

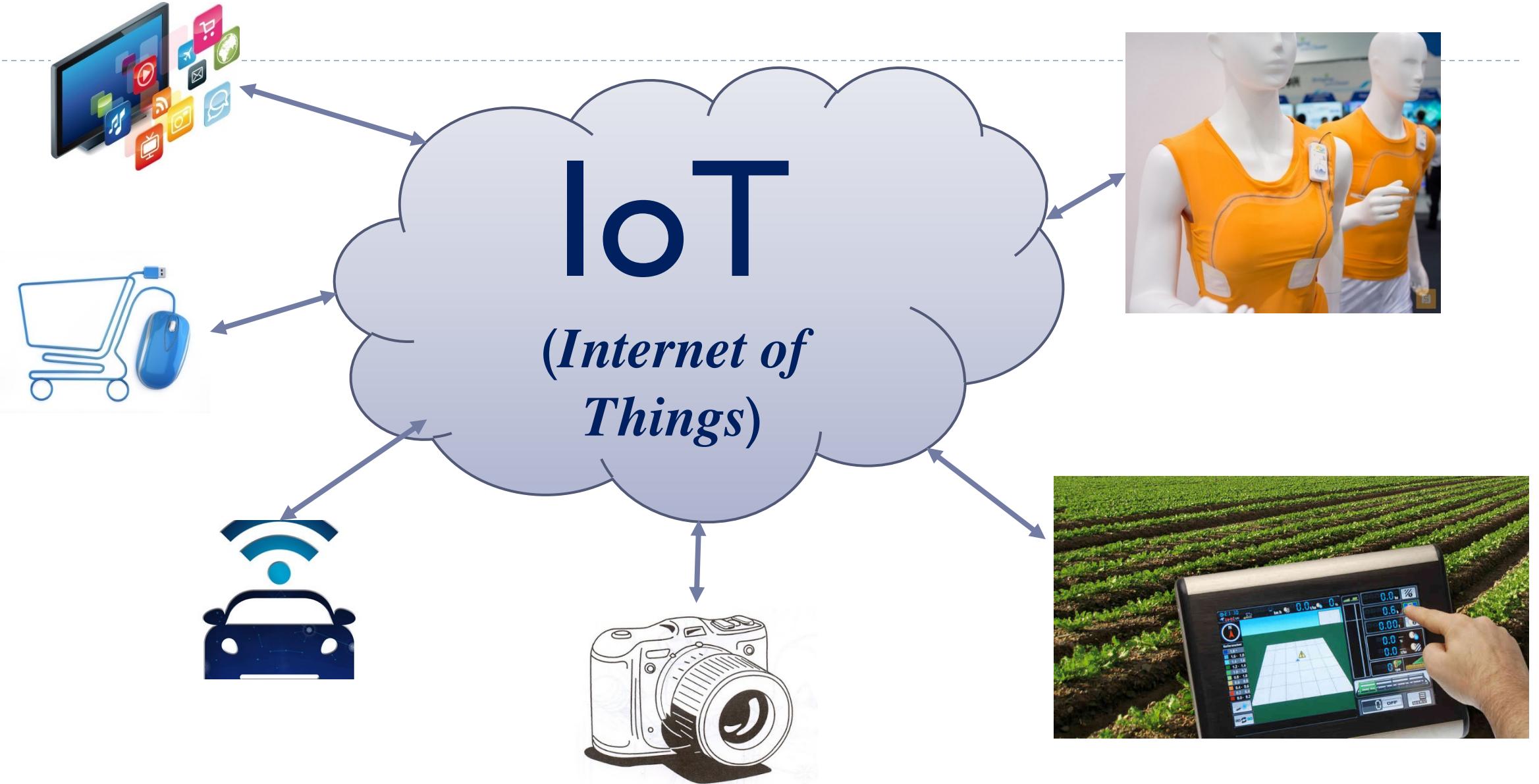
Máster en Tecnologías de Análisis de Datos Masivos: BIG DATA

Internet de las Cosas en el Contexto de Big Data

Tema 2: Herramientas y plataformas para Internet de las Cosas

Juan Antonio Martínez juanantonio@um.es
Antonio F. Skármata Gómez skarmeta@um.es





ÍNDICE

BLOQUE I: Dispositivos y soluciones para recogida de información de sensores

BLOQUE II: Los dispositivos móviles como sensores de datos

BLOQUE III: Plataformas para gestión de la información



BLOQUE I - ÍNDICE

INTRODUCCIÓN

- Arquitectura funcional de un sensor
- M2M vs. IoT
- Gateways en una arquitectura IoT

Dispositivos para Internet de las Cosas

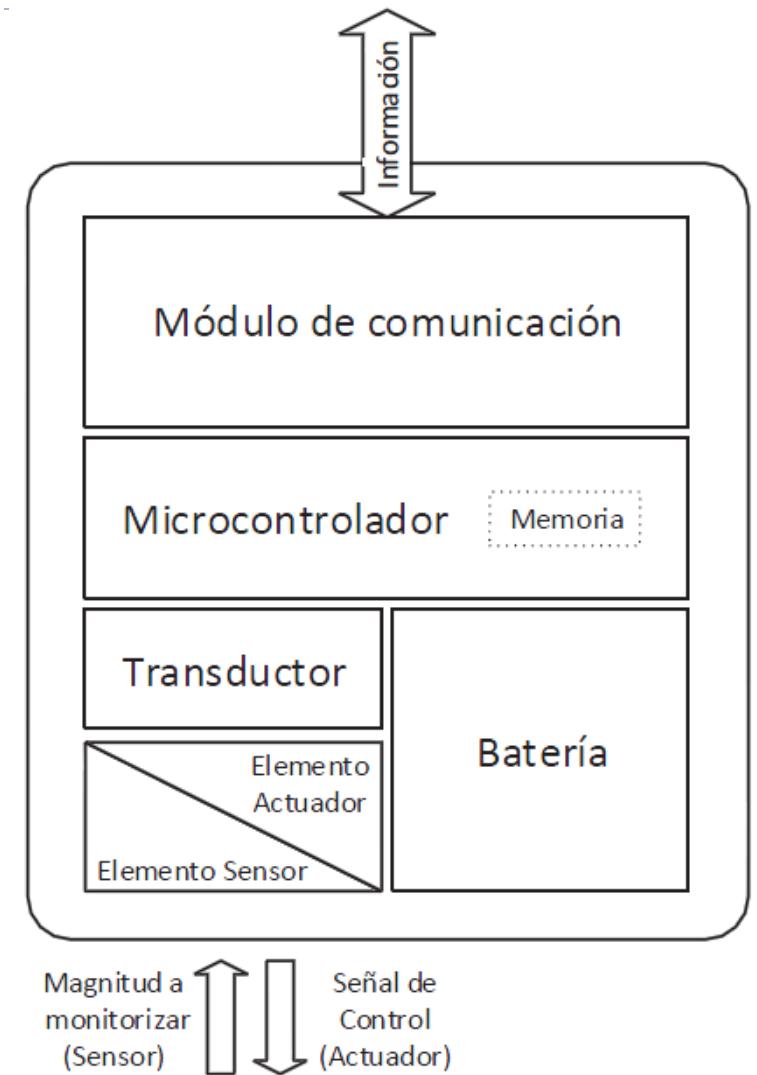
- Gateways OdinS para recogida de datos de sensores
- Otros ejemplos de plataformas hardware comerciales



Arquitectura funcional de un nodo sensor/actuador

La arquitectura de un elemento sensor o actuador se puede dividir en tres bloques funcionales:

- 1) **Bloque sensor/actuador:** define la funcionalidad del dispositivo, como sensor (medición de un parámetro) o como actuador (mecanismo activador/interventor).
- 2) **Bloque de procesado:** conjunto de elementos que dotan al dispositivo de inteligencia de computación y toma de decisiones.
- 3) **Bloque de comunicación:** envía y recibe mensajes hacia y desde los nodos y repetidores vecinos.



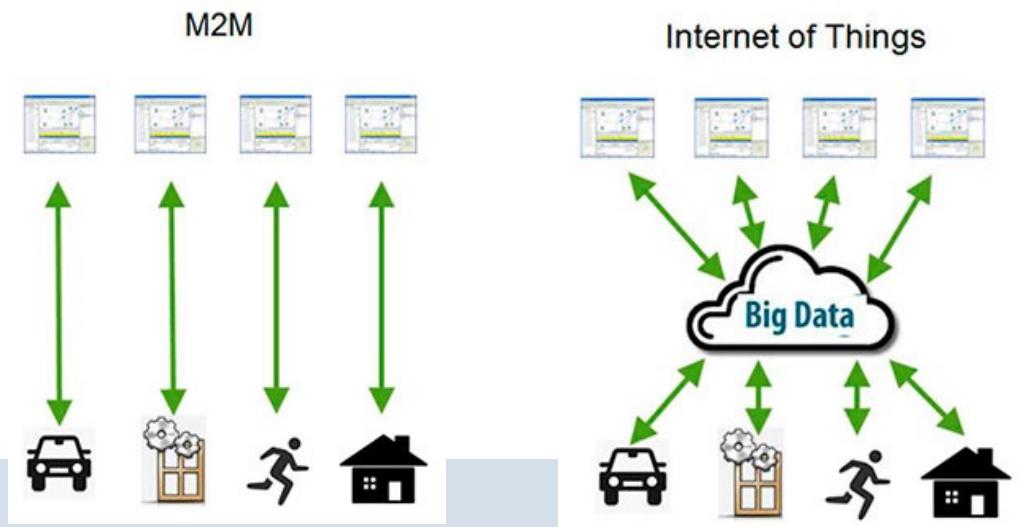
M2M

- M2M → Machine to Machine. Tecnologías que permiten a los dispositivos la comunicación entre ellos.
- M2M no señala una tecnología específica ni en el ámbito de las redes, ni en el de la información o las comunicaciones. Se utiliza para describir cualquier tecnología que permita a diferentes dispositivos conectados intercambiar información a través de redes fijas o móviles de corto o largo alcance, y ejecutar acciones sin la intervención manual de seres humanos.
- El M2M se considera como una de las partes fundamentales del Internet de las Cosas (IoT).
- Al contrario de otras tecnologías más “trending” el M2M es la capa que subyace debajo del glamour, por lo que su explosión comercial va a estar más ligada a un nivel más técnico. Algunas de estas tecnologías de más alto nivel utilizan o utilizarán M2M y serán conductores de su éxito en el mercado.



M2M vs. IoT

- M2M se refiere a dispositivos que capturan eventos, transmiten la información por la red, hacia una aplicación que la traduce a información útil. Mientras IoT es una red de cosas identificables individualmente que se comunican sin interacción humana usando conectividad extremo a extremo.
- El IoT le viene a dar un mayor sentido y una mayor universalidad a lo que ya se tenía con M2M permitiendo que todas las cosas se puedan conectar con las otras sin importar que tipo de solución M2M estemos hablando.
- En M2M las cosas solo envían información que recolectan, en IoT las cosas envían, reciben, procesan información y toman acciones con la ayuda de soluciones de Big Data.



Gateways en una arquitectura IoT

- Normalmente los sensores y dispositivos tienen **capacidades interconexión limitadas**, utilizando protocolos de bajo consumo (Bluetooth Low Energy, Zigbee, LoRaWAN, Sigfox...) para conectarse a redes de área local (LAN) o residencial (HAN), es decir, no tienen acceso directo a Internet



Es necesario un **Gateway para coordinar la red de sensores y actuar de proxy con la red donde la plataforma IoT expone sus interfaces para recibir y proporcionar información.**

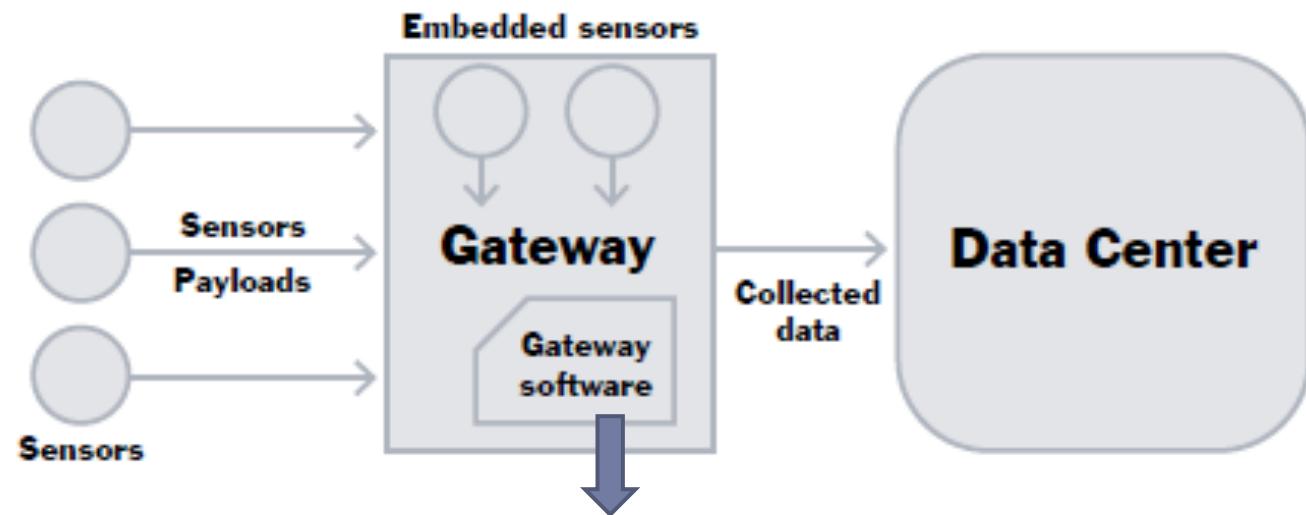
- El tratamiento de la **información en bruto** procedente de sensores y dispositivos, directamente en la plataforma IoT, puede ser **en ocasiones ineficiente** en términos de rendimiento y ancho de banda.



Es necesario disponer de Gateways con capacidad de almacenamiento temporal de la información (caching) y procesamiento de la misma. De este modo es posible filtrar y agregar la información antes de enviarla a la plataforma, así como garantizar que no se pierde información en caso de interrupción temporal de las comunicaciones.

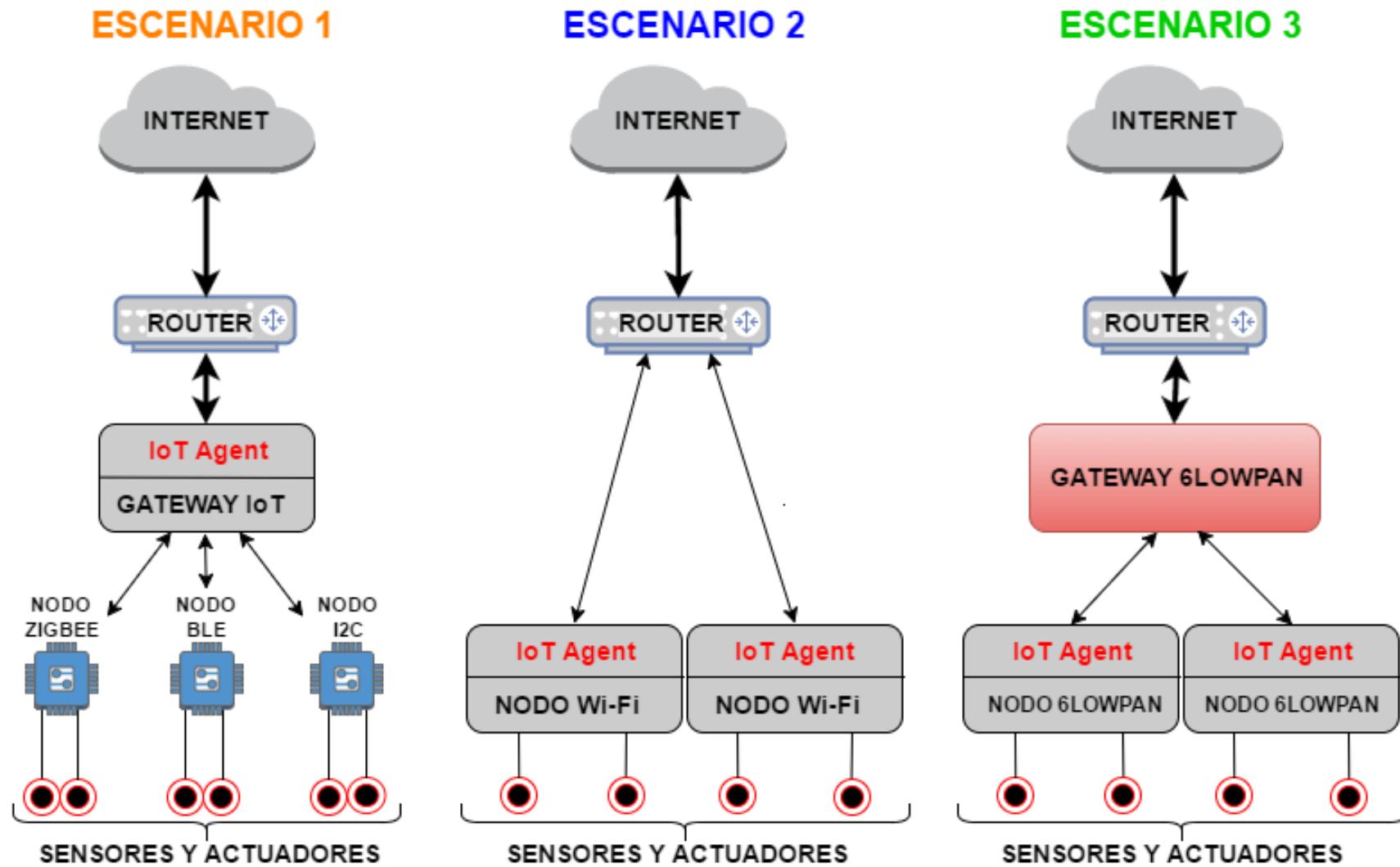
Gateways en una arquitectura IoT

- Disponer de un Gateway que **centralice las operaciones** de monitorización de todos los dispositivos conectados a su red de sensores, **reduce la complejidad** del sistema de monitorización, que sólo se tiene que conectar al Gateway.

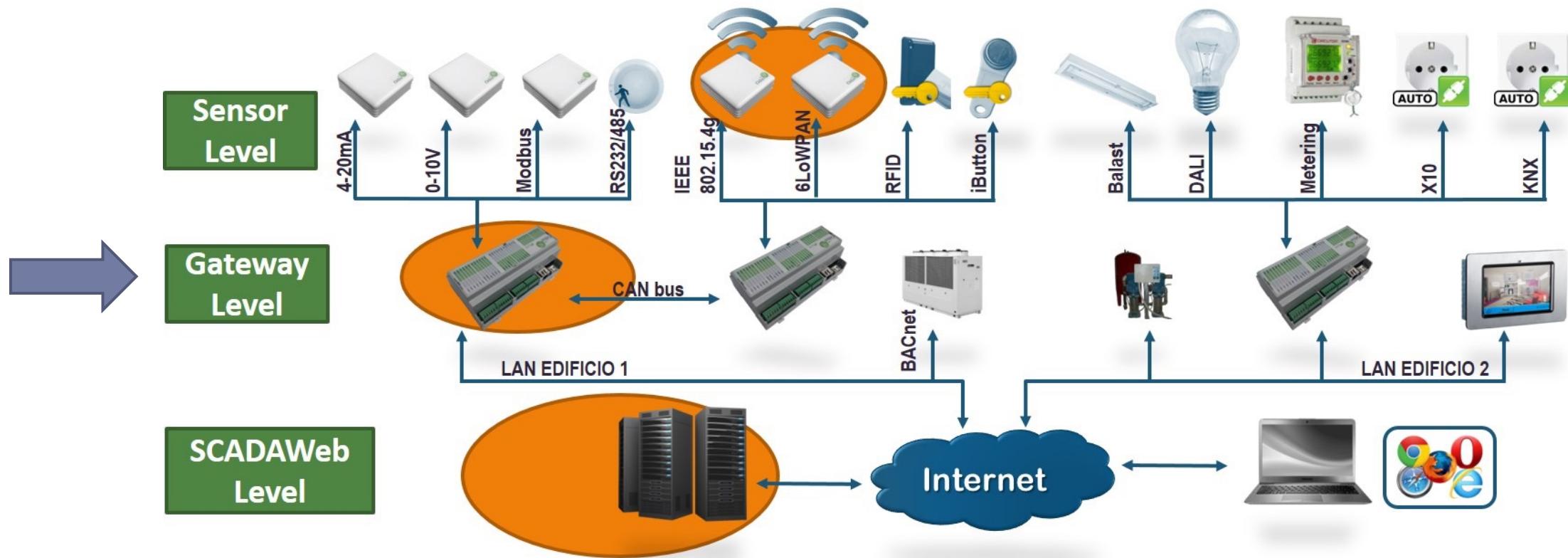


- **Recuperación ante fallos:** Interrupción de la alimentación, caída de comunicaciones, fallos en sensores o dispositivos...
- **Soporte para actualizaciones automáticas,** gestionadas desde entorno remoto centralizado.
- **Soporte para configuración,** gestionada desde entorno remoto centralizado.

