RFID y NFC.
- **Capa de procesamiento o capa middleware.** Almacena, analiza y procesa grandes cantidades de datos que provienen de la capa de transporte. Puede administrar y proporcionar un conjunto diverso de servicios a las capas inferiores, aquí encontramos tecnologías como bases de datos, computación en la nube y procesamiento de big data.

En la siguiente figura se presenta una comparación entre cada una de estas arquitecturas:

Figura 1. Capas de arquitectura.



1.3. Componentes IoT

Un sistema IoT no se puede llevar a cabo o es imposible sin el uso de sensores para recoger datos del medio ambiente, estos son pequeños, de bajo costo y tienen un bajo consumo de energía.

La información recogida por los sensores es convertida a información digital para así poder tratarla, almacenarla y enviarla a otros dispositivos.

Los sensores son dispositivos electrónicos capaces de medir magnitudes físicas o químicas y transformarlas en señales eléctricas y se clasifican en:

- **Basadas en aplicaciones.**

Los sensores son seleccionados de acuerdo con el requerimiento a implementar, tales como procesos de control industrial, medidas y automatización, automóviles, electrónica de consumo, productos médicos.

- **Basadas en tipo de señal de salida.**

Sensores digitales: la salida del sensor es binaria o digital, la cual puede ser procesada directamente a través de controlador o procesador.

Sensores análogos: la señal de salida es continua, se requiere un conversor análogo digital para leer el sensor.

- **Basadas en tipo de sensores.**

Acelerómetros: usados en sistemas dinámicos.

Biosensores: basados en tecnología electromecánica, usados como dispositivos médicos, prueba de agua, prueba de aire, pruebas de alimentos.

Sensores de imágenes: usados ampliamente en vigilancia, biometría y gestión de tráfico.

- **Basadas en propiedades.**

Sensores de temperatura: termocuplas, termistores, detectores de temperatura.

Sensores de flujo: masa térmica, presión diferencial, campo electromagnético, desplazamiento de posición.

Sensores de presión: fibra óptica, manómetros.

Sensores de nivel: frecuencia de radio ultrasónica, radar, desplazamiento térmico.

Sensores de proximidad y desplazamiento: fotoeléctricos, ultrasónicos, magnéticos.

Biosensores: electroquímicos, sensores resonantes.

Aceleración: acelerómetros, giroscopios.

Y los actuadores activan un determinado proceso utilizando señales eléctricas. Estos reciben una señal eléctrica y la transforman en movimiento en el ambiente Internet de las cosas; los actuadores se usan siempre que sea necesario encender o apagar otro dispositivo, aplicando una fuerza.

Los actuadores tienen dos clasificaciones:

1. Según la fuente de energía: se requieren para generar movimiento.

- **Hidráulicos.** Utilizan la presión del líquido para generar un movimiento mecánico.
- **Neumáticos.** Utilizan aire comprimido a alta presión para permitir el funcionamiento mecánico.
- **Eléctricos.** Utilizan fuentes de energía externa como baterías u otros tipos de energía eléctrica para generar movimiento.

2. Según su patrón de construcción y papel que desempeñan.

- **Actuadores lineales.** Se utilizan para permitir el movimiento de objetos o elementos en línea recta, se emplean principalmente para movimientos como empujar, levantar, tirar, bloquear, expulsar, sujetar o descender.
- **Motores.** Permiten movimientos rotativos precisos de los componentes del dispositivo u objetos completos.
- **Relés.** Se basan en electroimanes para manejar mecánicamente un interruptor. No necesitan mucha energía para operar la bobina del relé. Se usan en lámparas, calentadores, vehículos inteligentes, entre otros.
- **Solenoides.** Son un electroimán usado en electrodomésticos, equipos de oficina, automóviles, y también funcionan como controladores en los sistemas de fugas de agua y gas en lota.

2. Raspberry Pi y Arduino

A continuación, conoceremos qué son el Raspberry Pi y el Arduino, elementos importantes a la hora de hablar de Internet de las Cosas.

RASPBERRY PI



El Raspberry Pi es un minicomputador de bajo costo que puede ser conectado a un monitor o televisión.

Los componentes del Raspberry Pi son:

Fuente de potencia. Puede ser conectado con una conexión capaz de suplir 400 mA en 5V.

Pantalla. La pantalla es un monitor o televisor con entrada HDMI o DVI.

Teclado y mouse. Periféricos USB de baja potencia con menos de 0.1^a.

Tarjeta de memoria SD. La tarjeta SD usada para Raspberry Pi debe ser al menos clase 4 con mínimo 8 Gb de capacidad de almacenamiento, la tarjeta SD

necesita ser instalada sobre un sistema operativo desde otro computador. El proceso de instalación puede ser realizada sobre Windows, Mac, Linux, NOOBS y Raspbian.

Ventajas de Raspberry Pi

- Uso sencillo
- Tamaño muy pequeño
- Precio bajo
- Tienen pines de entrada/salida, por lo que también se pueden conectar a otros componentes electrónicos, como sensores, LEDs, etc.

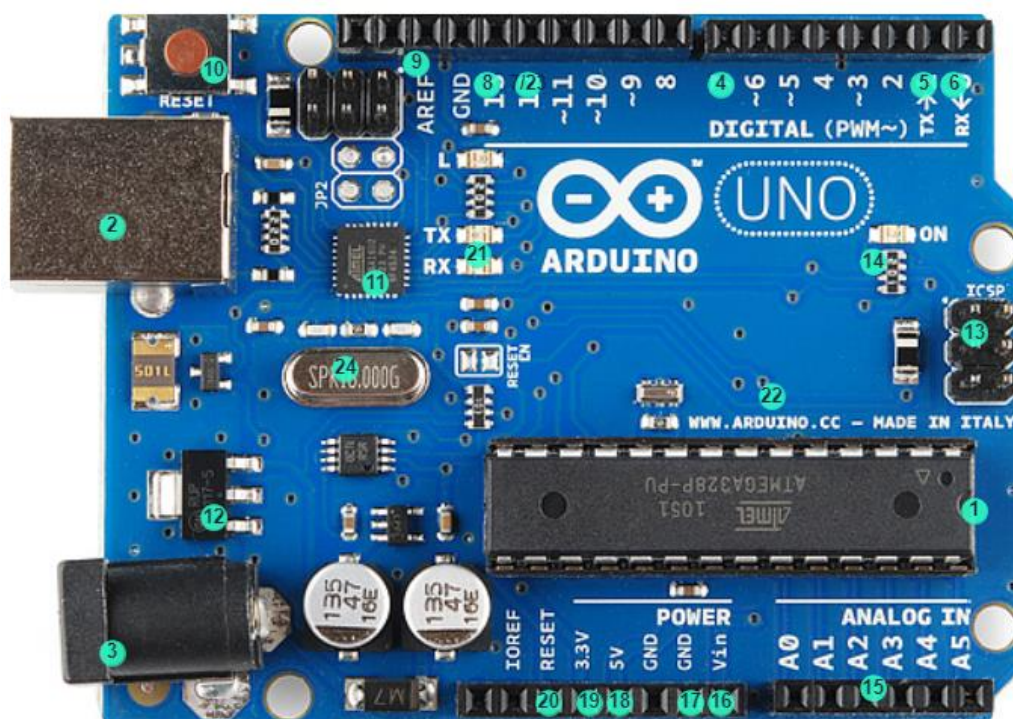
ARDUINO



Inventado en el Ivrea Interaction Design Institute, diseñado para la creación rápida de prototipos dirigido a aficionados, muy pronto su uso creció exponencialmente y se adapta a las últimas tendencias en el mercado, desde una placa de 8 bits hasta productos IoT, dispositivos portátiles y entornos embebidos.