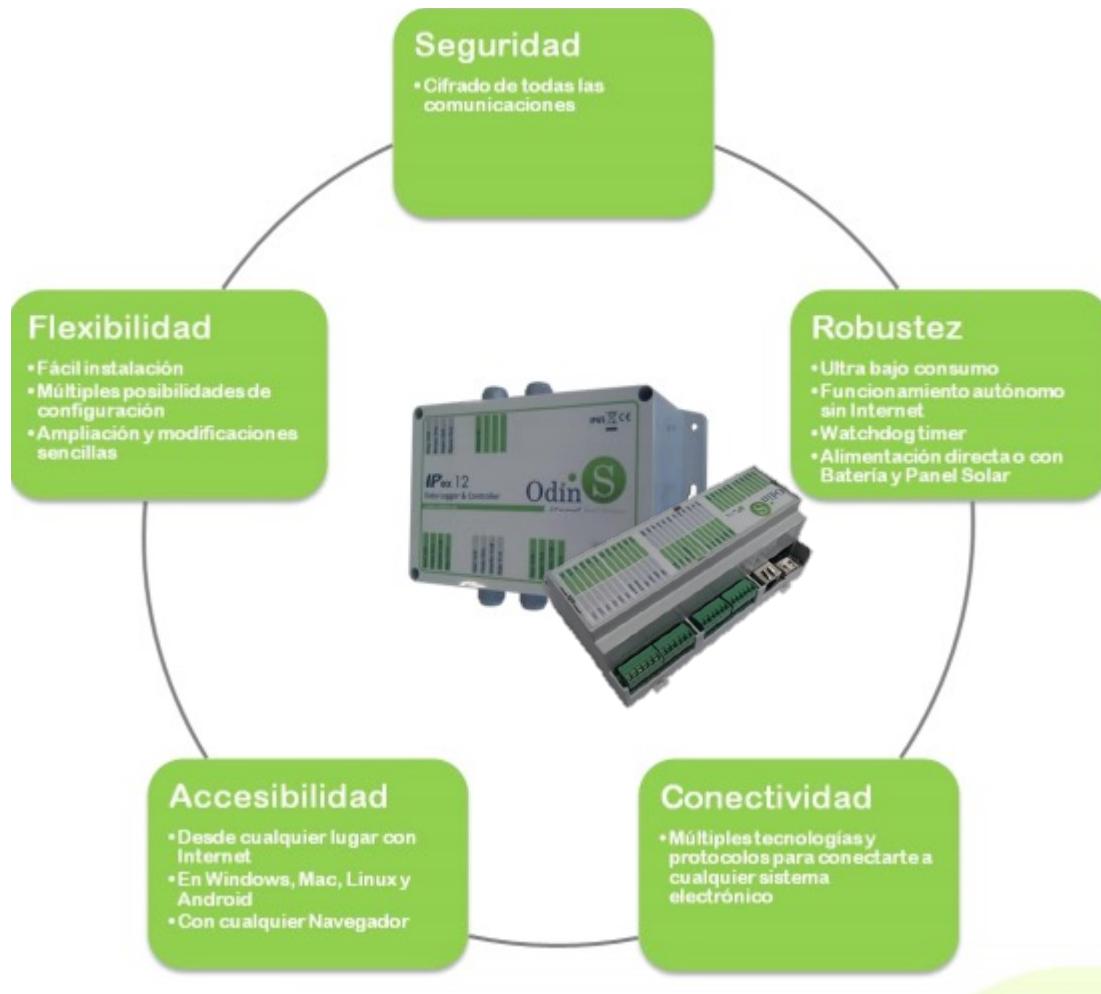
Gateways en una arquitectura IoT: Tipos de conexión



Gateways para recogida de datos de sensores



Gateways para recogida de datos de sensores.



CPU 32bits, 4MB ampliable por microSD. Interfaz Eth,USB, CAN, 3xRS232, 1xRS485 (soporta Modbus).

16 I/O configurables:

- Entradas digitales
- Entradas analógicas 0-10V o 4-20mA
- Salidas digitales (para relés)
- Salidas analógicas (0-10V)



Hasta 16 expansiones esclavas adicionales.

Puertos de comunicación extra: 3G, RF 433/868Mz, 6LowPAN/Zigbee, DALI, KNX.

IPex16

CPU de 16bits, 4MB ampliable con microSD a 32GB. Interfaz GPRS/3G, Serie, USB, IPexBus y SDI-12. Opción de puerto radio 802.15.4g.

- 4 entradas digitales contadoras de pulsos
- 4 salidas digitales (para válvulas y relés)
- 4 entradas analógicas por tensión (0-10V) y corriente (0-20mA) configurable por software.



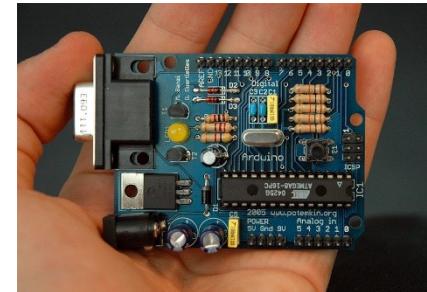
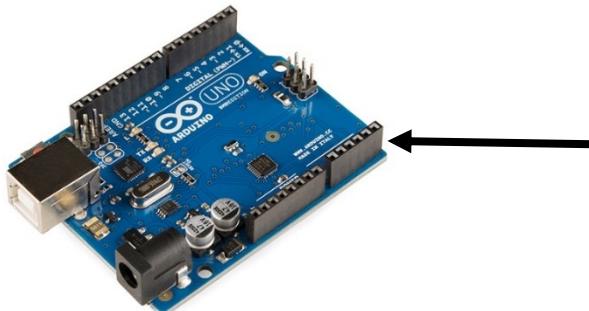
Hasta 12 expansiones esclavas adicionales

IPex12



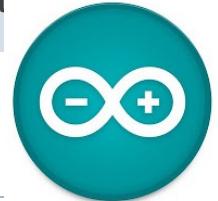
Arduino

- Arduino es una plataforma de creación de prototipos electrónicos de código abierto, basada en una placa con un microcontrolador y un entorno de desarrollo para crear software (programas) para la placa.
- Está diseñada para facilitar el uso de la electrónica en proyectos multidisciplinares. Puede tomar información del entorno a través de sus entradas y controlar luces, motores y otros actuadores.
- El **hardware** consiste en una placa con un **microcontrolador** y **puertos de entrada/salida**.
- El **software** consiste en un entorno de desarrollo que implementa el lenguaje de programación **Processing/Wiring**.
- Los proyectos hechos con Arduino pueden ejecutarse sin necesidad de conectar a un computador. Incluye además un cargador de arranque que es ejecutado en la placa

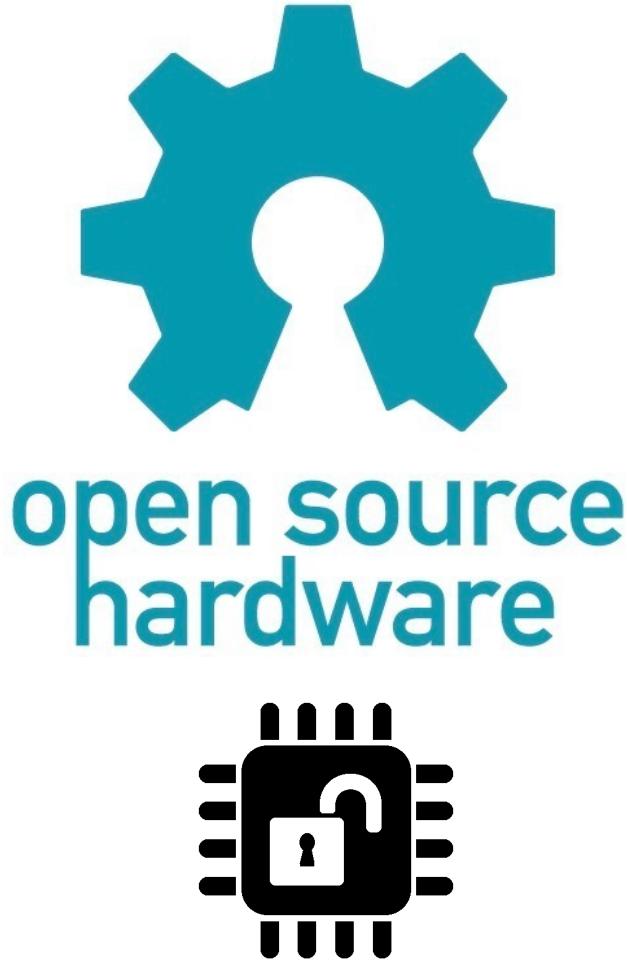


RS232

CONEXIÓN A INTERNET
13



Arduino



Al ser **Open-Hardware**, tanto su diseño, como su distribución es libre.

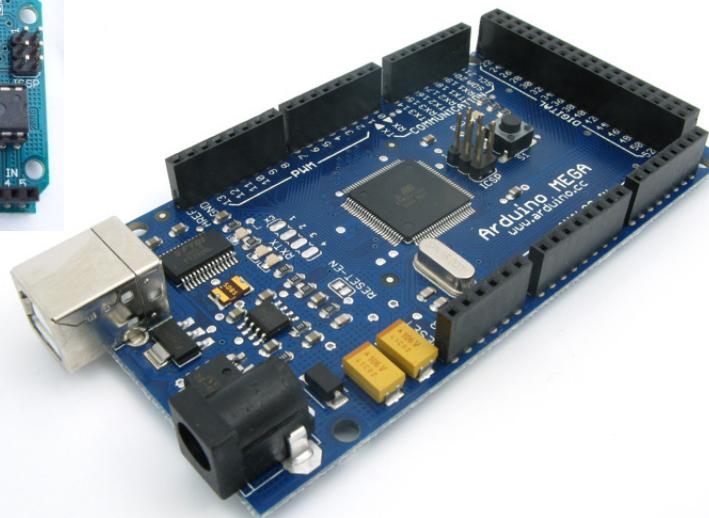
Es decir, puede utilizarse libremente para el desarrollo de cualquier tipo de proyecto sin necesidad de adquirir ninguna licencia



Arduino



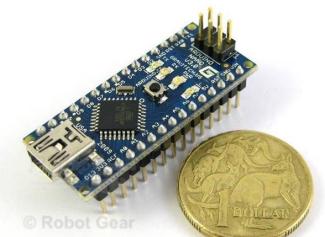
Duemilanove



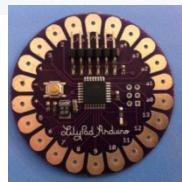
Mega



Uno



Nano



Lilypad



Mini



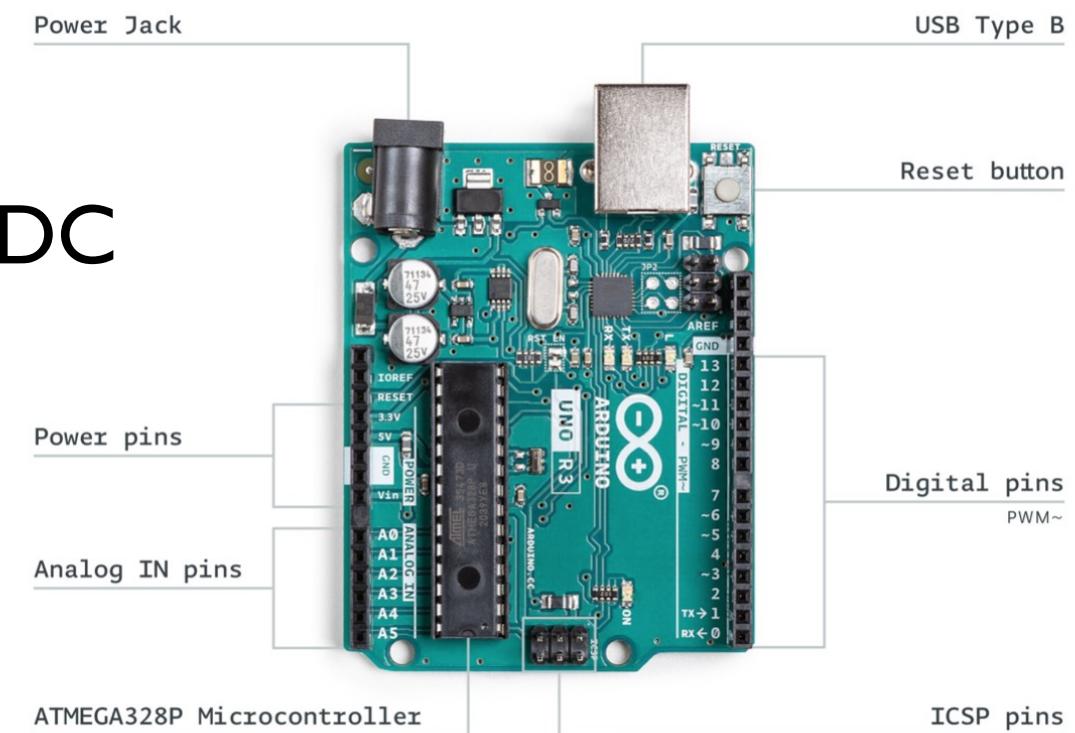
Leonardo



Arduino

- Arduino Uno: características

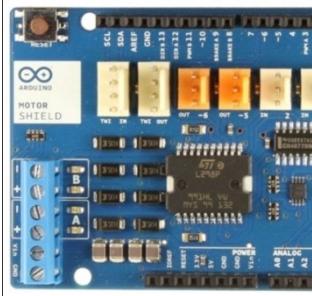
- Microcontrolador: ATMega328
- Voltaje de operación: 5V DC
- Voltaje de alimentación: 7 – 12V DC
- Pines digitales I/O: 14
- Pines entrada analógica: 6
- Interfaz de programación: USB
- Frecuencia del Reloj: 16 MHz



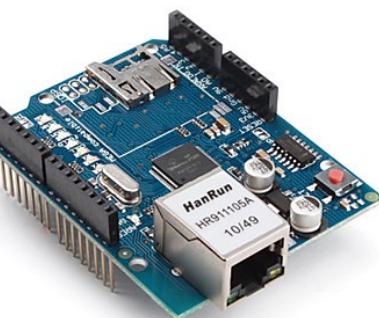
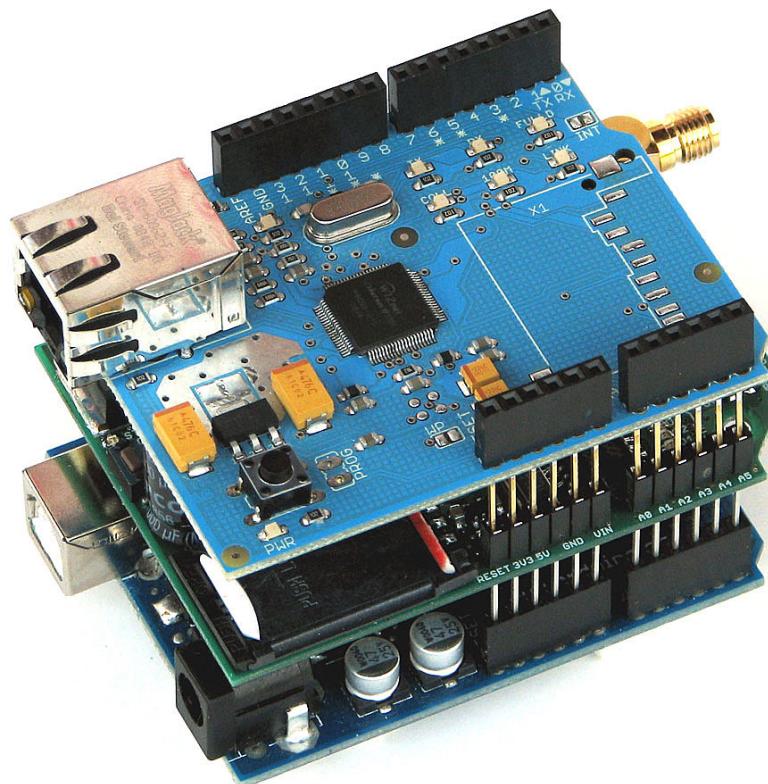


Arduino

- Arduino Shields: expansión de funcionalidad
- Permiten añadir a la placa básica de Arduino nuevas características: comunicación, sensorización, E/S en memorias externas, etc.



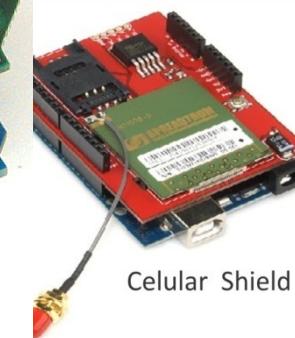
Motor Shield



Ethernet shield



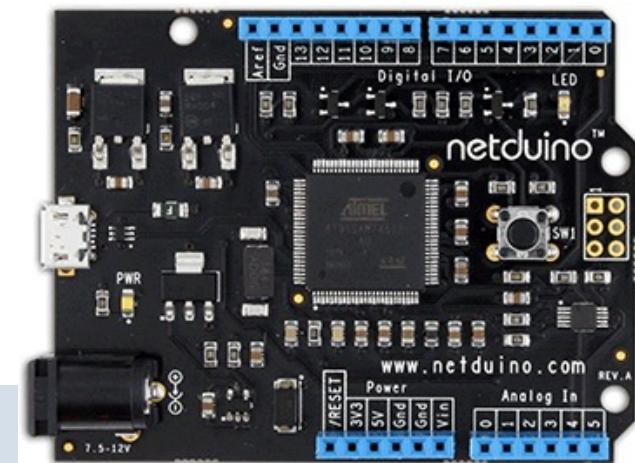
GPS - GSM Shield



Celular Shield

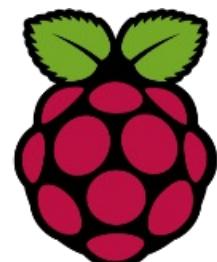
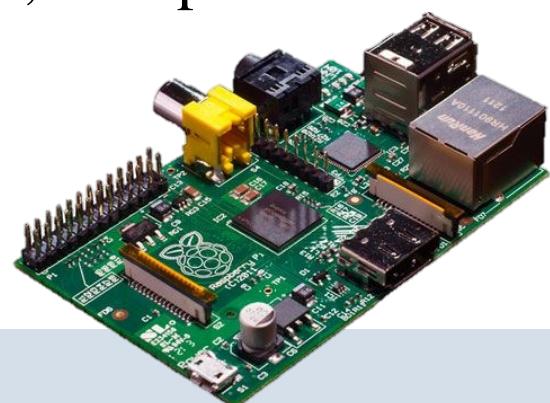
Netduino

- Netduino es una plataforma hardware que facilita la implementación de robots y pequeños autómatas, con **lenguajes de alto nivel**. El sistema operativo de la tarjeta es el .NET Micro Framework. Son programables en **C#**, directamente desde Microsoft Visual C# Express 2010, y son extremadamente potentes y flexibles.
- Netduino Plus incluye 22 GPIO con SPI, I2C, 4 UART (1 RTS / CTS), 6 PWM y 6 canales de ADC para interactuar con los interruptores, sensores, LEDs, dispositivos de serie, y mucho más. Además esta placa es ampliable con accesorios de terceros fabricantes que ofrecen una funcionalidad pre-construida como localización GPS y control de energía de la batería.



Raspberry Pi

- Raspberry Pi comenzó a fabricarse en 2012 como ordenador de bajo coste para facilitar la enseñanza de la informática en los colegios.
- Se trata de una **placa base** de pequeñas dimensiones (85 x 54 milímetros) en la que se aloja un chip Broadcom BCM2835 con procesador ARM hasta a 1 GHz de velocidad, GPU VideoCore IV y hasta 512 MB de memoria RAM.
- Se puede añadir teclado, ratón o cualquier dispositivo que funcione por USB, y conectarlo a un monitor o a una pantalla de televisión (HDMI). Su disco duro es conectado de forma externa, bien por USB o con una tarjeta SD.