El software Arduino es fácil de usar y se puede operar en plataformas Mac, Linux y Windows.

A continuación, identifiquemos cada uno de sus componentes:



1. **ATmega 328.** Microcontrolador. El corazón del Arduino, el procesador.
2. **Puerto USB.** Se utiliza para la alimentación de la interfaz Uno Genuino, la carga de nuestros programas, y para la comunicación con nuestra placa (a través de serie. PrintIn (), etc.) El regulador de tensión ubicado a su lado,

convertirá la tensión que le llega a través de este puerto USB y hará que trabaje a 5V.

3. **Conector de alimentación.** Esta es la forma de alimentar nuestra placa cuando no está conectada a un puerto USB para suministrar corriente eléctrica. Puede aceptar tensiones entre 7-12V.
4. **Los pines digitales.** Utilizaremos estos pines con instrucciones como `DigitalRead ()`, `DigitalWrite ()`. `AnalogRead ()` y `AnalogWrite ()` funcionarán únicamente en los pines con el símbolo PWM.
5. **Serial IN (TX).** Los puertos serie están físicamente unidos a distintos pines de la placa Arduino. Lógicamente, mientras usamos los puertos de serie, no podemos usar como entradas o salidas digitales los pines asociados con el puerto de serie en uso. Pines asociados con el puerto de serie como entrada y salida digital (TX es el que Transmite y RX es el que recibe).
6. **Serial IN (RX).** Los puertos serie están físicamente unidos a distintos pines de la placa Arduino. Lógicamente, mientras usamos los puertos de serie no podemos usar como entradas o salidas digitales los pines asociados con el puerto de serie en uso. Pines asociados con el puerto de serie como entrada y salida digital (TX es el que Transmite y RX es el que recibe).
7. **y 23. Pin 13 + L (on board led).** El único actuador incorporado a la placa. Además de ser un objetivo práctico para nuestra primera práctica de encender y apagar LED, este LED es muy útil para la depuración (debugging).
8. **GND.** Proporciona masa, tierra, negativo a los circuitos.
9. **AREF.** AnalogReference input V- Tensión a 5V, proporciona diferencia de potencial. Si se le aplica voltaje externo debe ser entre 0 y 5V solamente.
10. **Botón Reset.** Restablece el microcontrolador. ATmega.

- 11. Chip.** Chip de comunicación serie.
- 12. Regulador de tensión.** Se debe tener en cuenta que para que trabaje a 5V la placa deberá recibir unos 6,5V – 7V, pero todo lo que esté por encima de este valor se desperdiciará (es decir, sobrecalentará la placa de Arduino y mayor calor que tendrá que disipar el regulador). En cualquier caso, no está recomendado aplicar al regulador más de 12V y a los 20V se dañará.
- 13. ICSP.** In-Circuit Serial Programming.
- 14. Led de encendido.** Indica que la placa está recibiendo alimentación.
- 15. Pines analógicos.** Utilizaremos estos pines con instrucciones como Analogread.
- 16. Vin.** Voltaje Input. Deberá llegarle una tensión regulada y estable (ya que no pasa por el regulador -nº12-) de 5V.
- 17. GND.** Proporciona masa, tierra, negativo a nuestros circuitos.
- 18. 5V.** Voltaje Input.
- 19. 3,3V.** Voltaje Input.
- 20. Reset pin.** Restablece el microcontrolador ATmega. A su izquierda está el conector IOREF: Digital Reference input V, y a continuación un pin reservado para futuras finalidades.
- 21. TX y RX LED.** Estos LEDs indican la comunicación entre la placa y el ordenador. Estos leds parpadearán rápidamente durante la carga de nuestros programas, así como durante la comunicación serie. Útil para la depuración debugging.
- 22. Casa fabricante original.** En el siguiente enlace puedes comprobar fácilmente si tu placa se trata de una imitación o el original: <https://www.arduino.cc/>
- 23.** (Consultar ítem 7).

24. Reloj / Crystal 16 Mhz oscilador. Se usa como reloj externo en el montaje del Arduino.

Ventajas de Arduino

- **Costo.** Los tableros de Arduino son menos costosos comparados con otros dispositivos como microcontroladores.
- **Plataforma.** El software de Arduino (IDE) es compatible con la mayoría de sistemas operativos como Macintosh OSX, Windows, Linux.
- **Fácil de usar.** El software de Arduino (IDE) es fácil de usar, además tiene flexibilidad para programadores expertos.
- **Código abierto.** El Arduino es un software de código abierto que puede ser programado con lenguajes C, C++, AVR-C.

3. Conectividad IoT

El IoT, o internet de las cosas, consiste en conectar a internet cualquier dispositivo teniendo en cuenta tres aspectos fundamentales: captura de datos (telemetría, geoposicionamiento), tratamiento de datos (Business Intelligence, Machine Learning) y transformación del negocio.

Los dispositivos IoT generan grandes cantidades de datos digitales que contienen información de los usuarios, y las máquinas de análisis del Big Data y las plataformas en la nube pueden acceder a esos datos para generar información relevante y útil para los consumidores.

Para lo anterior se necesita una arquitectura global IoT que se compone de los siguientes componentes, trabajando conjuntamente:

- **Cosas.** IoT inicia con sensores conectados a las cosas para recolectar los datos y actuadores para realizar acciones correspondientes a los comandos recibidos desde la nube.
- **Gateway.** Es usado para filtrado, preprocesamiento y comunicación con la nube y viceversa.
- **Cloud Gateway.** Se usa para transmitir datos entre otros gateway y los servidores centrales IoT.
- **Data Lake.** Repositorio de almacenamiento que contiene datos en bruto.
- **Big data Warehouse.** Se usa para recopilar datos valiosos.
- **Aplicaciones de control.** Se emplea para enviar datos a los actuadores.
- **Aplicaciones de usuario.** Habilita a los usuarios el monitoreo de los datos y toma de decisiones sobre el control de dispositivos conectados.

3.1. Protocolos de conexión

A continuación, se estudiarán los diferentes protocolos de conexión.

Modelo OSI

El modelo de interconexión de sistemas abiertos (OSI, por sus siglas en inglés), es un modelo de red creado por la ISO (Organización Internacional de Normalización) y se compone de siete capas:

1. **Física.** Envía la información de la capa de enlace de datos. Como los datos no pueden viajar como números o código binario, estos son codificados en señales (eléctricas, lumínicas o de ondas) dependiendo del tipo de medio de comunicación que conecte a nuestro dispositivo emisor o receptor. El trabajo de codificación lo realiza la tarjeta de red o NIC. Los datos pueden viajar por cables de cobre, fibra óptica o por el espacio como un tipo de onda, usando redes inalámbricas como el wifi.
2. **Enlace de datos.** Toma toda la información y la traduce en información binaria, además, se divide en dos subcapas:
 - LLC (Logic Link Control), MAC (Media Access Control).
 - LLC comunica el software de la red con el hardware de la misma por medio de drivers de tarjeta de red, una vez se realiza la comunicación se envían los datos a la MAC (se agrega un direccionamiento físico), la IP indica en qué red en el mundo se encuentra el dispositivo y la dirección MAC dice cuál es el dispositivo de la red.
3. **Red.** Una de las capas más importantes es la capa 3, pues determina la mejor ruta para que un mensaje llegue a otro dispositivo. Es una capa de direccionamiento y enrutamiento que no toca el contenido de los datos, sino