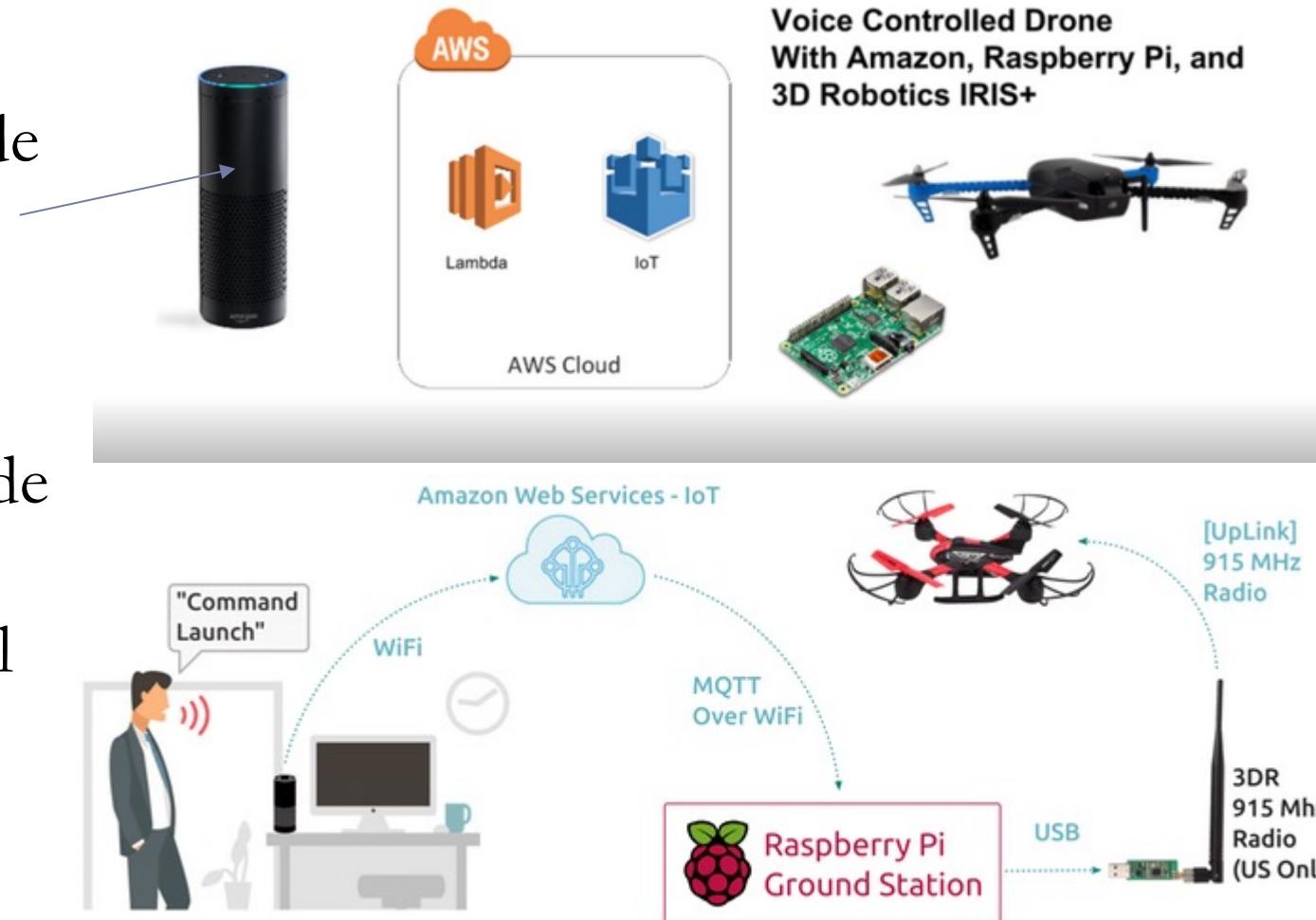
Raspberry PI 5



- Broadcom BCM2712 2.4GHz quad-core 64-bit Arm Cortex-A76 CPU, with cryptography extensions, 512KB per-core L2 caches and a 2MB shared L3 cache
- VideoCore VII GPU, supporting OpenGL ES 3.1, Vulkan 1.2
- Dual 4Kp60 HDMI® display output with HDR support
- 4Kp60 HEVC decoder
- LPDDR4X-4267 SDRAM (4GB and 8GB SKUs available at launch)
- Dual-band 802.11ac Wi-Fi®
- Bluetooth 5.0 / Bluetooth Low Energy (BLE)
- microSD card slot, with support for high-speed SDR104 mode
- 2 × USB 3.0 ports; 2 × USB 2.0 ports
- Gigabit Ethernet, with PoE+ support (requires separate PoE+ HAT)
- 2 × 4-lane MIPI camera/display transceivers
- PCIe 2.0 x1 interface for fast peripherals (requires separate M.2 HAT or other adapter)
- 5V/5A DC power via USB-C, with Power Delivery support
- Raspberry Pi standard 40-pin header
- Real-time clock (RTC), powered from external battery

Raspberry Pi: Ejemplo de aplicación

Amazon Echo es un dispositivo de comando de voz de Amazon con funciones, incluyendo respuesta a preguntas y reproductor de música. El dispositivo responde al nombre de “Alexa”.



Fuente <https://www.hackster.io/veggiebenz/voice-controlled-drone-with-raspi-amazon-echo-and-3dr-iris-c9fd2a>

BeagleBone

- BeagleBone es un ordenador de placa reducida basada en Linux. Diseñada por Texas Instruments con una filosofía open-source y de bajo coste.
- Utilizada ampliamente para aplicaciones de visión artificial



	Raspberry Pi 2	Beaglebone Black Rev C
Processor	900Mhz Quad Core ARM	1 Ghz ARM Processor
RAM	1 Gb	512MB DDR3
Storage	Micro SD Card (Required) Expandable via USB	4 GB on board flash storage. Expandable by USB & Micro SD
Audio	Stereo over HDMI or 3.5mm Jack	Stereo over HDMI
GPIO**	26/40 Pins	65/92 Pins
HDMI	Full Size HDMI port	Micro HDMI
Peripherals	4 USB ports 1 10/100 Ethernet port	1 USB port 1 10/100 Ethernet port
Power Source	Micro USB	Micro USB or 5VDC connection

Nanode



Nanode es un panel de microcontrol de código abierto que tiene **conectividad a Internet integrada**. Es un pilar fundamental y de bajo coste para facilitar la experimentación con el Internet de las Cosas.

Creación de proyectos de hardware de código abierto capaces de interactuar con aplicaciones con base en la nube y eventos en el entorno online.



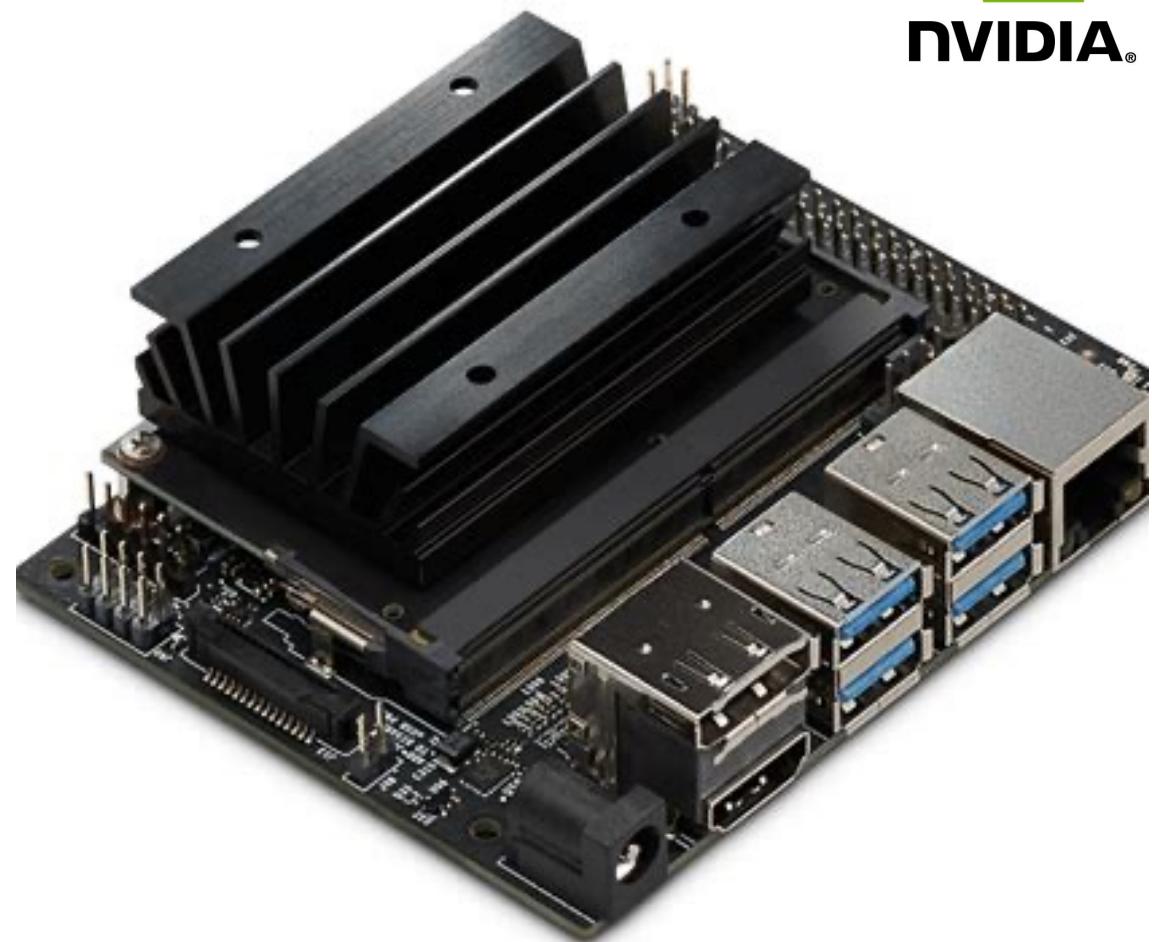
WaspMote

- **WaspMote** es una plataforma modular **Open Source**, desarrollada por Libelium, optimizada en cuanto a consumo y aplicaciones. Pensada para la creación de grandes redes de sensores, con gran autonomía energética.
- Utiliza el mismo entorno de desarrollo que Arduino.
- Usa el protocolo de comunicación Zigbee, dispone de diferentes módulos para dotarle de diferentes medios de comunicación adicionales (Bluetooth, GPS, y GPRS).

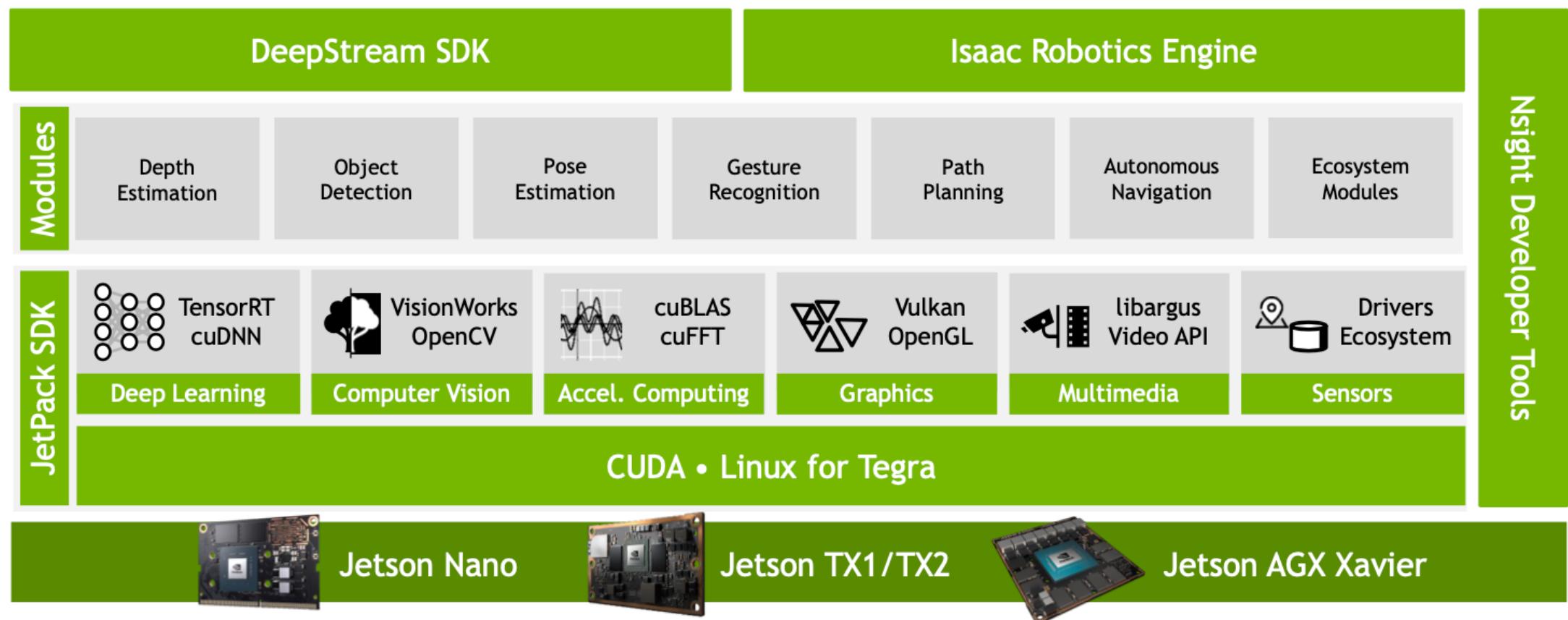


Jetson Nano

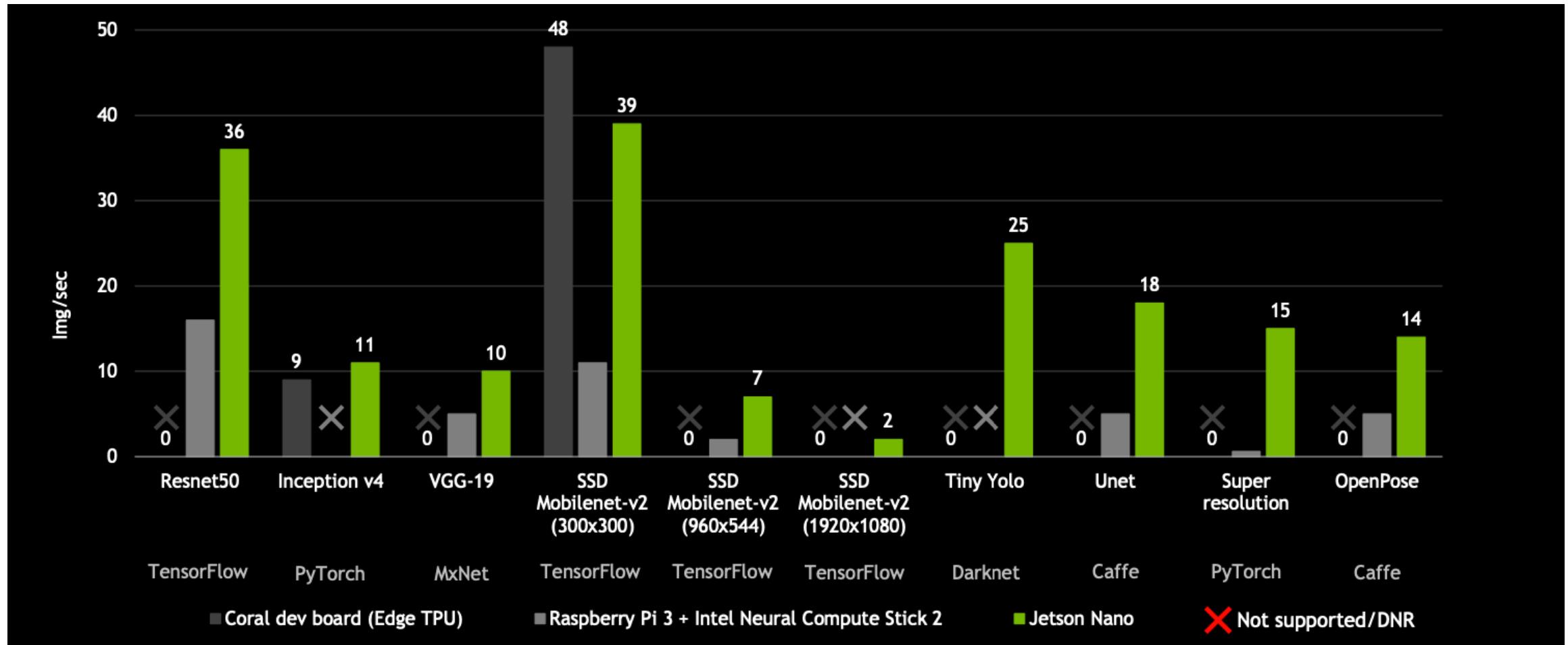
Developer Kit Technical Specifications	
GPU	128-core NVIDIA Maxwell™
CPU	Quad-core ARM® A57 @ 1.43 GHz
Memory	2 GB 64-bit LPDDR4 25.6 GB/s
Storage	microSD (Card not included)
Video Encoder	4Kp30 4x 1080p30 9x 720p30 (H.264/H.265)
Video Decoder	4Kp60 2x 4Kp30 8x 1080p30 18x 720p30 (H.264/H.265)
Connectivity	Gigabit Ethernet, 802.11ac wireless*
Camera	1x MIPI CSI-2 connector
Display	HDMI
USB	1x USB 3.0 Type A, 2x USB 2.0 Type A, 1x USB 2.0 Micro-B
Others	40-pin header [GPIO, I2C, I2S, SPI, UART] 12-pin header [Power and related signals, UART] 4-pin fan header*
Mechanical	100 mm x 80 mm x 29 mm



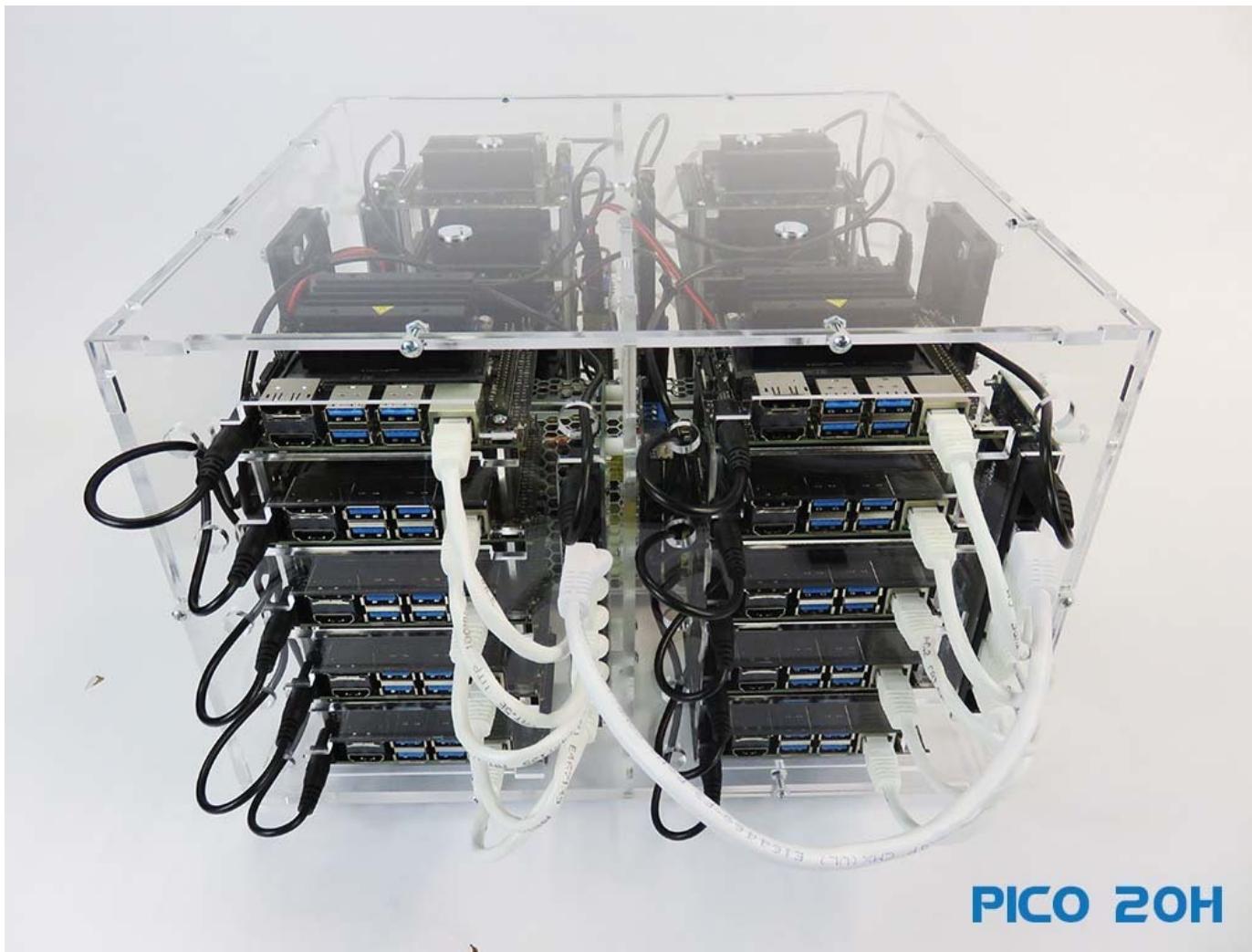
Jetson Nano



Jetson Nano



Jetson Nano



ÍNDICE

BLOQUE I: Dispositivos y soluciones para recogida de información de sensores

BLOQUE II: Los dispositivos móviles como sensores de datos

BLOQUE III: Plataformas para gestión de la información

BLOQUE III - Índice

- Características generales de los dispositivos móviles
- Smartphones como sensores de datos
- WEARABLES como sensores de datos
- Código QR (Quick Response)
- Tecnología RFID & NFC



Características generales de los dispositivos móviles

- Son aparatos pequeños.
- La mayoría de estos aparatos se pueden transportar en el bolsillo del propietario o en un pequeño bolso.
- Tienen capacidad de procesamiento.
- Tienen conexión permanente o intermitente a una red.
- Tienen memoria (RAM, tarjetas MicroSD, flash, etc.).
- Normalmente se asocian al uso individual de una persona.
- Tienen una alta capacidad de interacción mediante la pantalla o el teclado

Movilidad

Tamaño reducido

**Comunicación
inalámbrica**

**Interacción con
las personas**



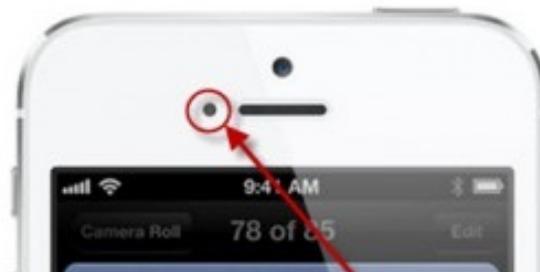
Smartphones como sensores de datos

- El **giroscopio** es el que permite girar la pantalla.
- El **acelerómetro** mide la aceleración que el teléfono experimenta con respecto a la fuerza de la gravedad, es decir, detecta el movimiento y la orientación.
- El **magnetómetro** es capaz de detectar campos magnéticos. El magnetómetro es uno de los sensores que utilizan las aplicaciones de la brújula para señalar el polo norte. Las aplicaciones para detectar metal también usan este sensor.
- El **podómetro** es el encargado de medir con precisión los pasos que da el usuario.