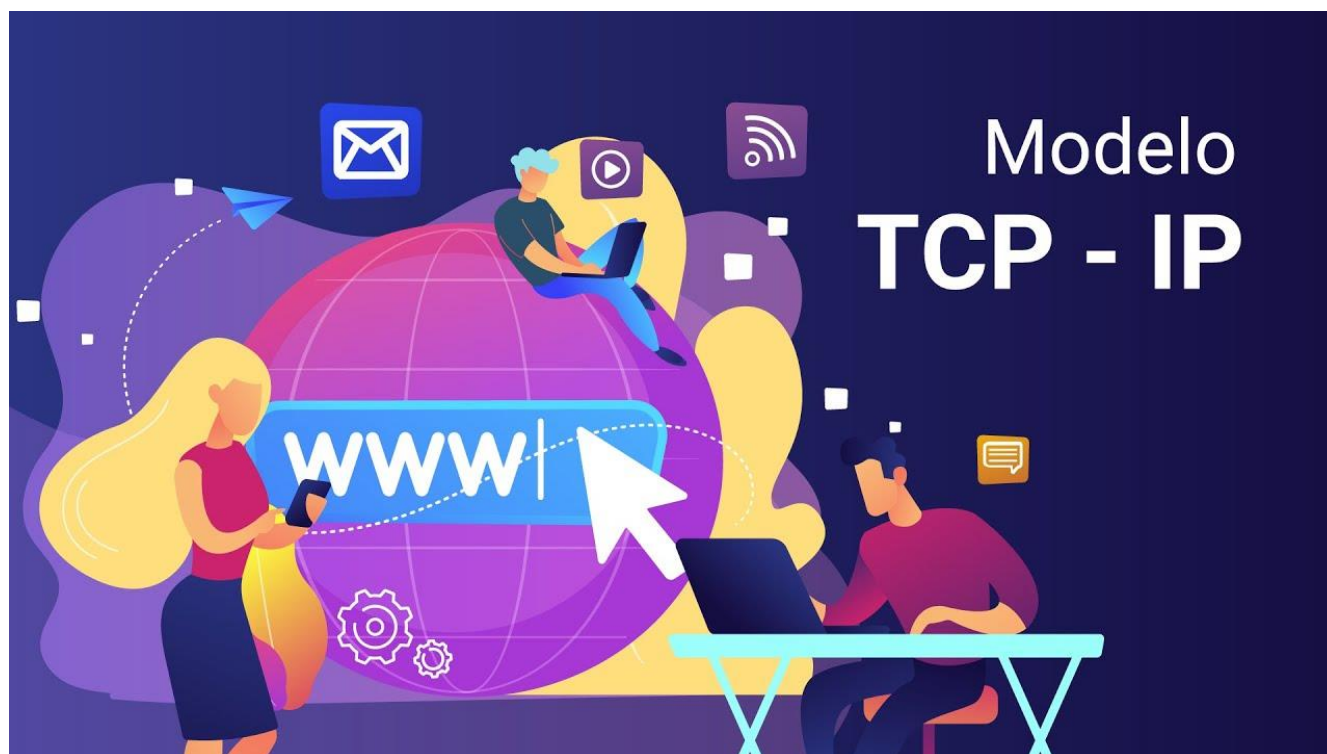
que les agrega a dicho contenido direcciones IP origen e IP destino; cada dispositivo en el mundo tiene una dirección. En esta capa también se hace enrutamiento, para ello se usan los routers.

- 4. Transporte.** Segmenta los datos en paquete, se hace en el dispositivo emisor y el dispositivo receptor hace todo lo contrario; a este al proceso se le conoce como segmentación y reensamblaje. Los datos son particionados, puesto que las redes tienen anchos de banda limitados.
- 5. Sesión.** Maneja la conversación entre dos equipos (hosts) y establece el camino para las aplicaciones que se van a comunicar entre el dispositivo emisor y el receptor; el diálogo de establecer, mantener y cerrar se llama sesión.
- 6. Presentación.** Se encarga del formato que deseamos analizar e implementa mecanismos de codificación como ASCII o Unicode para representar los datos a un formato que las computadoras entienden, se traduce lenguaje de alto nivel a lenguaje binario. Por ejemplo: si queremos conocer una imagen, esta capa se encarga de dicha acción.
- 7. Aplicación.** Esta es la capa que los usuarios ven y sirve de interfaz entre las aplicaciones donde generamos los datos y el resto de la red. Los datos a transmitir son generados por una aplicación. No nos referimos a las aplicaciones mismas, sino a los procesos que funcionan a nivel de software en nuestro aplicativo; por ejemplo: enviar un mensaje con WhatsApp, ver un video, enviar un correo, Twitter, navegador web.

Modelo TCP/IP

El siguiente video presenta el Modelo TCP/IP, a través de un ejemplo.

Video 2. Modelo TCP/IP



[Enlace de reproducción del video](#)

Síntesis del video: Modelo TCP - IP

Este video presenta a través de un ejemplo sencillo el Modelo TCP – IP. Del lado izquierdo tenemos un servidor web y del lado derecho tenemos un equipo con un navegador web.

El servidor necesita enviar una página web al cliente, la página web es procesada por la capa de aplicación que le da el formato a la página y la codifica en

código entendible por el computador, que es el código binario, aquí se le puede agregar un algoritmo de encriptación para enviarla segura por la red.

Como aquí tenemos una comunicación web, en este caso se hace uso del protocolo HTTP que agrega un encabezado a la página web al enviar un HTTP OK. Una vez que la página web es formateada y codificada por la capa de aplicación, la capa de transporte obtiene los datos de la página web y la fragmentada para que pueda ser enviada por el medio de comunicación, en este caso entra en acción el protocolo TCP.

Aquí se agrega información adicional a la página encabezada, tal como número de puerto origen y puerto destino de la comunicación. Una vez encapsulada la página en la capa de transporte, en la capa de red se agrega el encabezado adicional como dirección IP origen y dirección IP destino, información de direccionamiento lógico.

En la capa de enlace de datos no se toca el contenido de la página, se agrega un encabezado al inicio y un tráiler al final de ésta. En el encabezado la información de direccionamiento físico, MAC origen y MAC destino de los dispositivos que se comunican, en este caso MAC del servidor y MAC de cliente web.

El tráiler sirve para que el cliente web compruebe que la página no sufrió daños en el camino. Por último, la capa física ya no agrega encabezados, aquí solo se codifica, es decir se transforma en alguna señal que pueda viajar por medio de comunicación. Por ejemplo, señales eléctricas, ondas electromagnéticas, señales de luz, etc.

Finalmente, la página web viaja por el medio de comunicación hasta llegar al cliente web, cuando llegue al cliente web la capa física codifica la página y

transforma las señales eléctricas, ondas o luz, en binario. En la capa de enlace se desencapsula la página web, se le quita el encabezado y el tráiler de capa de enlace. En la capa de red se quita el encabezado de esta capa, cuando llegue a la capa de transporte se quita el encabezado de la capa de transporte. Si llegan varios segmentos se agrupan para formar un solo archivo.

En la capa de aplicación se quita el encabezado HTTP, se procesa el formato de información y traduce los 0 y 1 en lenguaje entendible por el ser humano.

Los mensajes encapsulados en la capa de aplicación, se llaman datos. En la capa de transporte al mensaje encapsulado se denomina segmentos. En la capa de red, se identifica como paquete del mensaje encapsulado. En la capa de enlace, el mensaje encapsulado se le llama trama y en la capa física se le llama Bits al mensaje.

3.2. Protocolo de comunicación MQTT

Es un protocolo simple para la transmisión de mensajes cortos de telemetría y control, desde o hacia una red de sensores / actuadores, es un protocolo de comunicación de máquina a máquina, machine to machine o M2M; este protocolo se localiza en las capas superiores del modelo OSI.

Existe un estándar MQTT a cargo del organismo OASIS que actualmente cuenta con la versión 5 del 7 de marzo de 2019; el protocolo MQTT se basa en el modelo de comunicación productor - consumidor.

Figura 2. Protocolo MQTT



De izquierda a derecha, el primer bloque contiene el texto "OSI 5-7" junto a "MQTT", lo que indica que el protocolo MQTT opera en las capas 5 a 7 del modelo OSI, que son la capa de sesión, la capa de presentación y la capa de aplicación.

El segundo bloque está etiquetado con "OSI 4" y "TCP", implicando que el protocolo de control de transmisión (TCP) opera en la capa 4, la capa de transporte.

El tercer bloque está marcado como "Capa OSI 3" e "IP", señalando que el protocolo de internet (IP) corresponde a la capa 3, la capa de red del modelo OSI.

La estrategia comunicativa se basa en la publicación de mensajes específicos (topics) y la suscripción a esos mensajes. Estos mensajes son una trama de datos con un payload que contiene la información real para el servidor o para el resto de nodos. En una red de sensores esta información contiene valores medidos por los sensores, tales como temperatura, humedad, estado, etc.

El broker es el encargado de saber quién está en la red, qué mensajes puede enviar y envía mensajes de un lado a otro.

Cuando existe una aplicación o un nodo que publicará un mensaje, no se necesita saber nada acerca del nodo que recibirá y consumirá el mensaje. Con el protocolo

MQTT la única dirección y puerto que se necesitan conocer es la del broker de mensajes.

Quien publica y quien recibe no necesitan estar conectados a la vez, porque el broker tiene la capacidad de almacenar mensajes, aunque en la mayoría de los casos los mensajes se consumen en tiempo real, generalmente sucede con clientes que no están sincrónicamente conectados.

En resumen, el protocolo MQTT se basa en que alguien publica un mensaje con el identificador de un topic específico, el broker distribuye el mensaje a todos los clientes (aplicaciones o dispositivos) que están suscritos a ese topic y esos clientes reciben y consumen los datos de esos mensajes.

MQTT se usa actualmente en autos, casas y ciudades inteligentes, así como en fábricas, productos de consumo, etcétera.

En conclusión, se puede desarrollar una aplicación que envía información, sin necesidad de saber qué usuario o aplicación la va a utilizar.

3.3. Computación en la nube

A computación en la nube o cloud computing son tecnologías que ofrecen servicios mediante conexión a internet desde cualquier punto y en cualquier momento. Presentan características únicas al servicio del internet de las cosas, ofreciendo recolección, almacenamiento y gestión de los datos recogidos por los dispositivos, además de alta disponibilidad.

La computación en la nube ofrece tres posibilidades en cuanto a modelos de servicio, a saber:

- a. **SaaS (software como un servicio).** Aplicaciones que se ejecutan en la nube, en este caso para el usuario es invisible la infraestructura de red, almacenamiento, servidores, sistemas operativos y la nube asegura correcto funcionamiento del aplicativo. Ejemplos de este modelo son Google Docs, Gmail, Dropbox.
- b. **PaaS (plataforma como un servicio).** En este caso la nube alberga todo lo necesario para construir y entregar aplicaciones basadas en la nube, en este caso el usuario puede controlar la aplicación, pero no puede administrar la infraestructura subyacente de la nube. Ejemplos de este modelo son Openshift y Google App Engine.
- c. **IaaS (infraestructura como un servicio).** En este modelo se proporciona a las empresas servidores, almacenamiento, redes y centros de datos listos para su uso; el proveedor se encarga de la comunicación, el procesamiento o el almacenamiento.

3.4. Pasarelas de cloud para IoT

Debido a la necesidad de gestionar de forma ágil y eficiente la alta cantidad de datos que generan los dispositivos IoT, han nacido plataformas de computación en la nube orientadas a IoT. Ahora se ampliará la información al respecto.

- **Kaa Iot.** Plataforma multipropósito, flexible de código abierto y lista para producción y establecer soluciones de IoT de extremo a extremo, aplicaciones

conectadas y dispositivos inteligentes, ofrece control y supervisión de dispositivos en tiempo real, recopila y analiza datos de sensores, gestiona cantidad infinita de dispositivos en tiempo real, admite uso de protocolos abiertos como MQTT y CoAP.

- **OpenIoT.** Plataforma middleware de código abierto que unifica IoT con los servicios de computación en la nube, proporciona servicios de registro, implementación y descubrimiento de sensores, monitoreo de dispositivos, visualización de información, desarrollo de aplicaciones sencillas.
- **ThingSpeak.** Plataforma de código abierto que permite captura de datos de sensores, almacenamiento en la nube, visualización y análisis de los mismos, con los datos recogidos crea alertas y se comunica a través de servicios de la red, como Twitter, correo, etc.
- **AWS IoT.** Servicio no open source para IoT de Amazon, proporciona gateway de dispositivos, agente de mensajes MQTT, y ofrece servicios de seguridad e identidad, registro de dispositivos.
- **AZURE IoT.** Plataforma creada por Microsoft que ofrece a los usuarios la creación de soluciones IoT acordes con sus necesidades.
- **Google Cloud IoT.** Ofrece herramientas que posibilitan la conexión de dispositivos como procesamiento, almacenamiento y análisis de datos en la nube.

Nodo cliente

Es la capa de percepción que permite obtener o enviar datos al mundo físico o virtual mediante sensores o actuadores, sensores como temperatura, humedad, estado de un dispositivo, sensores lumínicos, etc.

Broker MQTT

Es un elemento primordial para realizar el envío de los datos entre los clientes, debe estar en constante funcionamiento y ser visible a red externa. Entre muchos brokers públicos existen dos bastante usados y disponibles:

- Mosquito Broker: servicio gratuito destinado a la realización de pruebas.
- Cloud MQTT: servicio de pago y asegura disponibilidad del servicio.

Nodo servidor

Este nodo cuenta con funciones de servidor web y base de datos y como cliente MQTT suscrito a los topics de interés y almacena información en la base de datos.

Usuarios finales

Cualquier usuario del servicio que disponga de un terminal con acceso a internet podrá acceder a la aplicación web y visualizar la información proveniente de los sensores de manera sencilla.

3.5. Adquisición de datos en la nube

Para el objetivo final de los datos recogidos provenientes de la capa de percepción (sensores y otros dispositivos), es necesario describir patrones y correlaciones en tiempo real para impactar en forma positiva a las empresas en su toma de decisiones, como, por ejemplo, la analítica de datos en IoT puede controlar la producción, la temperatura en procesos industriales, monitorear la salud de los pacientes, etcétera.

Aquí entra el concepto del Big Data que, según el Instituto Nacional de Estándares y Tecnologías (NIST), consiste en un extenso conjunto de datos principalmente con características de volumen, variedad, velocidad y variabilidad que requieren una arquitectura escalable para un almacenamiento, manipulación y análisis eficientes.

Según la Unión Internacional de Telecomunicaciones (ITU-T), el Big Data es un paradigma para habilitar la colección, almacenamiento, gestión, análisis y visualización –potencialmente con limitaciones de tiempo real–, de un extenso conjunto de datos con características heterogéneas.