Smartphones como sensores de datos

- El **sensor de proximidad** detecta señales u objetos que se encuentran cerca de él y, normalmente, se sitúa cerca del auricular del teléfono para reconocer cuándo el usuario pega la oreja al hablar por el móvil.
- El **sensor de luz** mide el brillo de la luz ambiental. Así, el software del teléfono utiliza estos datos para ajustar el brillo de la pantalla automáticamente.
- El **termómetro** controla la temperatura interior del dispositivo y la batería para evitar daños.



Smartphones como sensores de datos

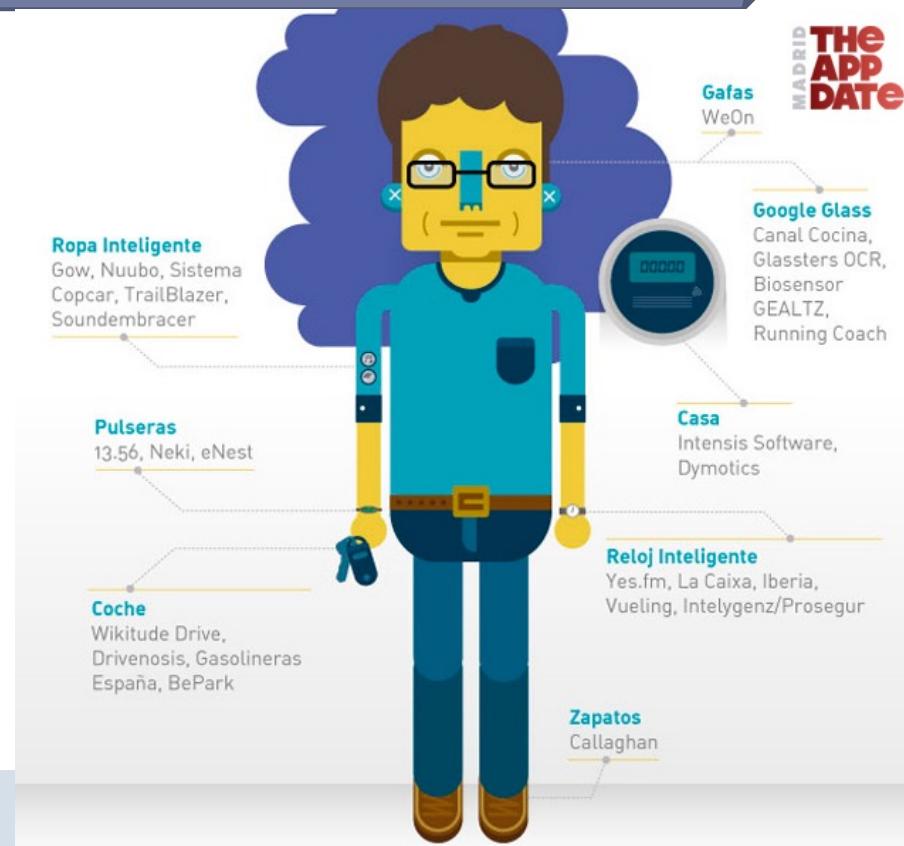
- **Sensor de huellas dactilares.** Un sensor de huellas dactilares permite darle muchos usos, como el desbloqueo de pantalla o para pagar con el móvil. El usuario desliza o apoya su dedo en el smartphone y éste reconoce su huella.
- El **pulsómetro** sirve para medir las pulsaciones, el ritmo cardiaco del usuario a través de los vasos sanguíneos de los dedos.
- El **sensor de humedad del aire** mide la humedad que hay en el ambiente y se utiliza, por ejemplo, en las aplicaciones de health para saber si el usuario está o no en su 'zona de confort', es decir, si hay la temperatura óptima de aire y humedad.



WEARABLES como sensores de datos

Con **WEARABLE** se denota a toda la tecnología que se lleva puesta, como si se tratara de una prenda o de un complemento de nuestra indumentaria habitual. La particularidad de estos dispositivos es que interactúan continuamente con el usuario y con otros aparatos electrónicos con la finalidad de realizar alguna función específica, recopilando gran cantidad de datos.

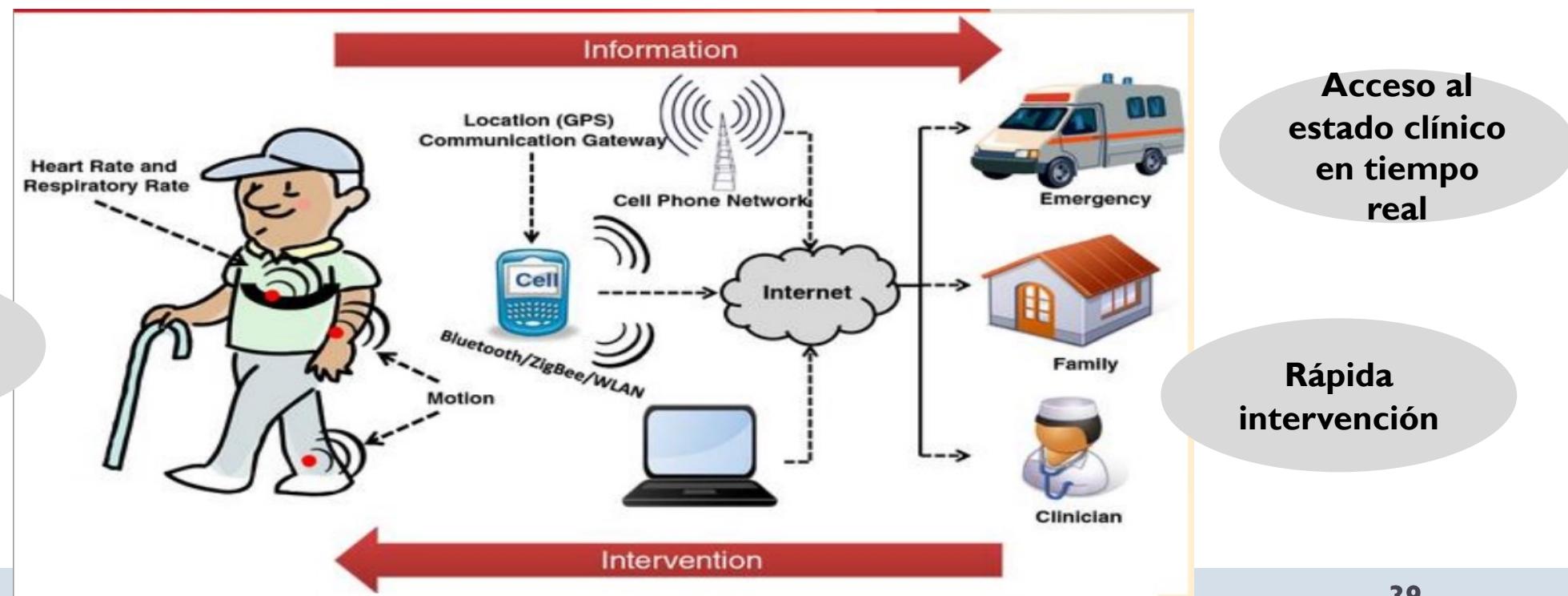
- **Desmonopolizar la atención constante del usuario:** no aísla al usuario del mundo exterior. El usuario puede atender otras cosas mientras usa WEARABLES.
- **No restrictivo para el usuario:** no debe restringir la movilidad del usuario mientras este realiza otras tareas.
- **Observable por el usuario:** en todo momento o dentro de limitaciones aceptables. El medio de salida es constantemente perceptible por el portador.
- **Controlable por el usuario:** Puede tomar control en el momento que lo desee.



WEARABLES como sensores de datos

Wearables de salud

- *E-health*, alude a la práctica de cuidados sanitarios apoyada en tecnologías de la información y las comunicaciones (TIC). Las estadísticas recogidas por estas prendas electrónicas WEARABLES permiten conocer determinados parámetros sanitarios de la persona que las lleva.



WEARABLES como sensores de datos

Google Shoes



- Las Google Shoes son unas zapatillas inteligentes capaces de conversar con el usuario.
- Las zapatillas son capaces de registrar los datos generados por cada movimiento realizado por el usuario y guardarlos en una aplicación web en el smartphone del propietario y se traducen en tiempo real en un comentario divertido y motivador.
- Estos comentarios pueden ser subidos a los medios sociales a través de la aplicación.

WEARABLES como sensores de datos

Smartwatches



- Incluyen funciones como podómetros, medidores de ritmo cardiaco, GPS, control de los ciclos del sueño, a parte de poseer muchas de las funciones de un Smartphone.
- Idóneos para **monitorizar** nuestro entrenamiento deportivo, usarlos como guías de viaje o controlar las notificaciones que recibamos sin necesidad de visualizar directamente el Smartphone.
- Muchas alternativas: Sony, Garmin, Apple Watch, etc.

<https://www.youtube.com/watch?v=WCNT27BBGWQ>

WEARABLES como sensores de datos

Fitness Trackers



- Su tamaño pequeño y su aspecto moderno hacen que sus **funcionalidades** se vean **reducidas**, hasta incluso sólo una funcionalidad.
- Las pulseras fitness pueden ser un elemento motivador para gente que lleva un estilo de vida sedentario.



<https://www.mi.com/es/product/xiaomi-smart-band-7/>

WEARABLES como sensores de datos

Smart Glasses

- Dentro de la categoría de las gafas inteligentes destacan **GoogleGlass**. Se utilizan únicamente con la voz. Permiten estar conectados pudiendo sacar fotos, vídeos o realizar llamadas desde las propias gafas.

Realidad aumentada <https://www.youtube.com/watch?v=0i4v0Texqco>

- XREAL Air 2 Ultra: 1080p (120Hz)
- Meta Quest 3



WEARABLES como sensores de datos

▶ Apple VISION PRO



EyeSight. An outward display reveals your eyes while wearing Apple Vision Pro, letting others know when you are using apps or fully immersed.

WEARABLES como sensores de datos

▶ Apple VISION PRO



Responsive, precision eye tracking.

A high-performance eye-tracking system of LEDs and infrared cameras projects invisible light patterns onto each eye. This advanced system provides ultraprecise input without you needing to hold any controllers, so you can accurately select elements just by looking at them.

WEARABLES como sensores de datos

▶ Apple VISION PRO



WEARABLES como sensores de datos

▶ Apple VISION PRO



8-core CPU with 4 performance cores and 4 efficiency cores
10-core GPU
16-core Neural Engine
16GB unified memory



12-millisecond photon-to-photon latency
256GB/s memory bandwidth

Sensors

Two high-resolution main cameras
Six world-facing tracking cameras
Four eye-tracking cameras
TrueDepth camera
LiDAR Scanner
Four inertial measurement units (IMUs)
Flicker sensor
Ambient light sensor

Optic ID

Iris-based biometric authentication
Optic ID data is encrypted and accessible only to the Secure Enclave processor
Secures personal data within apps
Make purchases from the iTunes Store and App Store

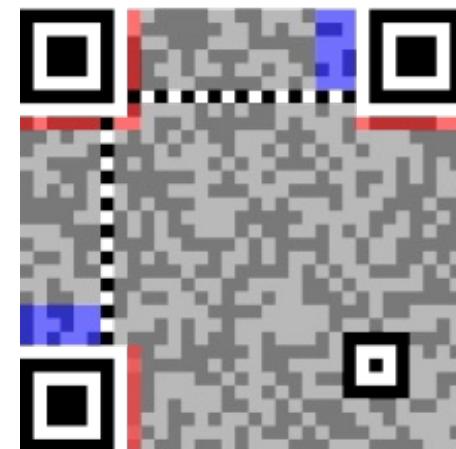
Código QR (Quick Response)

Un **código QR** es una matriz en dos dimensiones formada por una serie de cuadrados negros sobre fondo blanco. Esta matriz es leída por un lector específico (Lector de QR) en nuestro dispositivo móvil y de forma inmediata nos lleva a una aplicación en Internet ya sea un mapa de localización, un correo electrónico, una página web o un perfil en una red social.



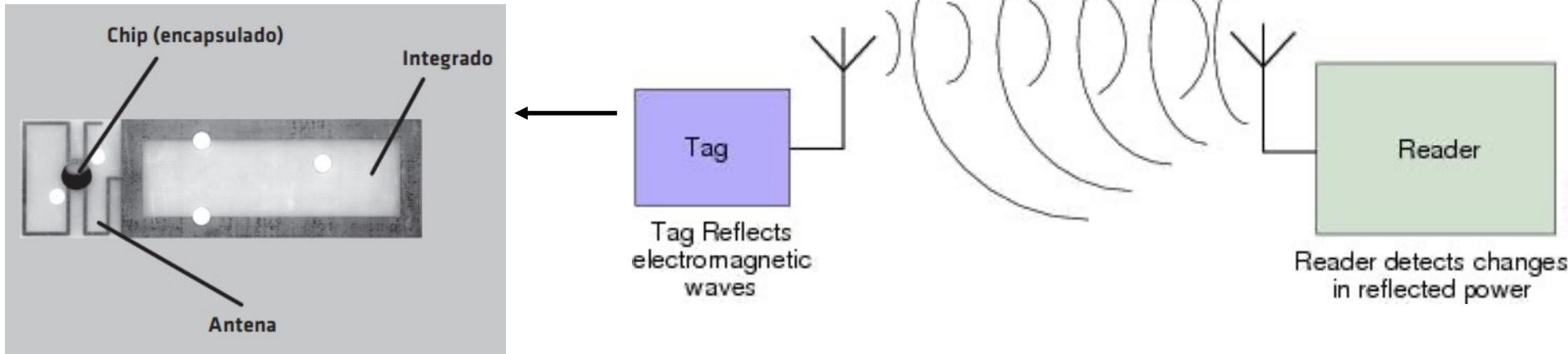
Código QR (Quick Response)

- Están compuestos por **tres cuadrados en las esquinas** que permiten detectar al lector la posición del código QR y una serie de cuadrados dispersos que codifican el alineamiento y la sincronización.
- Se pueden incorporar imágenes **personalizada** a nuestro código QR dando un aspecto más artístico y personal.
- Técnicamente son elementos complejos, pero como usuarios, simplemente debemos conocer que a través de un lector y escaneando el código a través de la cámara de nuestro Smartphone.
- **Son libres y cualquiera puede crear sus propios QR.**



- 1. Información de la Versión
- 2. Información del Formato
- 3. Corrección de Errores y Datos
- 4. Patrones Requeridos
 - 4.1. Posición
 - 4.2. Alineamiento
 - 4.3. Sincronización

Tecnología RFID



Los **tags** RFID constan de dos elementos básicos: un chip y una antena. El chip y la antena, montados, forman un integrado.

Tipos de tags:

RFID pasivo



- Pasivas
- Activas
- Semi-pasivas

RFID activo



Definición

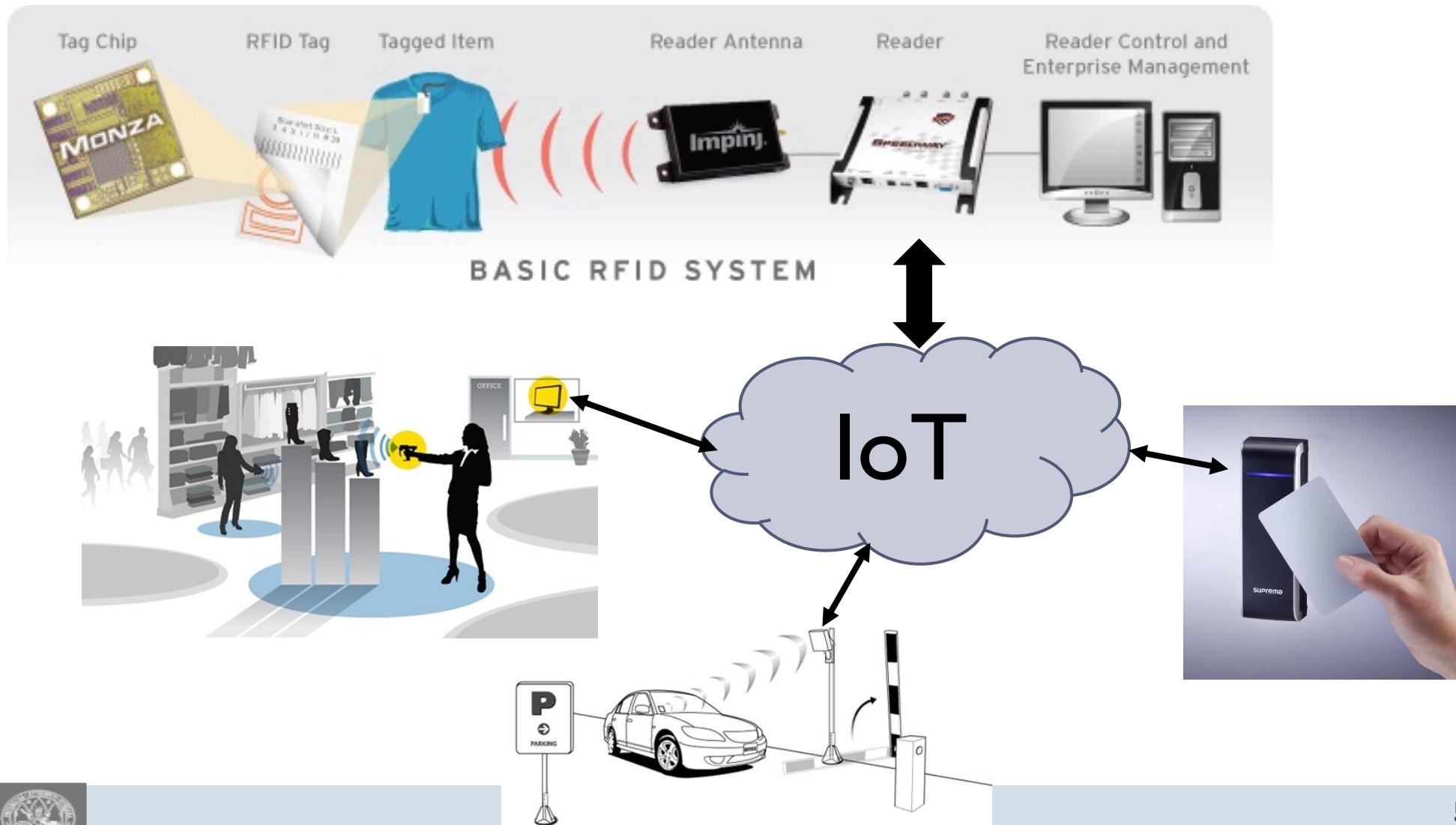
La identificación por radiofrecuencia o **RFID** (*Radio Frequency IDentification*), es una tecnología de identificación remota e inalámbrica en la cual un dispositivo lector o **reader**, se comunica a través de una antena con un **transponder** (también conocido como tag o etiqueta) mediante ondas de radio.

Tecnología RFID

- El **alcance de lectura** de un sistema RFID (distancia al tag a la que debe estar la antena del lector para leer los datos almacenados en el chip del tag) varía de unos cuantos centímetros a decenas de metros, en función de la frecuencia que se utilice, de la potencia y de la sensibilidad direccional de la antena.
- La tecnología HF se utiliza en las aplicaciones de corto alcance, ya que el alcance máximo de lectura es de unos tres metros.
- La tecnología UHF proporciona un alcance de lectura de 20 metros o más.
- El alcance también depende enormemente del entorno físico inmediato; la presencia de metales y líquidos puede causar interferencias que afecten al desempeño del alcance y de la lectura/escritura.
- **Seguridad.** Los chips RFID son difíciles de hackear.
- Hay numerosas **normas** que garantizan la diversidad de frecuencias y aplicaciones. La **Organización Internacional de Estándares (ISO)** y **EPCglobal Inc.** son dos de las organizaciones de normas más importantes para la cadena de suministros.



Tecnología RFID e IoT



ÍNDICE

BLOQUE I: Dispositivos y soluciones para recogida de información de sensores

BLOQUE II: Los dispositivos móviles como sensores de datos

BLOQUE III: Plataformas para gestión de la información



BLOQUE III - ÍNDICE

▶ Conceptos

- **Plataformas Cloud Computing**
- **Cloudlets, Edge Computing y Fog Computing**

▶ Plataformas IoT

- **Demo**

Plataforma en IoT

- Una plataforma es un conjunto de hardware y un software sobre la que otras aplicaciones pueden funcionar.
 - La **plataforma en la nube** consta de un hardware en el que reside un sistema operativo, y este sistema operativo permitirá que la aplicación funcione, proporcionando entorno de ejecución necesaria para ello
- Cuando se está desarrollando una aplicación, la plataforma permite desplegar y ejecutar la aplicación
- Las plataformas de aplicaciones IoT proporcionan un amplio conjunto **herramientas**, aplicaciones con funcionalidades independientes que se pueden utilizar para construir aplicaciones IoT



Cloud Computing: introducción

- Paradigma basado en el concepto de provisión dinámica y bajo demanda de:
 - Servicios de alto nivel
 - Computación
 - Almacenamiento
 - Servicios de red
 - Infraestructuras IT en general
- Recursos disponibles a través de una red de acceso (normalmente Internet)
- Recursos disponibles sólo cuando se necesitan

Cloud Computing: perspectiva del usuario

- Usuarios **no necesitan invertir grandes sumas de dinero** para tener acceso a complejas infraestructuras IT de altas capacidades: computación, almacenamiento, etc.
- **Accesible** fácilmente
- Usuarios **sólo pagan** por lo que realmente **necesitan**
 - Cantidad de recursos
 - Tiempo de uso
- Los usuarios pueden acceder a los servicios contratados de forma **ubicua**



Cloud Computing: perspectiva empresarial

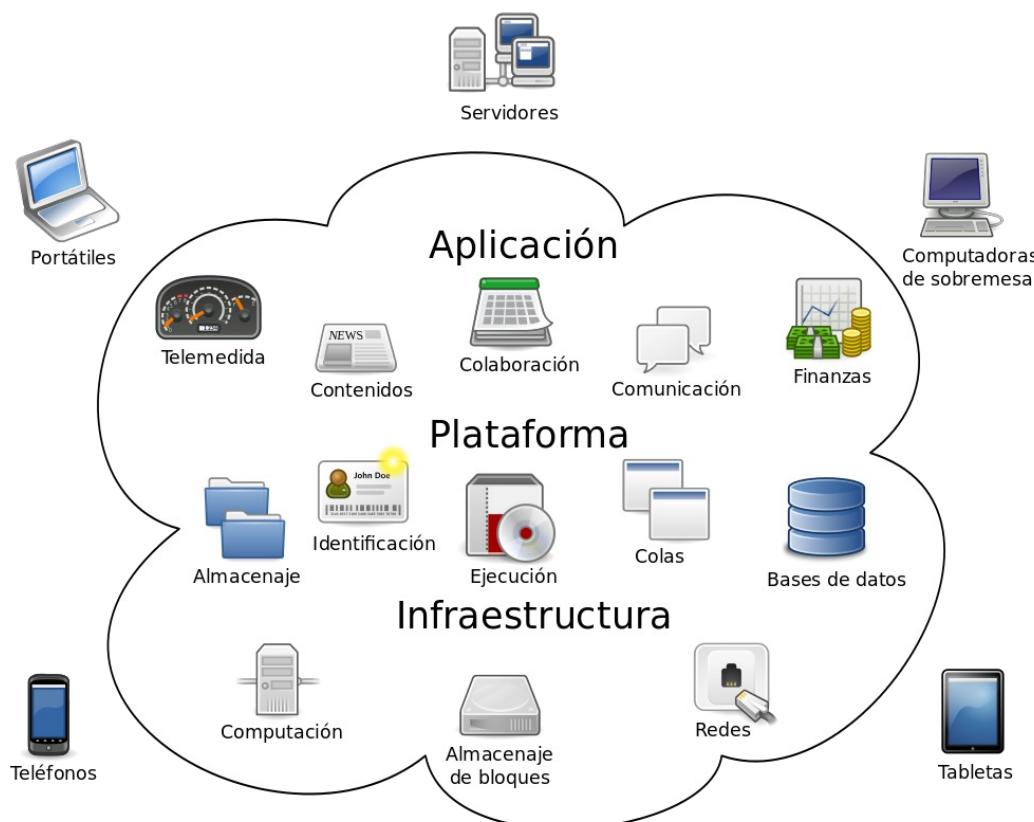
- **Empresas grandes** pueden **delegar** actividades no relacionadas con su actividad principal y centrarse en sus tareas de negocio
- Las **empresas pequeñas** pueden abordar económicoamente el paso desde el mundo de las ideas a su implementación, sin grandes inversiones económicas
- Los **desarrolladores** pueden centrarse en la lógica de negocio y en sus tareas de desarrollo, dejando de lado las tareas de despliegue y gestión de la arquitectura IT



Cloud Computing: definición

- Término asociado con la **virtualización** de infraestructuras y dispositivos
- La principal característica es su capacidad para proveer servicios de forma **flexible y dinámica**
- Estos servicios se suelen clasificar como:
 - IaaS (Infrastructure as a Service)
 - Virtualización de Hardware, computación o almacenamiento
 - PaaS (Platform as a Service)
 - Herramientas e interfaces para desarrollo Software
 - SaaS (Software as a Service)
 - Aplicaciones finales

Cloud Computing: definición



“Cloud Computing es un modelo para habilitar acceso conveniente por demanda a un conjunto compartido de **recursos computacionales configurables**, por ejemplo, redes, servidores, almacenamiento, aplicaciones y servicios, que pueden ser rápidamente aprovisionados y liberados con un esfuerzo mínimo de administración o de interacción con el proveedor de servicios. Este modelo de nube promueve la disponibilidad y está compuesto por **cinco características esenciales, tres modelos de servicio y cuatro modelos de despliegue**.”

National Institute of Standards and Technology (NIST)



Cloud Computing: características

- 1. Auto-servicio bajo demanda:** los usuarios consumen los servicios cuando los necesitan sin tener que comunicarse personalmente con el proveedor de servicios.
- 2. Escalabilidad y elasticidad:** los servicios pueden escalar rápidamente bajo demanda a través de añadir o eliminar recursos de cómputo.
- 3. Conjunto de recursos compartidos:** los recursos computacionales del proveedor se habilitan para servir a múltiples consumidores. Existe un sentido de independencia de ubicación en cuanto a que el consumidor no posee control o conocimiento sobre la ubicación exacta de los recursos que se le están proveyendo.
- 4. Acceso a través de Internet:** los servicios son entregados a través de internet utilizando mecanismos y protocolos estándar.
- 5. Modelo de pago por uso:** los servicios son monitoreados a través de métricas que permiten el establecimiento de diferentes modelos de pago.



Cloud Computing: pilares

- 1. Sistemas distribuidos:** permiten la gestión de grandes despliegues descentralizados, aumentando la escalabilidad de los sistemas
- 2. Virtualización:** concepto clave que ha permitido el desarrollo flexible de plataformas software sobre un hardware común por debajo:
 - Mejora en la utilización de los recursos hardware
 - Falta de espacio físico
 - Ahorro energético y de costes de gestión
 - Necesidad de arquitecturas flexibles
- 3. WEB 2.0:** habilita el fácil de acceso a estos servicios para los usuarios finales.
La mayoría de estos servicios puedes ser gestionados vía web

Cloud Computing: Virtualización

Virtualización de plataforma

- Esta involucra la simulación de **máquinas virtuales**. La virtualización de plataforma se lleva a cabo en una plataforma de hardware mediante un software "host" (en castellano anfitrión), que es un programa de control que simula un entorno computacional (máquina virtual) para su software "guest" (en castellano "invitado").
- Este software guest, que generalmente es un sistema operativo completo, corre como si estuviera instalado en una plataforma de hardware autónoma.
- Muchas máquinas virtuales son simuladas en una máquina física dada.
- Para que el sistema operativo guest funcione, la simulación debe ser lo suficientemente grande como para soportar todas las interfaces externas de los sistemas guest, las cuales se pueden incluir a los drivers de hardware.

Cloud Computing: Virtualización

- **Host:** anfitrión en un entorno virtualizado. Como ejemplo concreto podemos pensar en un equipo físico con un sistema operativo donde se crean máquinas virtuales.
- **Guest:** el recurso virtual acogido en el host. Típicamente la máquina virtual.
- **Hypervisor o VMM(Virtual Machine Monitor/Manager):** componente software donde reside (casi toda) la función de emulación y la gestión de las máquinas virtuales.
 - El VMM maneja, gestiona y arbitra los cuatro recursos principales de una computadora (CPU, Memoria, Red, Almacenamiento) y además podrá repartir dinámicamente dichos recursos entre todas las máquinas virtuales definidas en el computador central. De modo que nos permite tener varios ordenadores virtuales ejecutándose sobre el mismo ordenador físico.

Cloud Computing: Virtualización

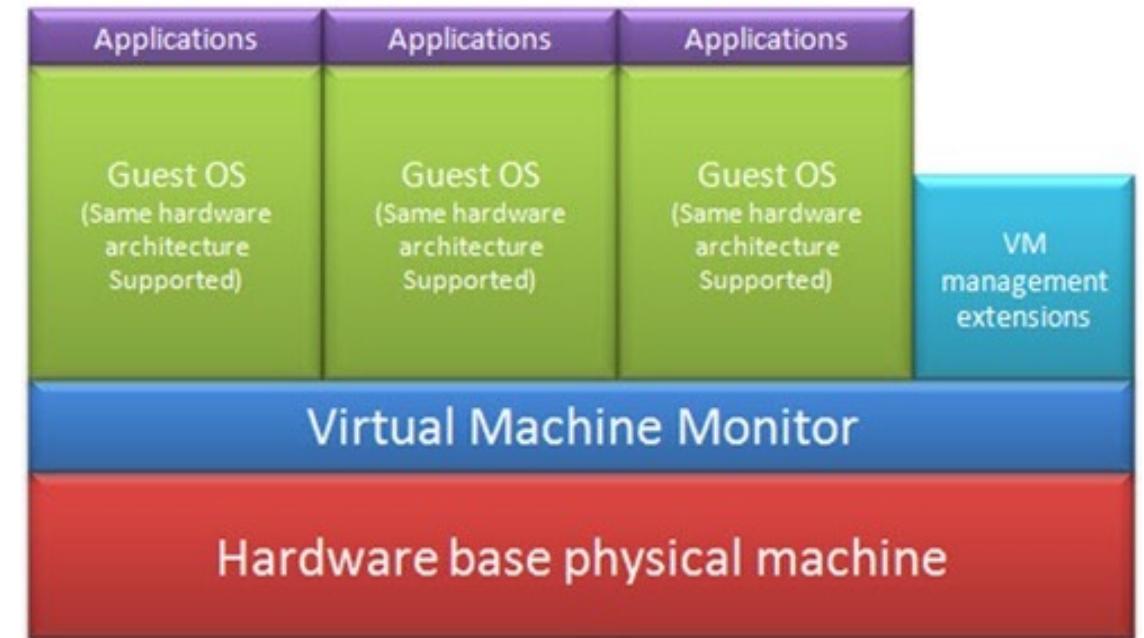
Tipos de virtualización

- **Full virtualization** se considera que la simulación del hardware es nativa/completa, tanto que permite que el guest ejecute el software sin modificación alguna, principalmente el sistema operativo.
 - Binary rewriting: cuando no hay soporte hardware en el procesador físico.
 - Hardware-assisted Virtual Machine (HVM): con soporte hardware en el procesador.
- **Paravirtualization** el hardware no es simulado, sino que el anfitrión tiene los medios para generar entornos aislados en los que se ejecutan los sistemas guest. El sistema operativo del guest tiene el kernel modificado.
- **Operating-system-level virtualization** - el kernel del sistema operativo host tiene mecanismos para crear contenedores del mismo tipo de sistema operativo.



Virtualización

- ▶ Tipos de virtualización
 - ▶ Virtualización nativa/completa
 - ▶ Proporciona una capa de adaptación a las máquinas invitadas (máquinas virtuales)
 - ▶ SO anfitrión, SO invitado y monitor de máquinas virtuales (VMM) o hipervisor
 - ▶ Permite ejecutar distintas versiones de SSOO (Windows Server 2003, Windows Server 2008, Linux, etc.)
 - ▶ Los ficheros de las imágenes son muy grandes



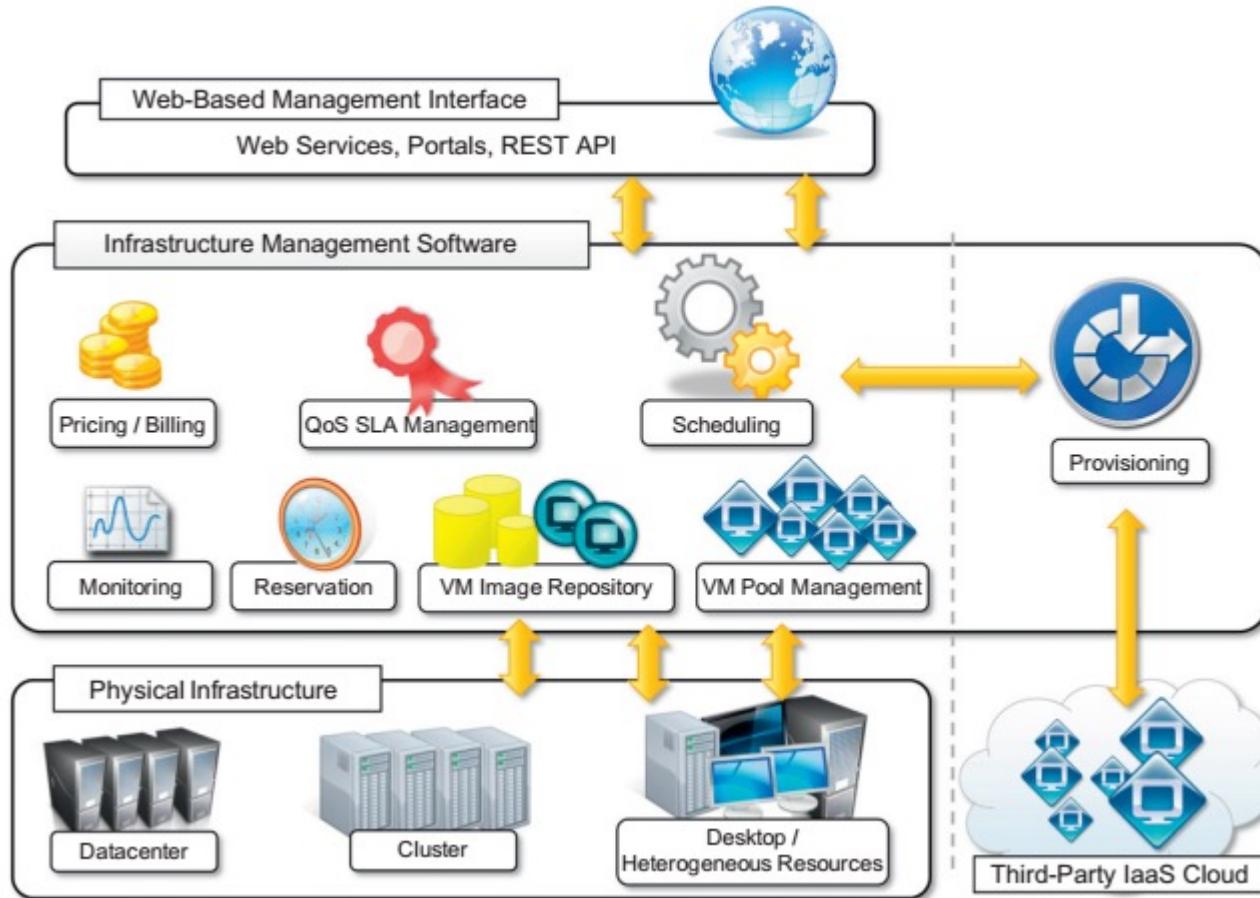
Cloud Computing: modelos de servicio

IaaS (**Infraestructura como Servicio**), proporciona la infraestructura informática, física u otros recursos como máquinas virtuales, imágenes de disco, bloque y el almacenamiento basado en archivos, firewalls, balanceadores de carga, direcciones IP, redes de área local virtuales, etc.

- El proveedor de servicios ofrece una infraestructura escalable y compartida a diferentes clientes.
- Los clientes pueden obtener servicios de infraestructura computacional bajo demanda.
- El cliente puede ajustar la insfraestructura a sus necesidades.
- Ejemplos de uso: web hosting, computación de alto rendimiento, infraestructura de pruebas, desarrollo y producción, almacenamiento en la nube.



Cloud Computing: IaaS



Ejemplos

Amazon EC2
OpenSource: OpenNebula,
OpenStack, VMware cloud,
etc.

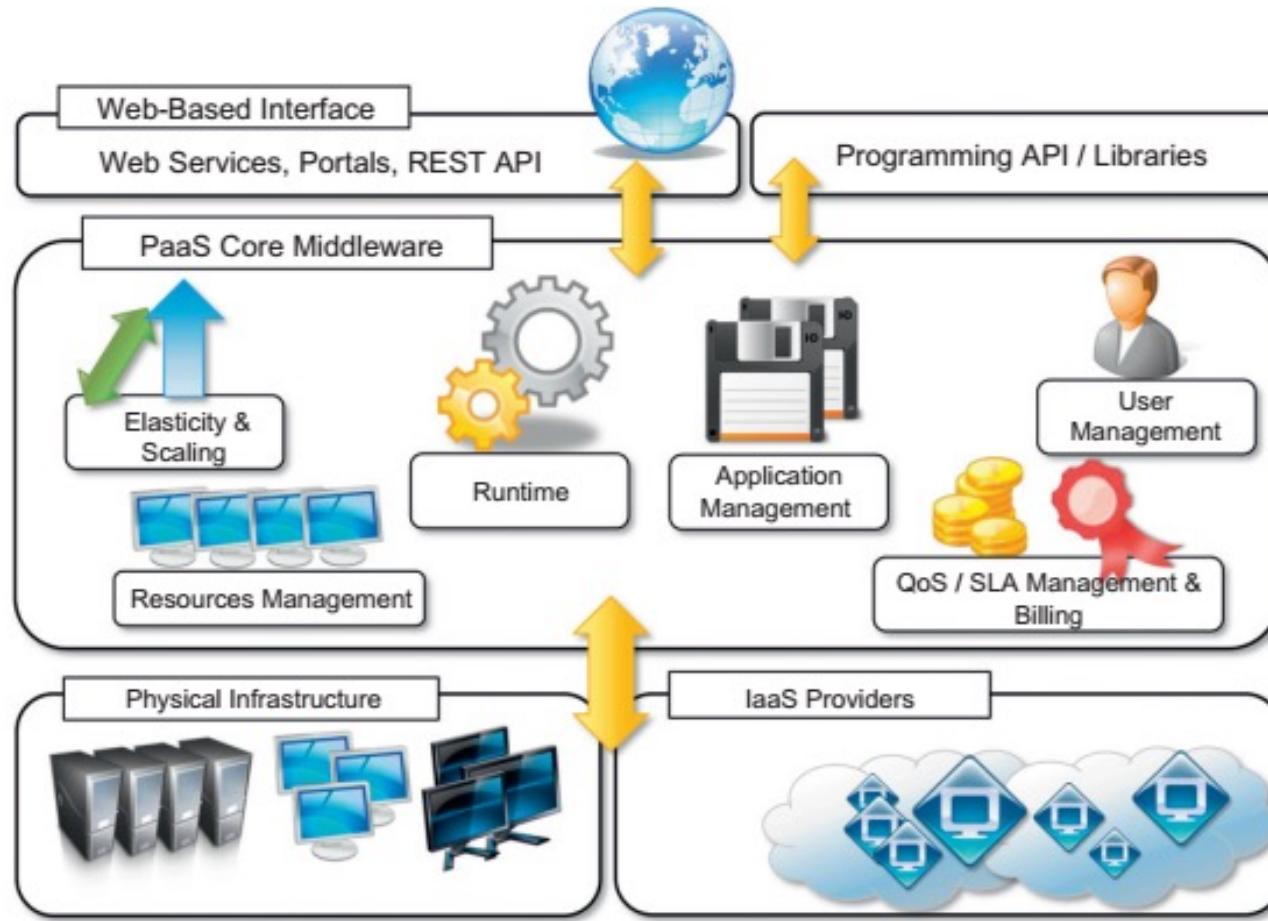
Cloud Computing: modelos de servicio

PaaS (Plataforma como Servicio). Proporciona las plataformas de cómputo que normalmente incluye sistema operativo, lenguaje de programación del entorno de ejecución, base de datos, servidor web, etc. En este caso los desarrolladores se encargan de desarrollar las propias aplicaciones que se ejecutan en la nube.