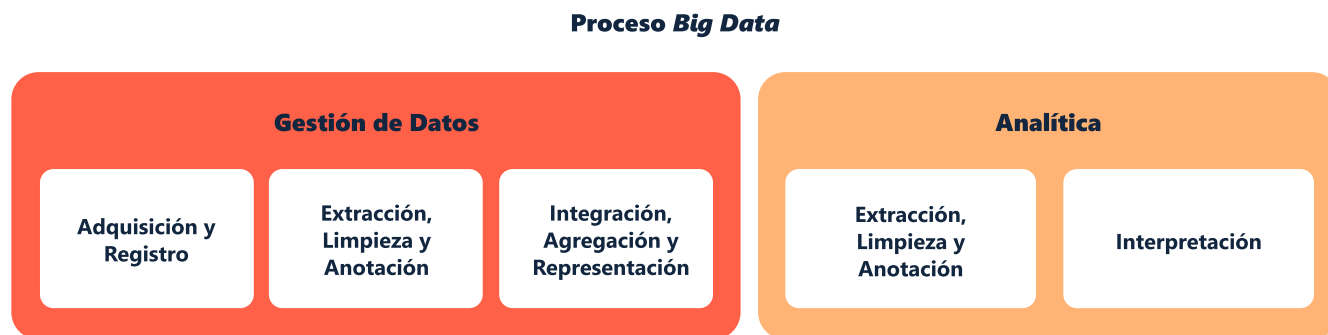
- **Velocidad.** Implica que los datos son generados con muy alta frecuencia y cambian constantemente, en el caso de IoT las medidas captadas por los sensores se realizan con mucha frecuencia.
- **Variedad.** Indica los diferentes tipos de datos recolectados, pueden ser texto, imágenes, videos, audios o logs, estructurados o no estructurados, los datos de IoT pueden ser almacenados en formatos JSON.
- **Veracidad.** Indica el grado de fiabilidad de la información, es clave hacer un filtrado con la tecnología del Big Data de lo que puede ser falso o no.

- **Volumen.** Indica la gran cantidad de datos recolectada de diferentes fuentes de datos, esta cantidad de datos se mide en terabytes (tb), petabytes (PB), incluso a los EB, en el caso de IoT provienen de sensores y dispositivos inteligentes.
- **Valor.** Hace referencia a la información útil de verdad que se puede extraer de los datos. Esa misma ayudará a las empresas a generar un valor agregado para tu negocio, que se convierte en conocimiento y en una acción o decisión.

Para soportar los modelos de datos, se han desarrollado varias plataformas del Big Data como Hadoop, Apache Spark, Apache Flinkm, Apache Storm.

El objetivo del Big Data es extraer la mayor cantidad de valor, información o conocimiento de los grandes volúmenes de datos para lograr conclusiones acertadas y mejorar la toma de decisiones. El proceso de análisis de datos está compuesto por una extracción, limpieza, transformación, modelado y visualización de los datos.

Figura 3. Proceso del Big Data



En la sección Gestión de Datos, hay tres bloques que representan etapas consecutivas: Adquisición y Registro, seguido de Extracción, Limpieza y Anotación, y finalmente Integración, Agregación y Representación.

En la sección Analítica, también hay dos bloques: uno repite Extracción, Limpieza y Anotación, lo que sugiere que estas acciones son relevantes tanto en la gestión como en la analítica de datos. El segundo bloque está etiquetado Interpretación, indicando el paso final en el proceso de analítica de Big Data.

El diagrama resalta los pasos clave en el manejo y análisis de grandes volúmenes de datos.

4. Protocolos de capa de aplicación

En TI, un protocolo de comunicaciones es un procedimiento de criterios que permiten que dos o más entidades de un sistema se comuniquen para transmitir la información por medio de cualquier tipo de conmutación de una extensión física. Dentro de los protocolos de comunicación, las partes involucradas deben acordar los protocolos de red que se dividen en las siguientes siete capas:

1. **Capa 1 - Nivel físico.** Cable coaxial o UTP. Cable de fibra óptica. Proporciona los medios mecánicos y eléctricos funcionales para activar, mantener o desactivar conexiones.
2. **Capa 2 - Nivel de enlace de datos.** Responsable de la transferencia de la información. Permite la comunicación entre la capa de red y la capa física por medio de la Ethernet que viene con cada computadora.
3. **Capa 3 - Nivel de red.** Posibilita que los datos de origen lleguen a su destino. Algunos protocolos de esta capa de red son: Psec, IPv4, IPv6.
4. **Capa 4 - Nivel de transporte.** Transferencia libre de datos confiables entre origen y destino. Destacan los protocolos TCP (provee flujo de bytes confidenciales de extremo a extremo) y UDP (a través del cual las aplicaciones envían datagramas IP encapsulados).
5. **Capa 5 - Nivel de sesión.** Organiza y sincroniza el diálogo y control de intercambio de los datos. El protocolo NetBIOS enlaza un sistema operativo de red con un hardware específico.
6. **Capa 6 - Nivel de presentación.** Trabaja el contenido de la comunicación, considerando que las computadoras manejan de forma distinta la semántica y

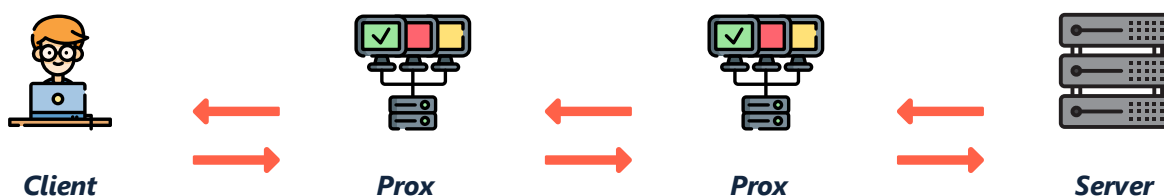
sintaxis de los datos. La norma ASN.1 clasifica los datos en simples o compuestos, y primitivos, contruidos o definidos.

7. Capa 7 - Nivel de aplicación. Puede acceder a las demás capas y definir los protocolos para el intercambio de datos. Por ejemplo, en el correo electrónico, se usa POP y SMTP, y protocolos de transferencia de ficheros como FTP.

4.1. HTTP

El protocolo HTTP, de su sigla en inglés Hypertext Transfer Protocol, permite realizar una petición de datos y recursos, como, por ejemplo, documentos HTML. Este protocolo es la base de cualquier intercambio de datos en la WWW (World Wide Web) que permite hablar un mismo idioma a la hora de transmitir información. El funcionamiento del HTTP se basa en un esquema de petición-respuesta, como se representa en la siguiente figura, donde cada petición individual es remitida a un servidor, que se encarga de gestionar y responder. Entre cada petición y respuesta, hay varios intermediarios, normalmente denominados proxy.

Figura 4. Funcionamiento del protocolo HTTP



De izquierda a derecha, el primer elemento es un ícono de una persona con un laptop etiquetado como Cliente, que envía una solicitud HTTP que pasa por dos íconos de servidores proxy, ambos etiquetados como Proxy. La solicitud llega finalmente a un ícono que representa un servidor de datos, etiquetado como Server.

Hay flechas bidireccionales entre cada par de elementos, indicando que la comunicación es de doble sentido. El diagrama explica visualmente cómo una solicitud HTTP de un cliente puede pasar a través de uno o más proxys antes de llegar al servidor y cómo la respuesta sigue un camino inverso.