- El proveedor de servicios ofrece el middleware requerido para el desarrollo y la ejecución de aplicaciones.
- No hay que preocuparse de la infraestructura.
- Se proveen APIs y Frameworks para el desarrollo de sistemas
- Ejemplos de uso: plataformas de desarrollo en la nube, entornos de programación y compilación on-line



Cloud Computing: PaaS



Ejemplos:

- Google AppEngine
- Microsoft Azure
- IBM Cloud (Bluemix)

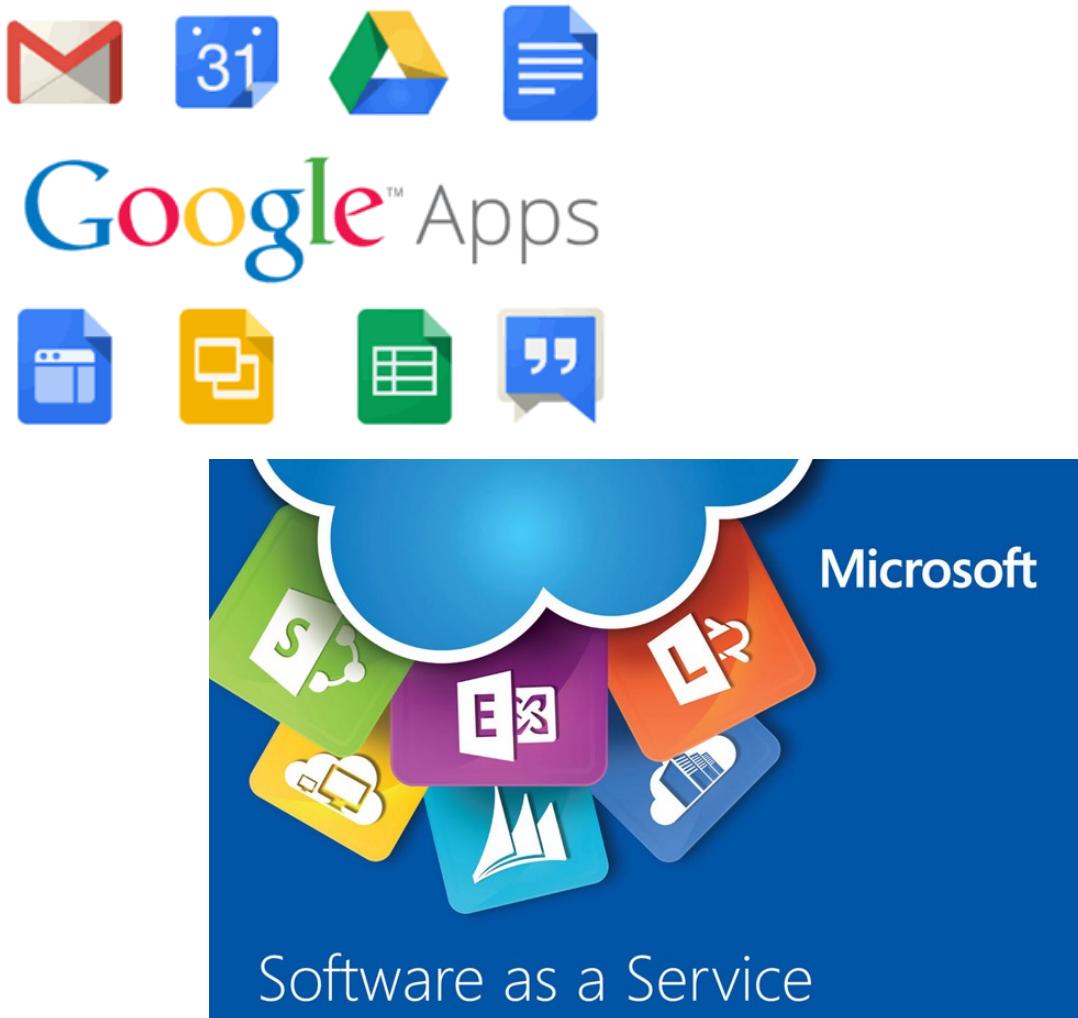
Cloud Computing: modelos de servicio

SaaS (Software como Servicio). Se trata de cualquier servicio basado en la web. En este tipo de servicios se accede a través del navegador sin atender al software. Todo el desarrollo, mantenimiento, actualizaciones, copias de seguridad es responsabilidad del proveedor. El usuario paga por el uso, por la infraestructura necesaria.

- Software es desarrollado y gestionado remotamente por un proveedor de servicios.
- Todo está en la nube → Los clientes no tienen que realizar operaciones en su infraestructura.
- El pago se realiza por suscripción (mensual, anual, etc.) o a través de métricas de uso.
- Ejemplos: redes sociales, editores on-line, aplicaciones científicas, etc.

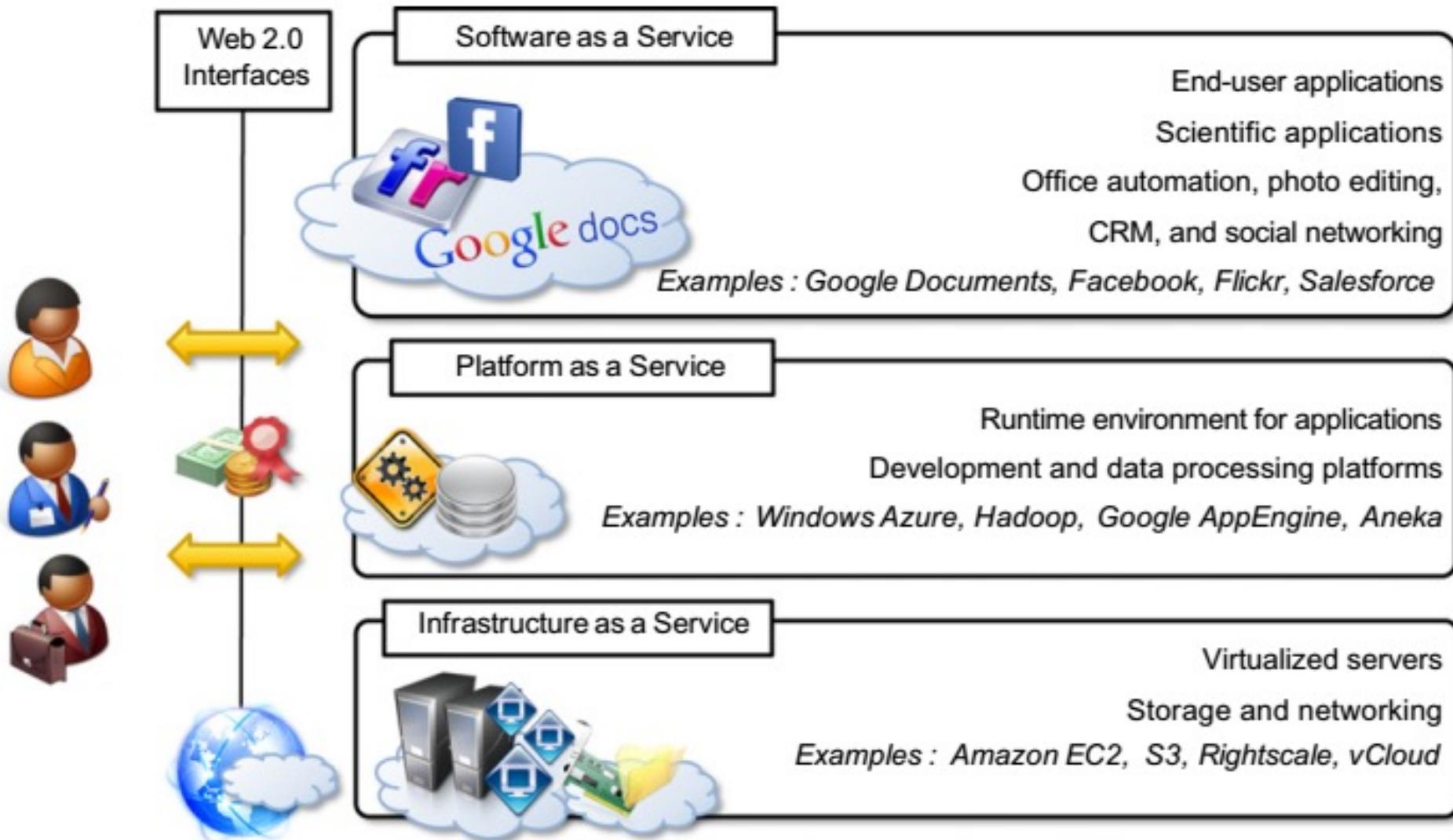


Cloud Computing: SaaS

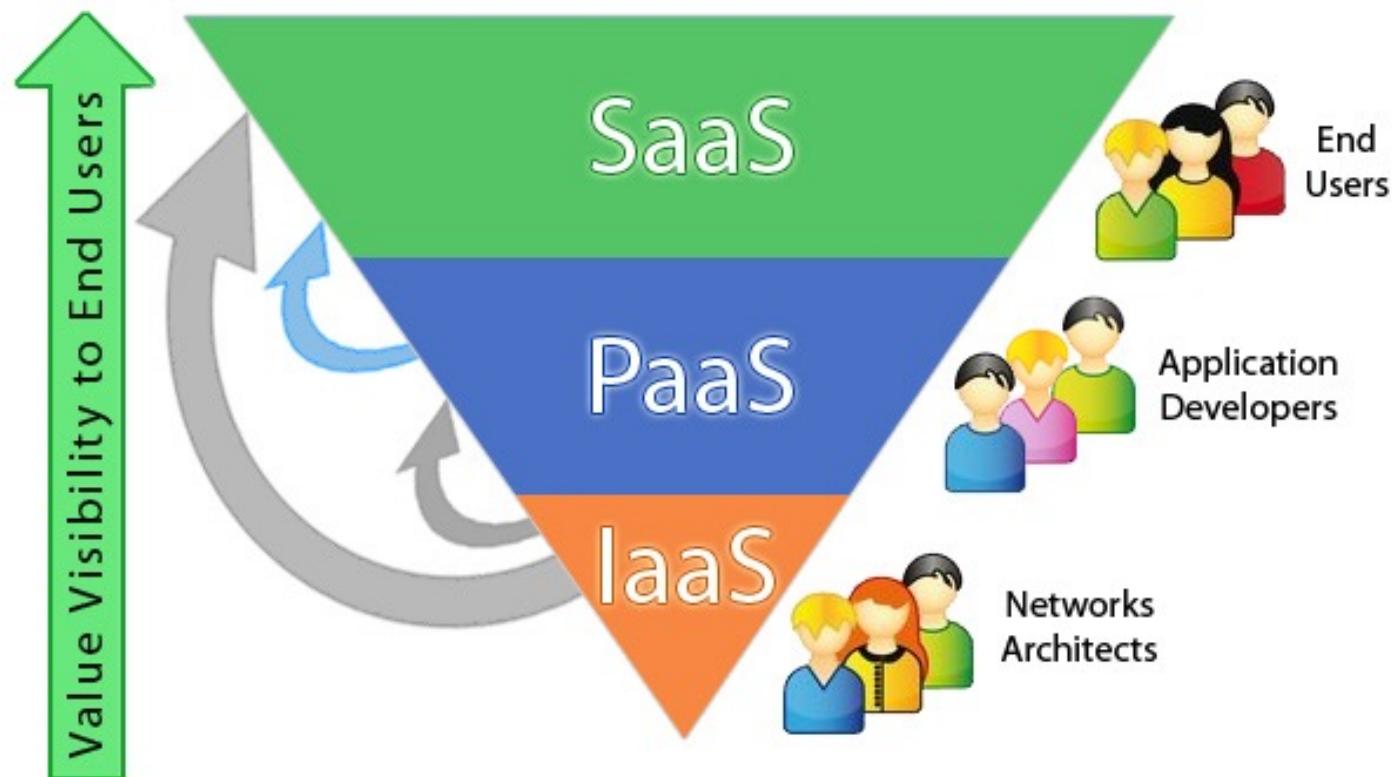


Ejemplos:
Google apps
Web-mail
Overleaf
Office 365

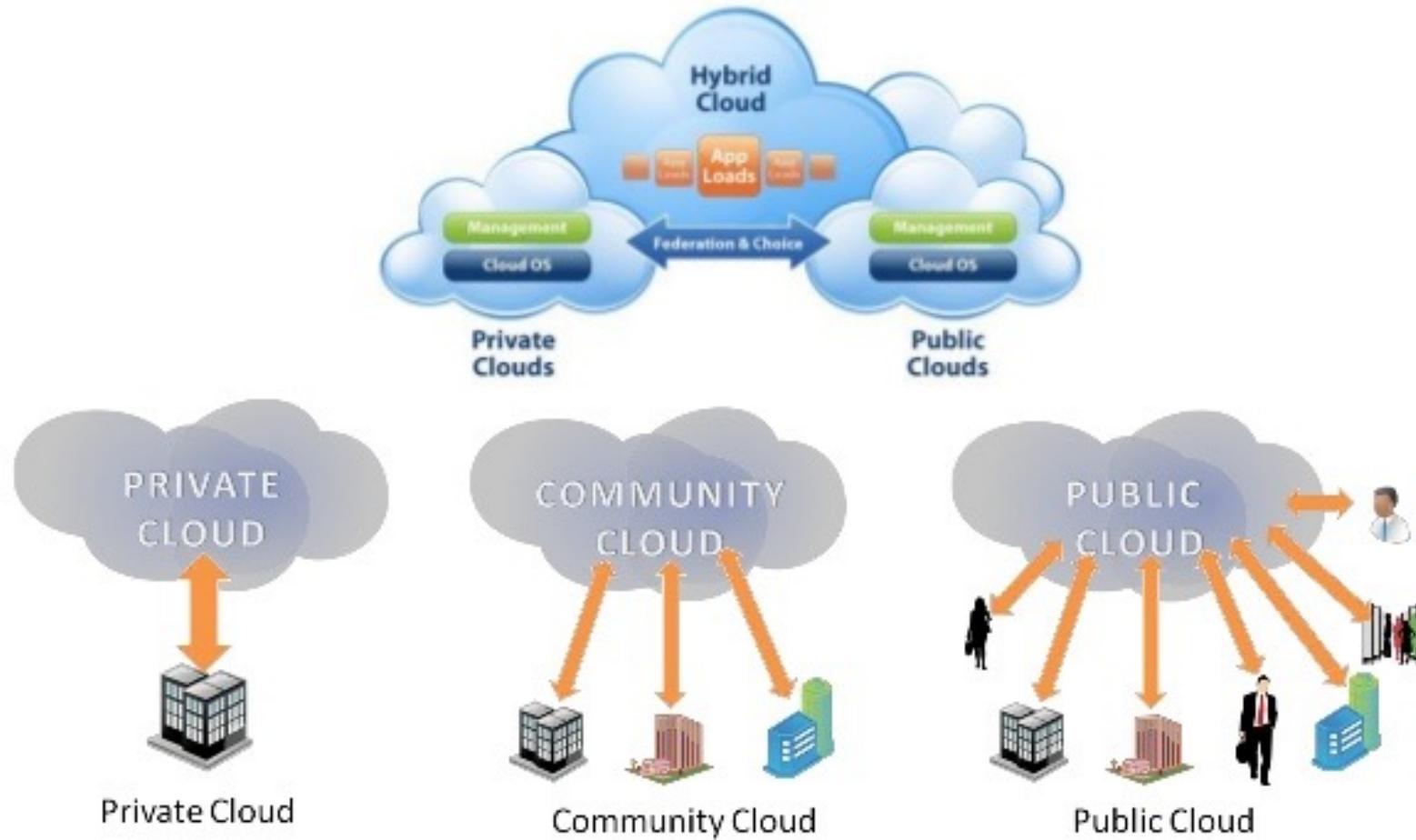
Cloud Computing: definición



Cloud Computing: modelos de servicio

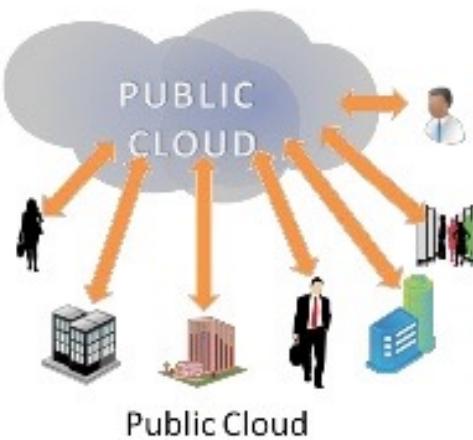


Cloud Computing: modelos de despliegue

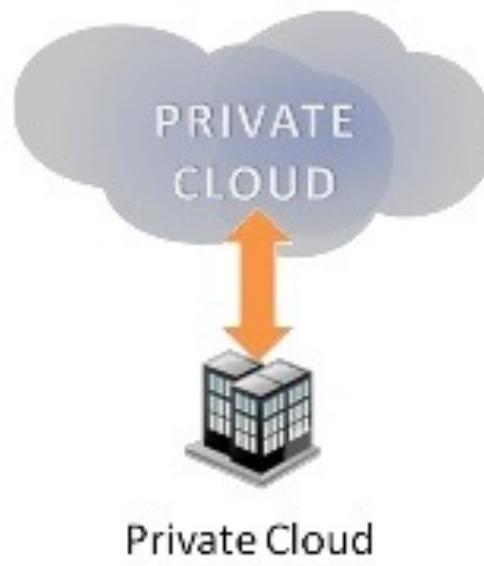


Cloud Computing: modelos de despliegue

1. En el modelo de **nube pública**, en general, los proveedores tienen grandes infraestructuras para soportar un gran número de usuarios.
2. Los servicios están **disponible al público** en cualquier momento.
3. Cualquier organización puede hacer uso de los servicios de forma **rápida** (bajo coste de configuración inicial).
4. Formadas por multitud de servidores geográficamente distribuidos
5. **Multi-usuario:** necesario el aislamiento de recursos



Cloud Computing: modelos de despliegue



1. Las redes públicas presentan un problema: falta de control o privacidad de los datos almacenados: “no todo el mundo quiere que Dropbox maneje sus datos personales”. Gobiernos, información sensible...
2. En el modelo de **nube privada** la infraestructura es operada por una sola organización.
3. Los servicios están disponibles a **usuarios internos** de la organización.
4. Sólo departamentos internos pueden hacer uso de los servicios de una manera rápida (elevado coste de configuración inicial).
5. Necesidad de desarrollar estándares de utilización

Cloud Computing: modelos de despliegue



1. En el modelo de **nube comunitaria** la infraestructura soporta los requerimientos de varias organizaciones con un objetivo común.
2. Los servicios están disponibles a los **usuarios de las organizaciones participantes**.
3. Sólo las organizaciones participantes pueden hacer uso de los servicios de una manera rápida (coste medio de configuración inicial).
4. Las organizaciones participantes pueden aprovechar los recursos de las demás.

Cloud Computing: modelos de despliegue

1. La **nube híbrida** es una combinación de los modelos anteriores.
2. En este modelo la infraestructura es operada por organizaciones externas (Public Cloud) e internas (Private Cloud).
3. Las organizaciones pueden mantener su infraestructura privada y en caso de requerir más recursos pueden utilizar infraestructuras públicas.
4. Los datos sensibles pueden mantenerse en la red privada.