Es importante recordar que HTTP se apoya en los protocolos de red y transporte, está desarrollado para ser leído e interpretado por los usuarios, es fácil de usar y se le permite avanzar con nuevas y futuras aplicaciones de internet.

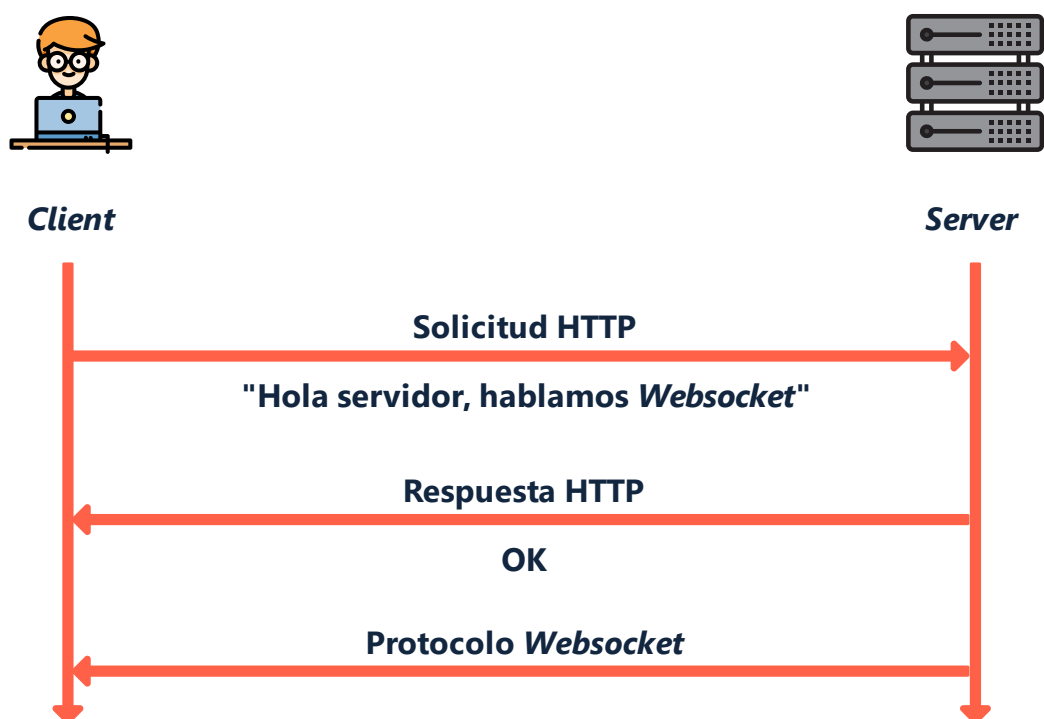
4.2. WEBSOCKET

Websocket es una tecnología avanzada que hace posible generar una sesión de comunicación de intercambio entre el navegador del usuario y un servidor. Con esta API, se puede remitir mensajes a un servidor y recibir contestaciones vigiladas por eventos, sin tener que consultar al servidor para una respuesta. Compatible con todos los navegadores, las peticiones y respuestas se envían en Websocket donde establecen una comunicación en el lenguaje del mismo protocolo.

La siguiente figura ilustra el proceso de utilización de Websocket que inicia con su apertura. Cuando se crea `new WebSocket(url)`, comienza la conexión de manera inmediata. Durante la unión, el navegador (usando cabeceras o “header”) le pregunta

al servidor: “¿Soportas Websockets?” y si el servidor responde “Sí”, la comunicación se prolonga en el protocolo WebSocket, que no es HTTP en absoluto.

Figura 5. Interacción protocolo WebSocket



4.3. JSON

JSON, acrónimo de JavaScript Object Notation, (Notación de objetos de JavaScript) es una sintaxis de serialización de datos jerárquica concisa compatible con todos los navegadores modernos.

Su formato lo convierte en una forma liviana de representar objetos, sin dejar de ser legible por humanos, por lo que ha reemplazado la notación XML en muchas plataformas.

5. Sensores para IoT

Asumiendo que los sensores son una pieza clave en el éxito del internet de las cosas, es importante analizar en qué consisten y su función en la IoT:

- **Sensores de temperatura.** Aunque empleados masivamente en sistemas de aire acondicionado o electrodomésticos, el desarrollo del IoT ha abierto posibilidades en otros ámbitos: producción industrial, agricultura, tecnología médica.
- **Sensores de proximidad.** Utilizados en el ámbito de la seguridad y la videovigilancia, detectan la presencia o ausencia de un cuerpo cercano a las propiedades, convirtiéndolas en una señal de fácil lectura.
- **Sensores de medición de gases.** Utilizados para medir cambios en la calidad del aire, están cada vez más presentes en entornos urbanos con el objetivo de ofrecer información sobre la contaminación atmosférica que soportan los ciudadanos.
- **Sensores de presión.** Habituales en instalaciones industriales, en el mantenimiento de sistemas de suministro de agua y calefacción o para la obtención de pronósticos meteorológicos.
- **Sensores de humedad.** Especialmente importantes en ámbitos como la agricultura, por ejemplo, mejorando la eficiencia de los sistemas de riego en la gestión de un recurso natural tan escaso como el agua.
- **Sensores de nivel.** Los sensores de nivel se utilizan para medir todo tipo de fluidos en estado líquido, gaseoso, materiales de consistencia pastosa, etc.

6. Almacenamiento para IoT

El almacenamiento para los dispositivos y aplicaciones de internet de las cosas requiere nuevas tácticas y rumbos, como los que se relacionan a continuación:

- **Dispositivos IoT de consumo.** Comenzaron la tendencia de IoT. Incluyen termostatos y abridores de puertas de garaje, pero también sistemas de cámaras en el hogar, dispositivos de TV, entre otros dispositivos domésticos inteligentes.
- **Dispositivos IoT empresariales.** Los datos no estructurados generados por los dispositivos IoT son un caso de uso perfecto para procedimientos de almacenamiento de objetos altamente escalables e, incluso, almacenamiento en la nube. La clave: garantizar capacidad de almacenamiento de datos y ancho de banda para el manejo de la velocidad requerida en los sistemas de almacenamiento IoT.
- **Dispositivos IoT de uso industrial.** Proveedores de la nube, como Microsoft con Azure IoT Edge, tienen productos para ayudar en estos casos de uso, pero no siempre funcionan cuando hay restricciones de ancho de banda o latencia.

6.1. Tipos

Existen diferentes tipos de sensores con características, funcionalidades y diversidad de marcas ofrecidas en el mercado. Los razonamientos a tener en cuenta para elegir un buen sensor se resumen en la siguiente infografía:

Selección de sensores para IoT:



Sensor de presión	Sensores de nivel e inclinación	Sensor de humedad	Sensor de carga	Sensor de temperatura
Sensores de movimiento	Sensores de proximidad	Sensores de flujo	Sensores ópticos	Sensores de aceleración y giroscopio
	Sensores de vibración	Sensores químicos	Sensores de fuerza, carga, tensión y torsión	

6.2. Tecnologías iscsi, fiber channel, sas y arreglos raid

El **ISCSI** es una forma sencilla de gestionar el almacenamiento de archivos, bases de datos o discos duros en un almacenamiento centralizado en servidor, que permite el transporte de datos a nivel de bloque entre el iniciador Iscsi y el destino de almacenamiento a través de redes TCP/IP.

La tecnología **fibre channel** transmite datos a alta velocidad entre los dispositivos de almacenamiento y los servidores utilizando 1 gigabit por segundo y soportando diferentes protocolos (SCSI, IP o ATM entre otros). Puede operar sobre cable o fibra óptica a distancias de 10 km sin uso de repetidores.

SAS es una tecnología que utilizan los computadores para transferir datos desde la tarjeta madre al almacenamiento y viceversa. Generalmente es más costoso y resulta

más adecuado para su uso en servidores o estaciones de trabajo informáticas de procesamiento pesado.

RAID es una tecnología de virtualización de almacenamiento de datos que adopta varios componentes de unidad de disco física en una sola unidad lógica para fines de redundancia de los datos, mejorando el rendimiento.

7. Puertos de comunicación

Los puertos de comunicación son las interfaces que se utilizan en las aplicaciones para conectarse con otros elementos, ya sean hardware o software, para enviar y recibir paquetes de datos. En cierta manera, es como el buzón de correos aplicado a un ordenador.

Las principales tecnologías de comunicaciones utilizadas en IoT, son:

- **RFID.** Radio Frequency Identification [12]
- **WSN.** Wireless Sensor Network [13]
- **NFC.** Near Field Communication[14]
- **WiFi.** estándar IEEE 802.11n
- **Bluetooth**

En general, podría decirse que están siendo utilizadas por diferentes protocolos de tecnologías de comunicaciones, dependiendo de la aplicación y su arquitectura técnica de alcance, manejo de datos, seguridad, consumo de energía, vida útil de la batería, etc.

7.1. Analógicos y digitales

Algunos cálculos se pueden medir de forma “analógica”, otros en forma “digital”.

Los sistemas digitales tienen una aceptación significativa en la tecnología moderna, fundamentalmente en la computación y sistemas de control automático. La tecnología digital hace referencia a “cantidades discretas”, con relación a magnitudes o

valores como la distancia y la temperatura o la velocidad, que podrían variar muy lento o muy rápido como un sistema de audio.

En la tecnología analógica es muy difícil almacenar, manipular, comparar, calcular y recuperar información con exactitud cuando ésta ha sido guardada.

Por lo anterior, la mayor parte de proyectos IoT se basan en tecnología digital.

7.2. Programación de bloques

La programación en bloques permite aprender la lógica de programación desde niños, mediante el uso de conexiones sencillas en forma de bloques. Cada bloque tiene una instrucción, condición o evento diferente. Al programar el paso a paso de una función, se deben encajar los bloques de forma ordenada y lógica, de modo que al juntarlos encajan como piezas de lego o rompecabezas, creando pilas o cadenas secuenciales de bloques, es decir, pequeños programas. Entre los principales objetivos de la programación por bloques está el que sea muy simple e intuitiva, ya que no requiere de que sepamos escribir código.

Existen múltiples herramientas y plataformas para incentivar el aprendizaje y volver más funcional la programación por bloques. A continuación, se destacan algunas de ellas:

- a. **Scratch.** Herramienta suficientemente intuitiva y fluida, que permite al usuario trabajar con lenguaje de programación en bloques, usando piezas que se parecen a las de un rompecabezas.

- b. Scratch Jr.** Se trata de una versión más sencilla y digerible de Scratch diseñada para los niños. La manera de presentar la programación en bloques facilita el entendimiento de usuarios de muy corta edad.
- c. Minecraft.** Videojuego muy popular en los últimos años, que insta a los jugadores a trabajar conceptos básicos de programación visual, aunque también, para usuarios con conocimientos avanzados, brinda funciones para programar en lenguajes más tradicionales como JavaScript.
- d. Arduino.** Ofrece dos modalidades de programación para niños: una visual y otra centrada en lenguaje de código escrito. La primera, que funciona con bloques con códigos integrados, es bastante popular en la enseñanza de robótica.
- e. Google Blockly.** Se conforma de un conjunto de comandos que pueden ordenarse como piezas de rompecabezas. Es fácil de utilizar y está enfocado a usuarios pequeños que no esencialmente sepan escribir código.

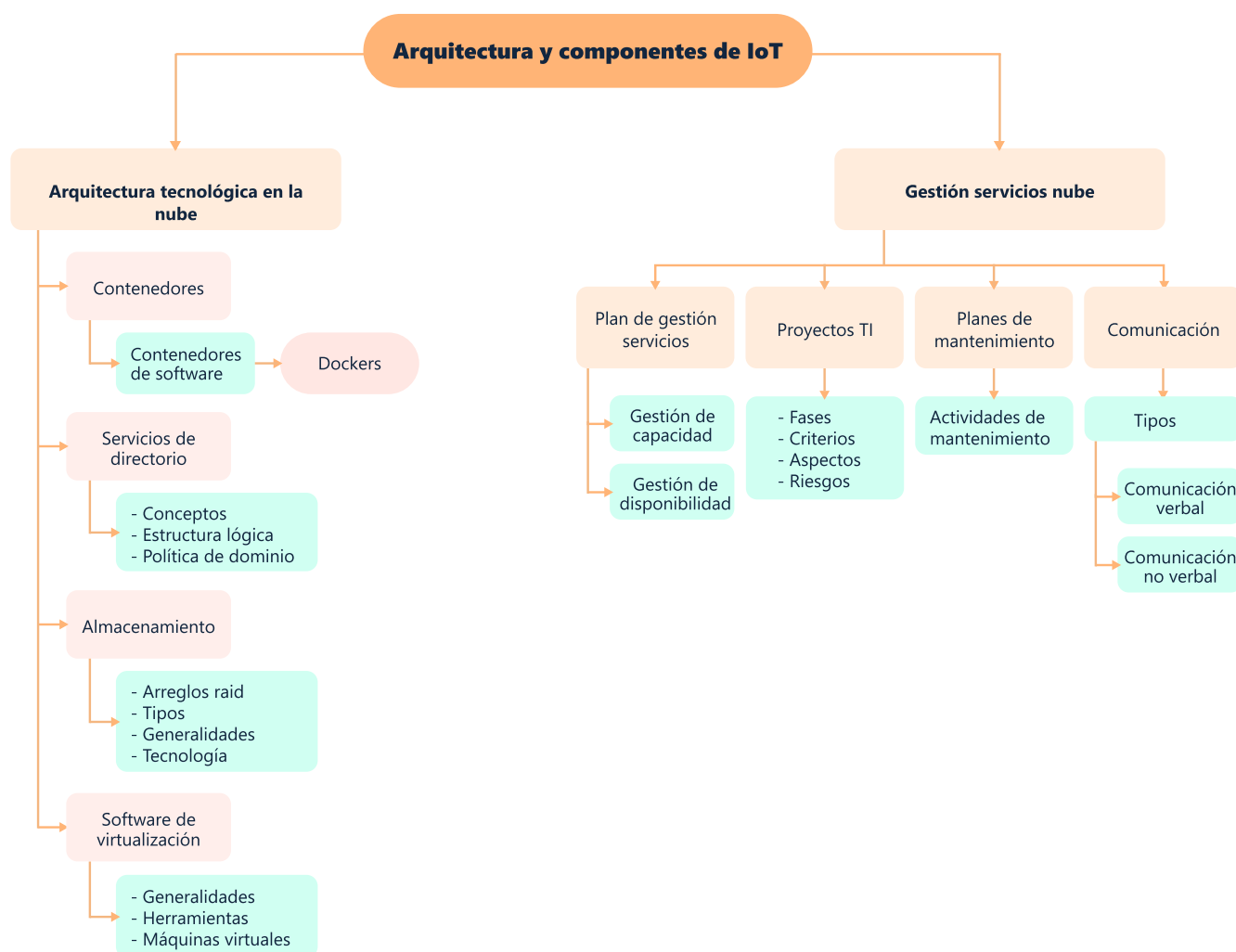
8. Criterios de configuración, interconexión y administración para IoT