Cloud Computing: ventajas

- ✓ **Bajo coste.** No hay necesidad de adquirir un hardware y un software determinado, lo que reduce costos operativos en infraestructura, mantenimiento y energía. La nube es más barata que la instalación y mantenimiento de un servidor propio o contratar los servicios de un proveedor.
- ✓ **Bajos requerimientos en los dispositivos de acceso,** ya que se accede a los servicios a través de interfaz web.
- ✓ **Mejora del rendimiento.** Los ordenadores que utilicen aplicaciones en la nube estarán menos sobrecargados porque tienen un menor procesado a nivel local.
- ✓ **Flexibilidad:** El servicio de nube se paga de acuerdo a la demanda.
- ✓ **Movilidad:** La información al quedar alojada en la nube pueden ser consultada por el usuario desde cualquier lugar

Cloud Computing: ventajas

- ✓ Mejor utilización de recursos hardware. Los elementos que forman la nube son explotados de forma más exhaustiva. **Se reduce el consumo energético** al haber menos equipamiento hardware por separado
- ✓ Menor tiempo de despliegue de aplicaciones, por las herramientas facilitadas en la nube
- ✓ Facilidad para agregar y eliminar servicios, al estar todos éstos virtualizados
- ✓ Acceso a toda la información en tiempo real
- ✓ Las actualizaciones de software son instantáneas y transparentes al usuario final
- ✓ Capacidad de almacenamiento casi ilimitada



Cloud Computing: desventajas

- ✓ **Recursos compartidos:** ¿están siempre disponibles? → QoS (Quality of Service)
- ✓ **Falta de estándares** entre plataformas: dificultad de migración
- ✓ **Software limitado** al ofrecido por la plataforma
- ✓ Muy **dependientes** del acceso a la **red**
- ✓ Riesgos de seguridad y privacidad de los datos
- ✓ **Altas latencias** en la recuperación de los datos
- ✓ **Servicio centralizado** en la nube



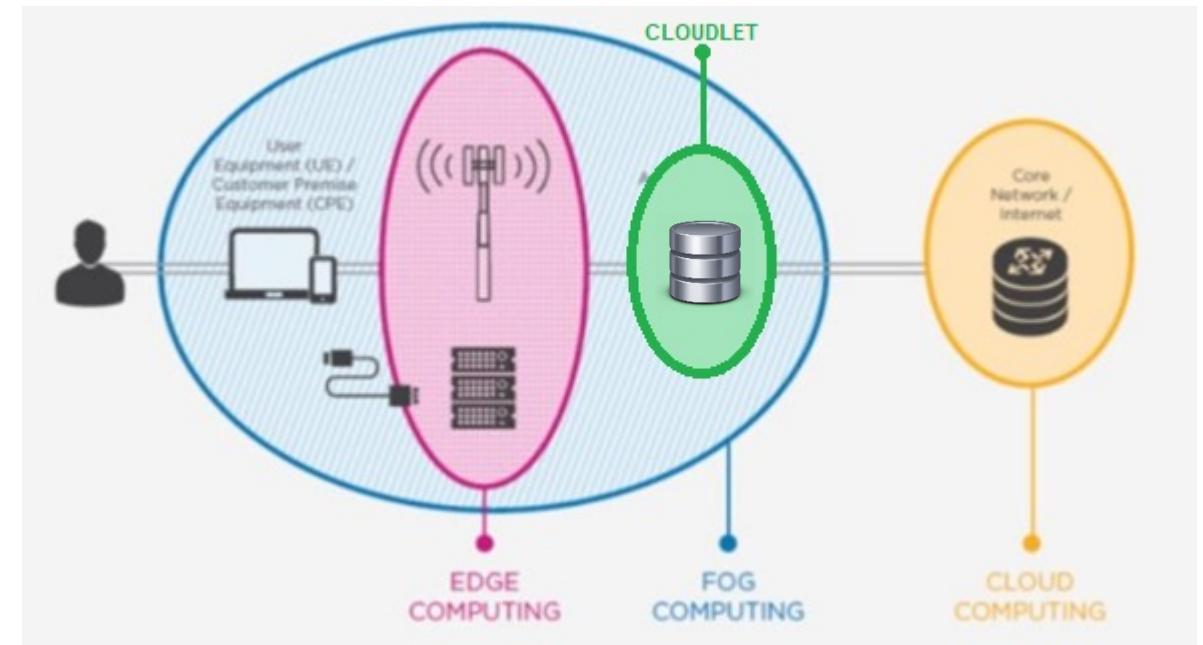
Cloud Computing: alternativas

- Nuevas aplicaciones con novedosos requerimientos:
 - Baja latencia
 - Alto ancho de banda
 - Servicios contextualizados y dependientes de la localización
- Los servicios cloud presentan problemas para manejar estos requerimientos:
 - Camino muy largo desde el usuario final hasta los servidores (**alta latencia**)
 - Posible colapso del backbone
 - Dificultad para proveer servicios contextualizados por el gran volumen y precisión de la información necesaria para esta contextualización



Cloud Computing: alternativas

- Tres alternativas:
 - Cloudlets
 - Mobile Edge Computing (MEC)
 - Fog Computing
- Aproximan la computación y almacenamiento al usuario

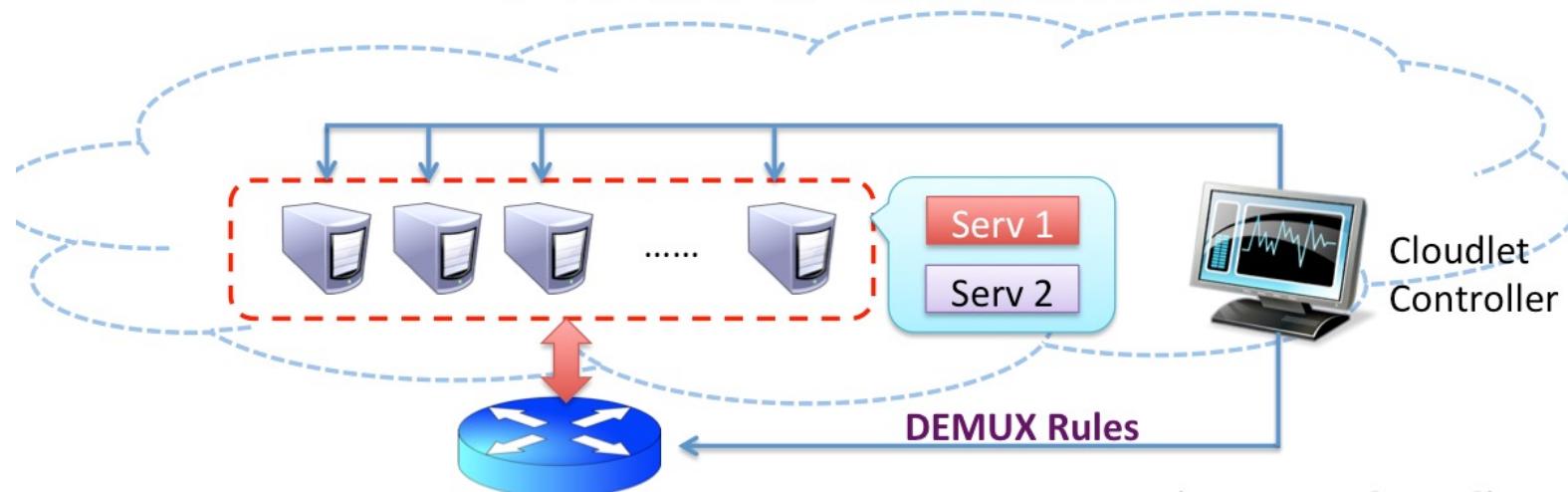


Cloudlets

- Término acuñado por la Carnegie Mellon University (USA)
- Propone instalar *datacenters* dedicados más próximos al extremo final de la red
- Sitúa una capa de procesado entre los dispositivos finales y el cloud: requiere de hardware y software específico: por ejemplo OpenStack++
- No ha tenido gran repercusión por la complejidad del despliegue

Cloudlets

Inside a Cloudlet



Resource Table (time slot: [t0, t1])

	CPU (cores)	Mem (GB)	Disk (GB)	BW (Gbps)
N ₁	7/1	1/1	250/50	5/5
N ₂	4/4	0/2	50/200	9/1
...

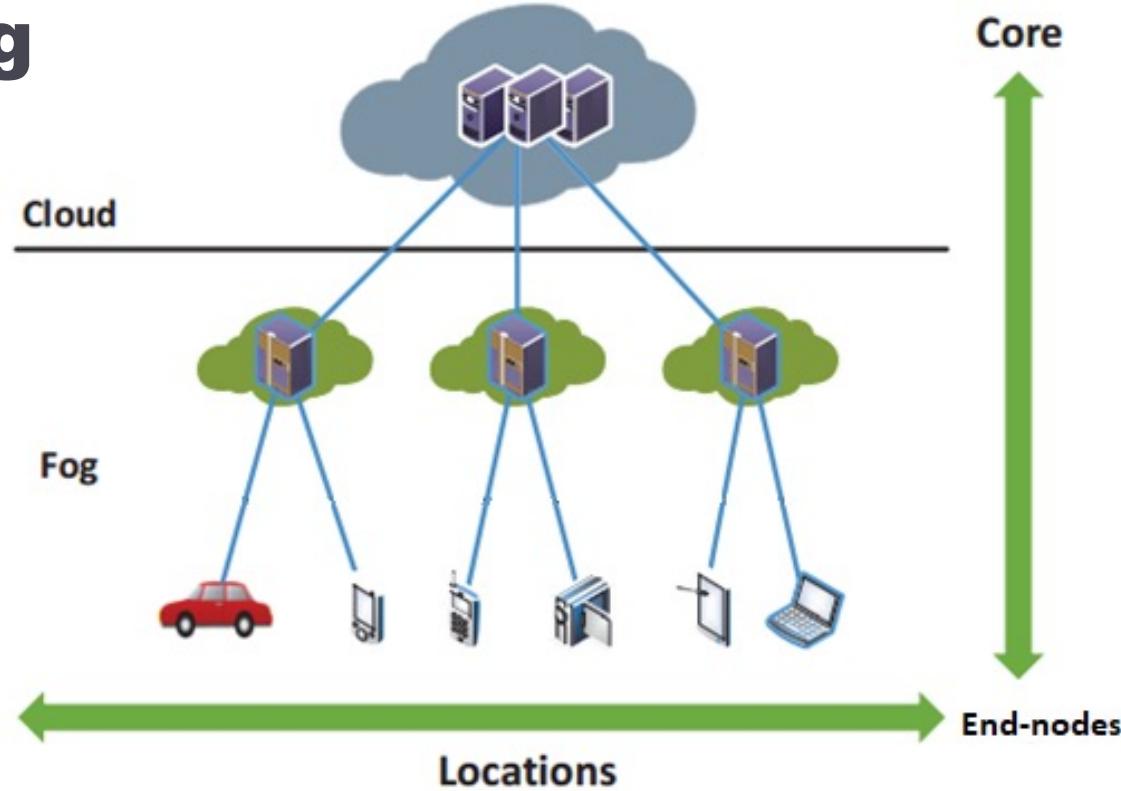
Reserved / Available

Fog Computing

- Término acuñado por Cisco en 2011 que propone situar la nube más cerca de los usuarios finales
- En un principio, el fog se situaba en los dispositivos finales (móviles, portátiles, dispositivos IoT con capacidad de procesado, etc.)
- Actualmente, el fog se entiende como toda la infraestructura comprendida entre los dispositivos finales y la red troncal



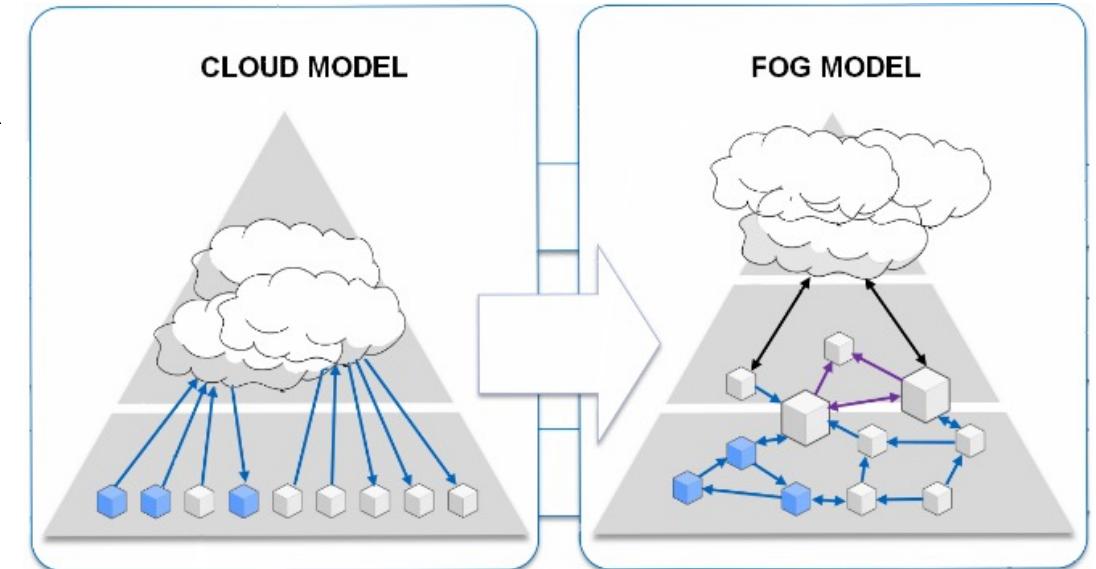
Fog Computing



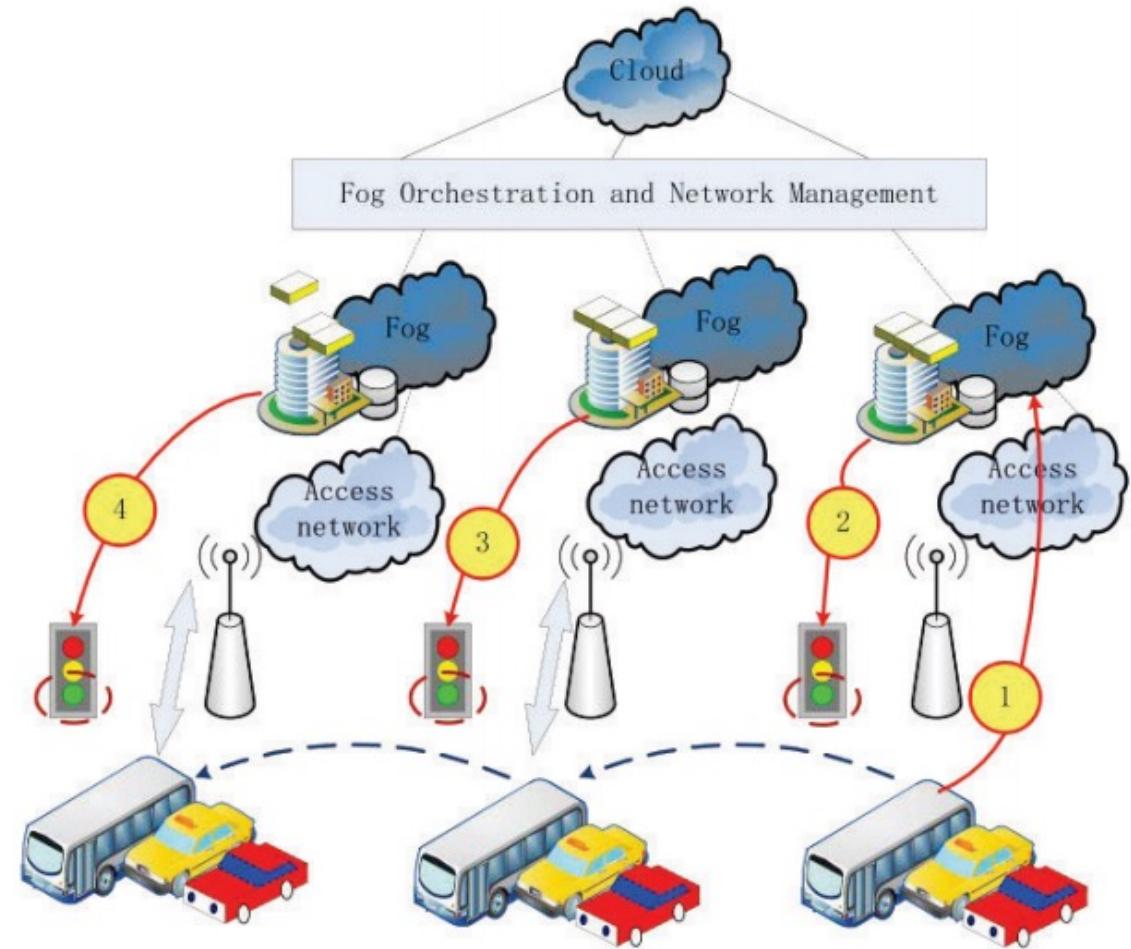
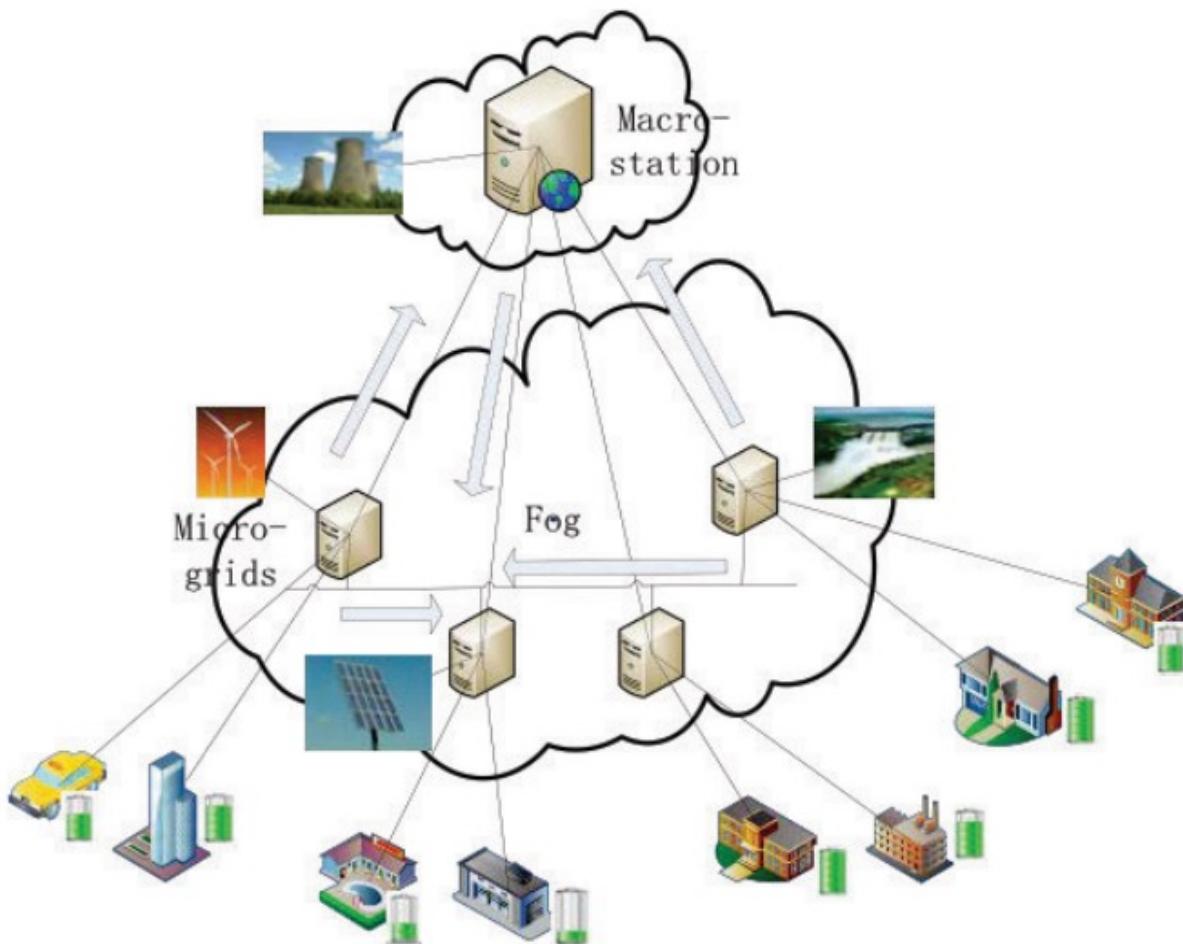
Modelo en el que los datos, procesamiento y aplicaciones se concentran en los dispositivos desde el extremo final de la red (dispositivos de usuario) y el equipamiento de red hasta el backbone con capacidad de procesado, en lugar de existir casi en su totalidad en la nube. Es una forma de computación distribuida de proximidad, donde cada uno de los dispositivos conectados a la red puede procesar los datos y sólo transmitir un resumen al siguiente nivel, o hacerlo sólo en determinados casos.