Las configuraciones automáticas de dispositivos actualizan grandes conjuntos de gemelos y resumen el progreso y el cumplimiento. En los siguientes pasos generales se describe cómo se desarrolla y usa la administración automática de dispositivos:

1. El integrador o fabricante de hardware de IoT implementa las características de administración de dispositivos dentro de una aplicación integrada, que pueden contener actualizaciones de firmware, instalación y actualizaciones de software, y administración de configuraciones.
2. El desarrollador de soluciones de IoT implementa configuraciones automáticas de terminales. La solución debe incluir la definición de una interfaz de operador para llevar a cabo las tareas de administración de dispositivos.
3. El operador de soluciones de IoT usa la solución de IoT para ejecutar tareas de administración de dispositivos, en concreto, para agrupar los dispositivos, iniciar cambios de configuración (como actualizaciones de firmware), inspeccionar el progreso y corregir los problemas que surjan.

Síntesis

A continuación, se presenta el diagrama que representa el resumen de las temáticas que están desarrolladas en el componente formativo:



Material complementario

Tema	Referencia	Tipo de material	Enlace del recurso
Adquisición de datos en la nube	Agencia española de protección de datos. (2013). Guía para clientes que contraten servicios de Computing. [Archivo PDF].	Documento PDF	https://www.oas.org/es/sla/ddi/docs/Gu%C3%ADa%20para%20clientes%20que%20contraten%20servicios%20de%20Cloud%20Computing%20-%20AGPD.pdf
Computación en la nube	Decreto 1377 de 2013. Por el cual se reglamenta parcialmente la Ley 1581 de 2012. 27 de junio de 2013.	PDF	https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=53646
Computación en la nube	Ley 1581 de 2012. Por la cual se dictan disposiciones generales para la protección de datos personales. 17 de octubre de 2012. D.O. No. 48587.	Libro	https://www.funcionpublica.gov.co/eva/gestornormativo/norma_pdf.php?i=49981
Conectividad IoT	Microsoft. (s.f.). Introduction to AD DS.	Página web	https://docs.microsoft.com/en-us/learn/modules/introduction-to-ad-ds/
Computación en la nube	Ministerio de Tecnologías de la Información y Comunicaciones (2023). Guía dominio MGGTI.G.ST - Gestión De Servicios De TI.	Guía	https://www.mintic.gov.co/arquitecturaempresarial/630/articles-237663_recurso_1.pdf
Computación en la nube	Ministerio de Tecnologías de la Información y Comunicaciones (2021). MinTIC expide la resolución que establece los lineamientos y estándares para la estrategia de seguridad digital.	Artículo	https://mintic.gov.co/porta/inicio/Sala-de-prensa/Noticias/162626:MinTIC-expide-la-resolucion-que-establece-los-lineamientos-y-estandares-para-la-

Tema	Referencia	Tipo de material	Enlace del recurso
			estrategia-de-seguridad-digital
Adquisición de datos en la nube	Public cloud services comparison. (s.f.).	Página web	https://comparecloud.in/

Glosario

Active Directory (AD): base de datos y conjunto de servicios que conectan a los usuarios con los recursos de red que necesitan para realizar su trabajo.

Ciberseguridad: conjunto de elementos articulados para la protección de la información alojados en dispositivos en el ciberespacio.

Cloud computing: la computación en la nube, conocida también como servicios en la nube, consiste en el uso de una red de servidores remotos alojados en Internet para almacenar, administrar y procesar datos e información, en lugar de un servidor local o una computadora personal.

Contenedores: en el sector transporte (por ejemplo, en buques y en trenes), se usan contenedores físicos para aislar diferentes cargas. Las tecnologías de desarrollo de software usan cada vez más un método denominado contenerización.

Continuidad: en tecnología, hace referencia al principio en el que los recursos, información, servicios y aplicaciones se mantienen en funcionamiento continuo ante un evento no deseado.

Contratación: pacto o convenio, oral o escrito, entre partes que se obligan sobre materia o cosa determinada, y a cuyo cumplimiento pueden ser compelidas (Real Academia Española, s.f.).

Disponibilidad: en tecnología, hace referencia al principio en el que los recursos, información, servicios y aplicaciones están disponibles para quienes tienen derecho de acceso a ellos.

Dominio: un dominio cuando hablamos de virtualización almacena una partición de directorio de dominio que consta de información sobre el dominio en el que se encuentra, más el esquema y las particiones del directorio de configuración para todo el bosque.

Framework: marco de trabajo específico, con instrucciones y estructuras, para la implementación de alguna acción.

Incidente: suceso repentino no deseado.

Migración: la migración de la TI consiste en trasladar datos o software de un sistema a otro.

Proveedor: sujeto o entidad que suministra un producto o servicios.

Servicio: conjunto de herramientas y actividades que sirven para responder a una necesidad.

Servicio de TI: conjunto de recursos tecnológicos disponibles para ser utilizados como apoyo operativo de las actividades de negocio dentro de una organización, que entregan valor a la misma, facilitando el cumplimiento de las actividades y objetivos.

Virtualización: tecnología que permite crear servicios de TI útiles, con recursos que están tradicionalmente vinculados al hardware.

Referencias bibliográficas

Microsoft. (2021). Línea base de seguridad de Azure para Azure Active Directory. <https://docs.microsoft.com/es-es/security/benchmark/azure/baselines/aad-security-baseline?toc=/azure/active-directory/fundamentals/toc.json>

Real Academia Española. (s.f.). Contratación. En Diccionario de la lengua española. <https://dle.rae.es/contratación>

Créditos

Nombre	Cargo	Regional y Centro de Formación
Milady Tatiana Villamil Castellanos	Responsable del Ecosistema	Dirección General
Claudia Johanna Gómez Pérez	Responsable de línea de producción	Regional Santander - Centro Agroturístico
Cesar Antonio Villamizar Núñez	Experto temático	Regional Norte de Santander - Centro de la Industria, la Empresa y los Servicios
Maribel Avellaneda Nieves	Diseñadora Instruccional	Regional Norte de Santander - Centro de la Industria, la Empresa y los Servicios
Carolina Coca Salazar	Asesora Metodológica	Regional Distrito Capital - Centro de Diseño y Metrología
Rafael Neftalí Lizcano Reyes	Responsable Equipo desarrollo curricular	Regional Santander - Centro Industrial del Diseño y la Manufactura
Sandra Patricia Hoyos Sepúlveda	Corrección de estilo	Regional Distrito Capital - Centro de Diseño y Metrología
Leonardo Castellanos Rodríguez	Desarrollador Fullstack	Regional Santander - Centro Agroturístico
Lucenith Pinilla Moreno	Desarrolladora Fullstack Junior	Regional Santander - Centro Agroturístico
Yazmín Rocío Figueroa Pacheco	Diseñadora de Contenidos Digitales	Regional Santander - Centro Agroturístico
María Alejandra Vera Briceño	Animadora y Productora Multimedia	Regional Santander - Centro Agroturístico

Nombre	Cargo	Regional y Centro de Formación
Laura Paola Gélvez Manosalva	Validadora de Recursos Educativos Digitales	Regional Santander - Centro Agroturístico
Diana Lizeth Lozada Díaz	Evaluadora para Contenidos Inclusivos y Accesibles	Regional Santander - Centro Agroturístico