Fog Computing: características

- ✓ Baja latencia y conciencia de la ubicación
- ✓ Amplia distribución geográfica
- ✓ Movilidad
- ✓ Alta cantidad de nodos
- ✓ Rol predominante de acceso inalámbrico
- ✓ Fuerte presencia de streaming y aplicaciones en tiempo real
- ✓ Heterogeneidad

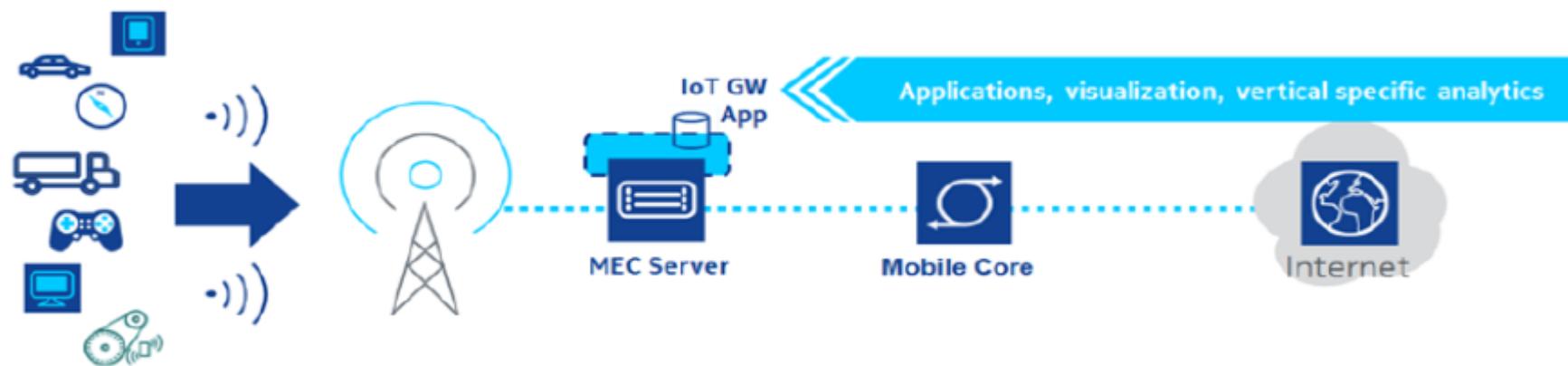


Fog Computing: ejemplos



Mobile Edge Computing

- Propuesto por el **ETSI** (European Telecommunications Standards Institute)
- Los elementos de procesado y/o almacenamiento se sitúan exclusivamente en las estaciones base o gateways que dan cobertura a los dispositivos finales
- Ligado al desarrollo de 5G, por la baja latencia necesaria (<1ms)

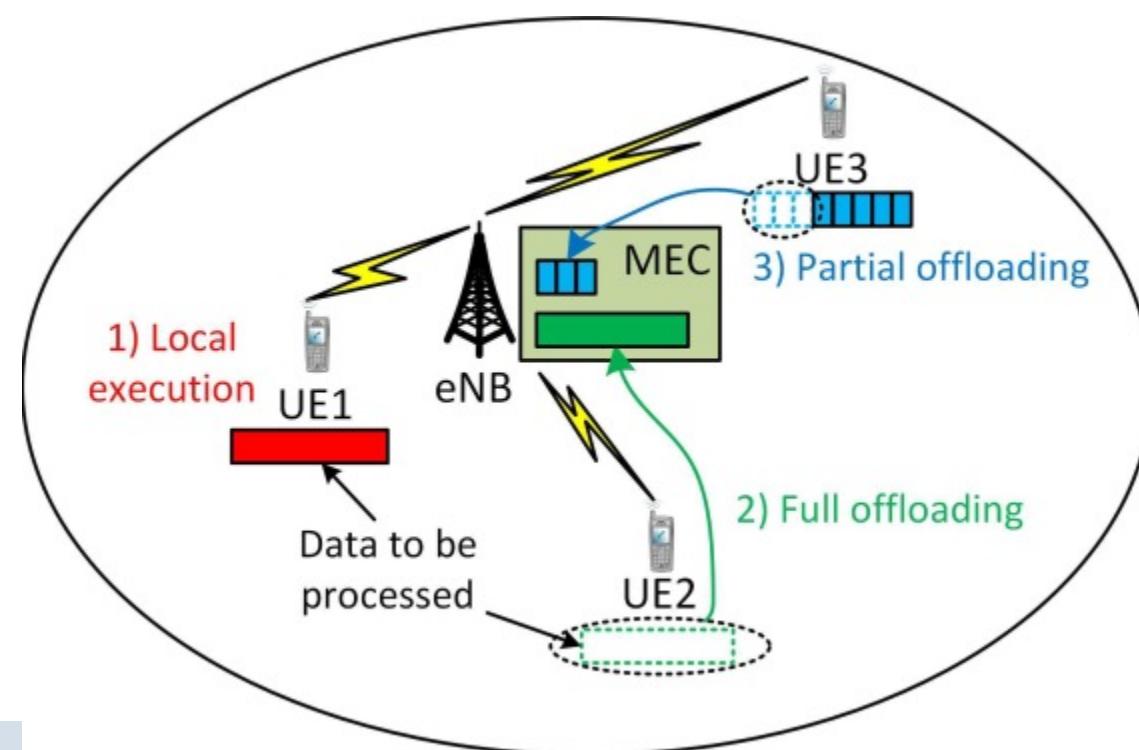


Mobile Edge Computing

- Permite tener rápido **acceso tanto a métricas** de red, como a métricas físicas de los enlaces
- Características de funcionamiento similares a la del fog computing pero presenta una menor heterogeneidad de los dispositivos involucrados
- Al tener acceso a las métricas de nivel físico de los nodos conectados a la estación base, se pueden tomar decisiones en configuración de las Comunicaciones
- Otra acepción: Multi-Access Edge Computing
 - En 5G se consideran múltiples tecnologías de acceso simultáneas
 - Es necesario decidir cual es la más apropiada:
 - Cobertura
 - Tipo de servicio

Mobile Edge Computing: ejemplos

- En IoT, los dispositivos finales están muy restringidos en su capacidad de cómputo:
 - Descarga de tareas de alta carga computacional (**computing offloading**)

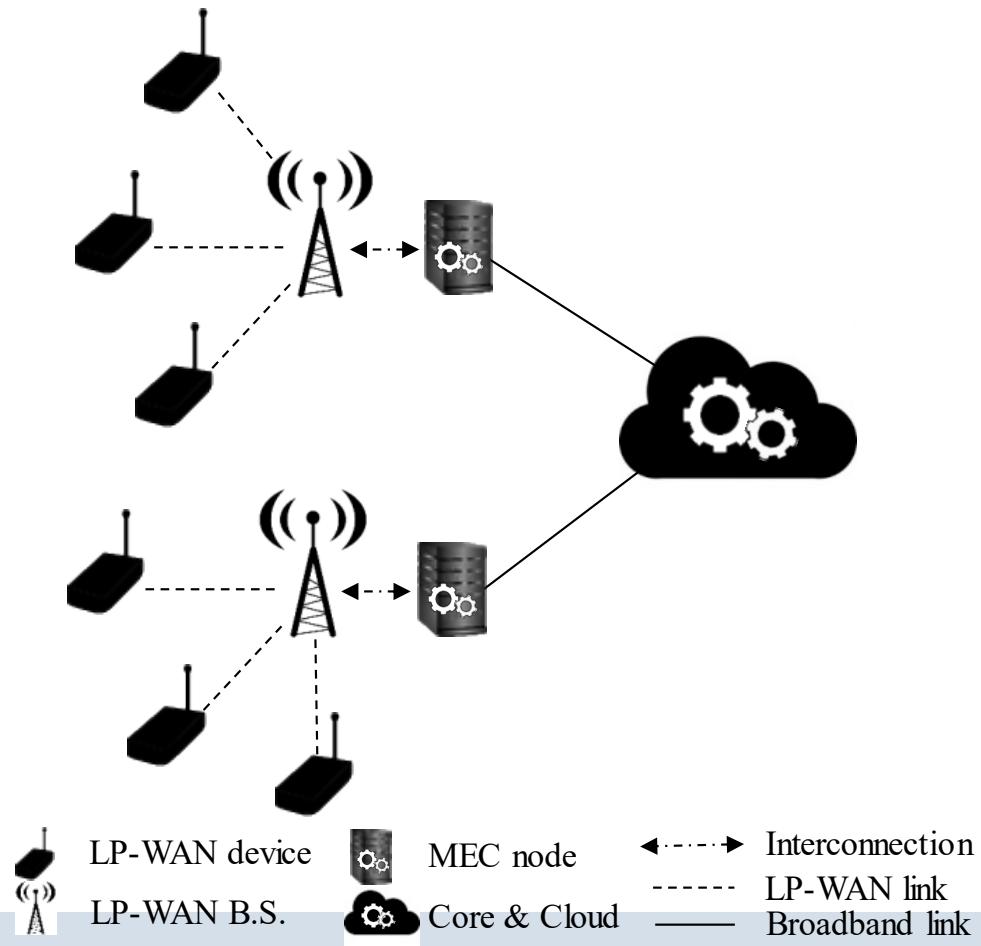


Fog & MEC vs Cloud

FOG & MEC	CLOUD
Almacenamiento y procesamiento local	Almacenamiento y procesamiento lejano
Servicios alojados en los dispositivos cercanos al borde de la red	Los servicios se alojan en servidores virtuales
Compatible con aplicaciones de IoT que demandan tiempo real o latencia predecible	Compatible con los servidores de hardware, aplicaciones y casi cualquier tipo de datos
Se basa en el principio de aislamiento de datos del usuario que se alojan en el borde de la red	Todos los datos se centralizan en uno o más centros de datos

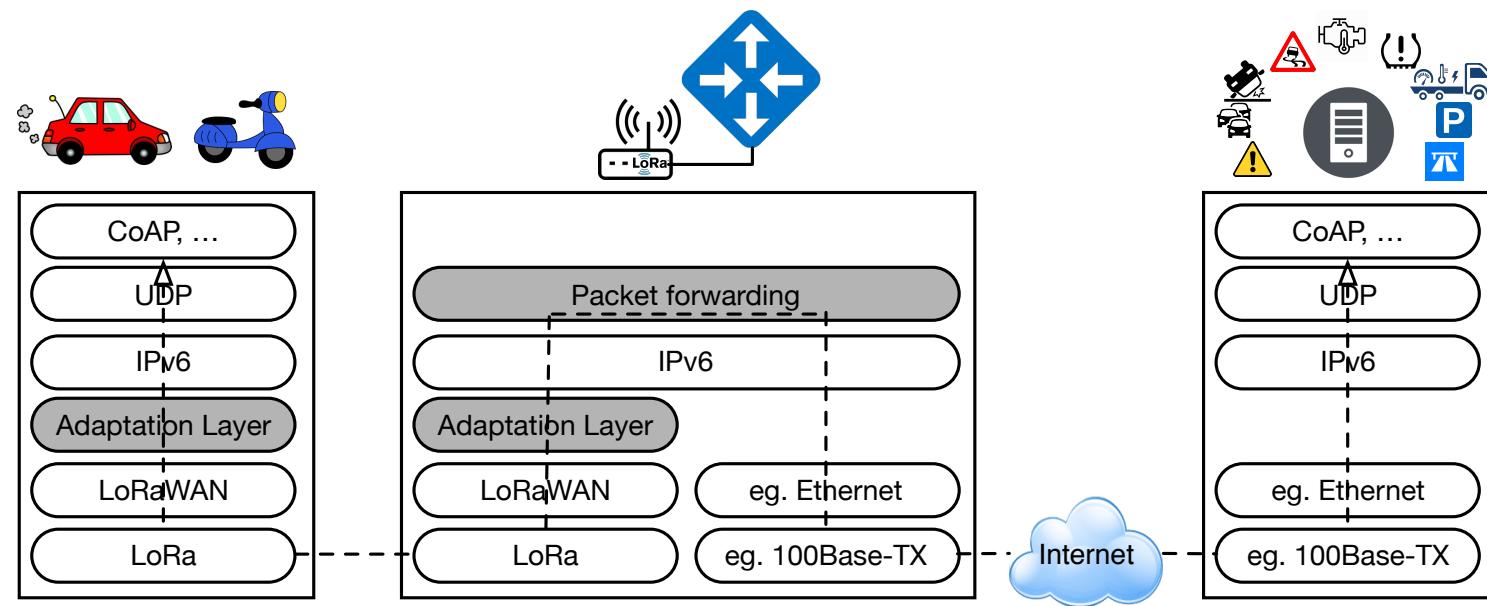
Caso de uso: integrar el paradigma MEC en redes LP-WAN

- Dispositivos LP-WAN muy restringidos tanto en su capacidad de cómputo como de transmisión de datos



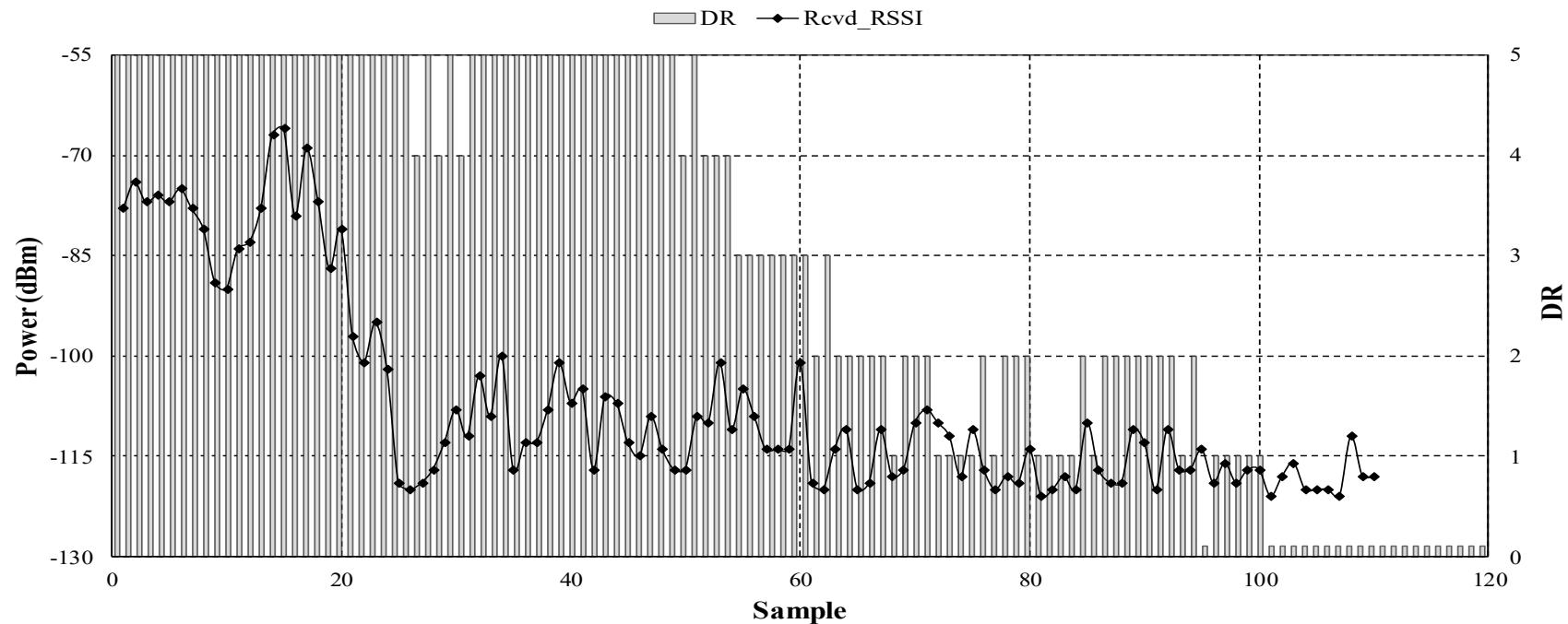
Caso de uso: integrar el paradigma MEC en redes LP-WAN

- El nodo MEC amplía la funcionalidad de la red:
 - Integración de IPv6 en el sistema



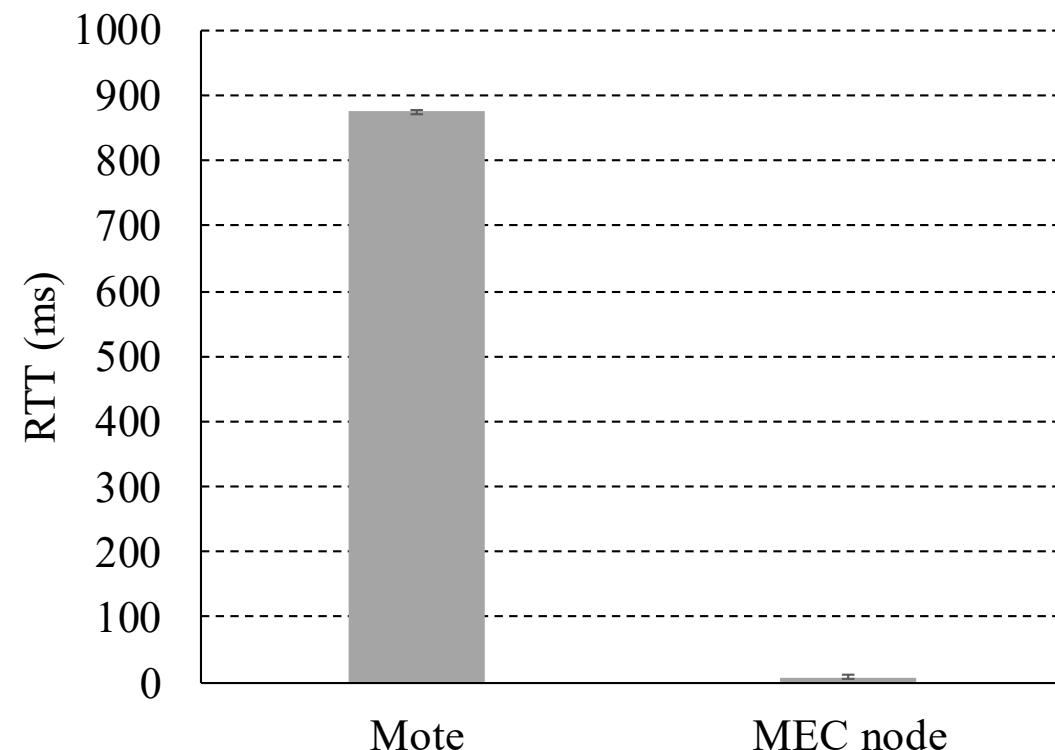
Caso de uso: integrar el paradigma MEC en redes LP-WAN

- El nodo MEC amplía la funcionalidad de la red:
 - Gestión adaptativa de los parámetros de modulación LoRaWAN (nivel físico)



Caso de uso: integrar el paradigma MEC en redes LP-WAN

- El nodo MEC amplía la funcionalidad de la red:
 - Data caching. Cuando se realiza una petición a un nodo final, el MEC evalúa si puede servir esa información (si es temporalmente válida) y la sirve:



BLOQUE I - ÍNDICE

Conceptos

- **Plataformas Cloud Computing**
- **Cloud Computing, Edge Computing y Fog Computing**

Plataformas IoT

- **Demo**

Plataformas IoT



Google Cloud Platform



Microsoft Azure



IBM Cloud



ThingSpeak



zetta

• • •

Plataformas IoT

Fuente: Bhumi Nakhuva and Prof. Tushar Champaneria, Department of Computer Science and Engineering, L. D. College of Engineering, India.
International Journal of Computer Science & Engineering Survey (IJCSES) Vol.6, No.6, December 2015

	Google	IBM	ThingWorx	Azure	Thing Speak	Zetta	Yaler	Amazon	Axeda
Escalabilidad	✓	✓	✓	✓	✓	✗	✓	✓	✓
Seguridad y privacidad	✓	✓	✓	✓	✗	✗	✗	✓	✓
Plug&Play	✓	✓	✓	✓	✗	✗	✗	✓	✓
Soporte millones de dispositivos	✓	✓	✓	✓	✗	✗	✗	✓	✓
Datos en tiempo real	✓	✓	✓	✓	✓	✗	✗	✓	✓
Almacenamiento de datos	✓	✓	✓	✓	✗	✗	✗	✓	✓
Soporte técnico	✓	✓	✓	✓	✓	✗	✓	✓	✓
Tipo de solución	PaaS	PaaS	Complete IoT	PaaS	Data Analytics	API to devices	-	IaaS	SaaS

Plataformas IoT - ThingSpeak



- ThingSpeak es una plataforma Open Source auspiciada por Mathworks (Matlab) con una API para almacenar y recuperar datos de los objetos usando el protocolo HTTP sobre Internet o vía LAN (Local Area Network).
- API disponible en GitHub para su descarga en un servidor propio. Es totalmente abierta, por lo que también se puede modificar su código fuente original y así contribuir a la comunidad con nuevas características.
- Permite desarrollar plugins y ofrecen la posibilidad de crear aplicaciones de forma nativa en la plataforma ThingSpeak.
- Integración con diversos dispositivos hardware y software: Arduino, Raspberry Pi, IoBridge / RealTime.io, Electric Imp, Móviles / Aplicaciones web, Redes Sociales, Análisis de datos con MATLAB.
- Aplicaciones: principalmente esta orientado al mundo del Smart Home.

Ver plataforma: <https://thingspeak.com/>

Plataformas IoT – Thinger.io

- Es una plataforma de **código abierto**. La encontramos en su propio servidor como en GitHub para instalarla en una máquina propia.
- La programación es muy sencilla. Hay una librería en el repositorio oficial de Arduino. Además, es compatible con ESP8266, Raspberry Pi o Intel Edison.
- Dispone de una consola de **administración** bastante potente donde podemos gestionar y geolocalizar nuestros objetos conectados.



	GRATIS	BUSINESS
Máximo número de conexiones servicios web	10	Consultar
Tiempo histórico (meses) Rápido acceso y estadísticas	30 días	Consultar
Número de dispositivos	3	Consultar
Precio	Gratis	Consultar

OpenStack Cloud Computing Platform

https://www.youtube.com/watch?v=_gWfFEuert8

IBM Cloud

<https://cloud.ibm.com>

<https://www.youtube.com/watch?v=HzugDzl2cfg&t=71s>



Máster en Tecnologías de Análisis de Datos Masivos: BIG DATA

Internet de las Cosas en el Contexto de Big Data

Tema 2: Herramientas y plataformas para Internet de las Cosas

Juan Antonio Martínez juanantoni@um.es
Antonio F. Skármata Gómez skarmeta@um.es

