

ARQUITECTURAS E INFRAESTRUCTURAS PARA INTELIGENCIA ARTIFICIAL

Máster en Inteligencia Artificial

Tema 3
Computación en la nube

Equipo Docente 

Contenido

1. Paradigma Cloud Computing
2. Modelos de despliegue Cloud
3. Modelos de Servicio
4. Virtualización
5. Contenedores

1. Paradigma Cloud Computing

The NIST Definition of Cloud Computing (2011)

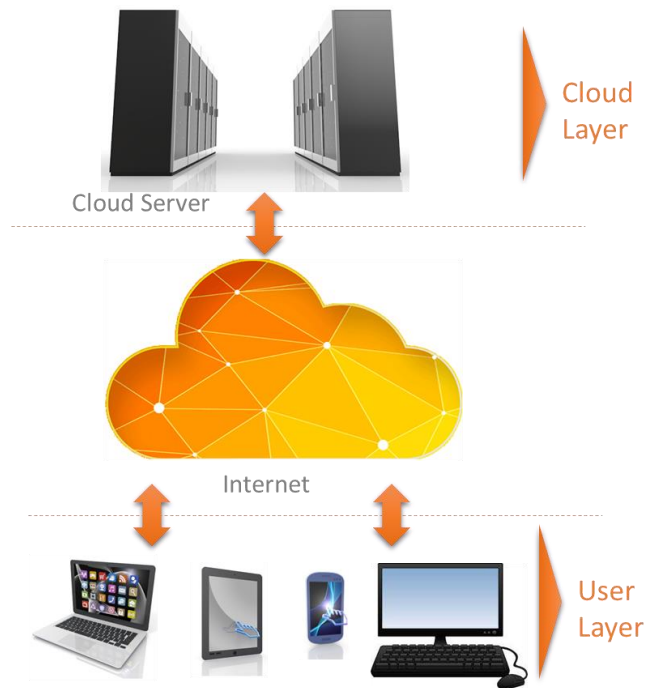
Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

NIST: National Institute of Standard and Technology

<https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-145.pdf>



1. Paradigma Cloud Computing



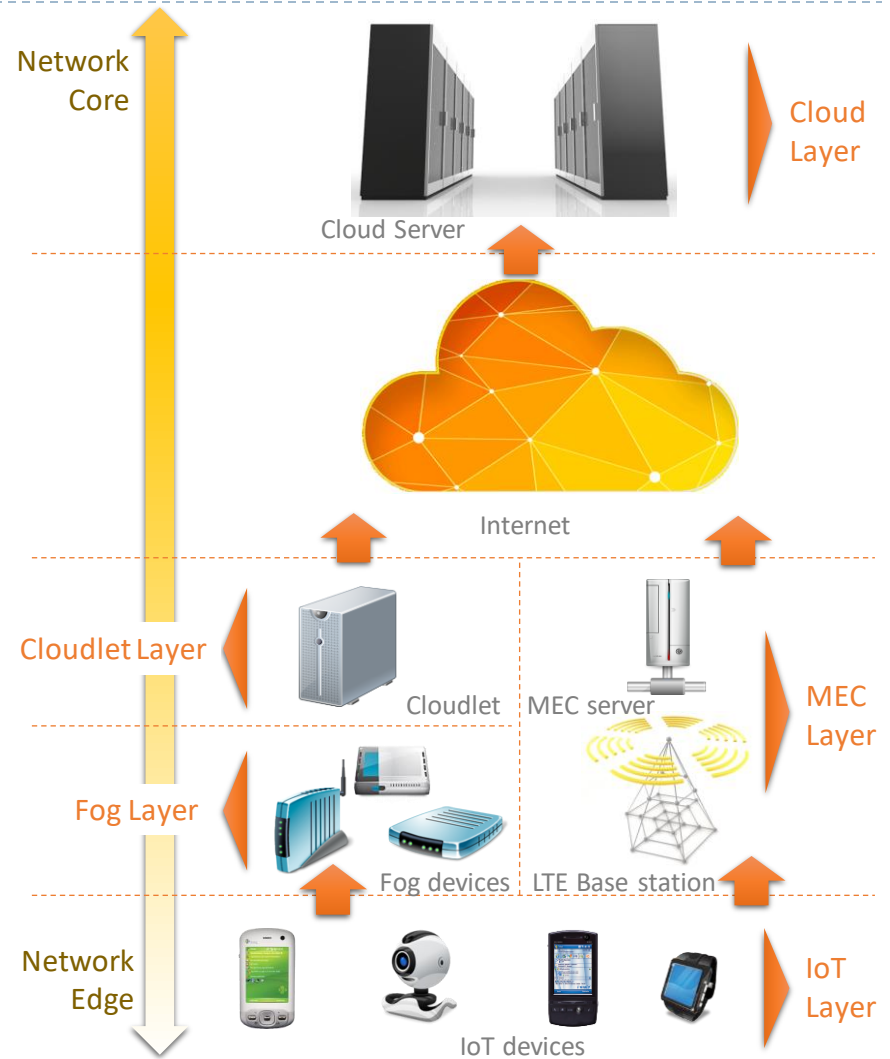
1. Paradigma Cloud Computing

- ▶ **Externalización de los servicios de TI**
 - ▶ Externalización de procesamiento.
 - ▶ Externalización del almacenamiento.
 - ▶ Externalización de los servicios.
- ▶ **Paradigma *Mobile Cloud Computing* (MCC)**
 - ▶ Externalizar la computación de dispositivos móviles y otros dispositivos IoT.

1. Paradigma Cloud Computing

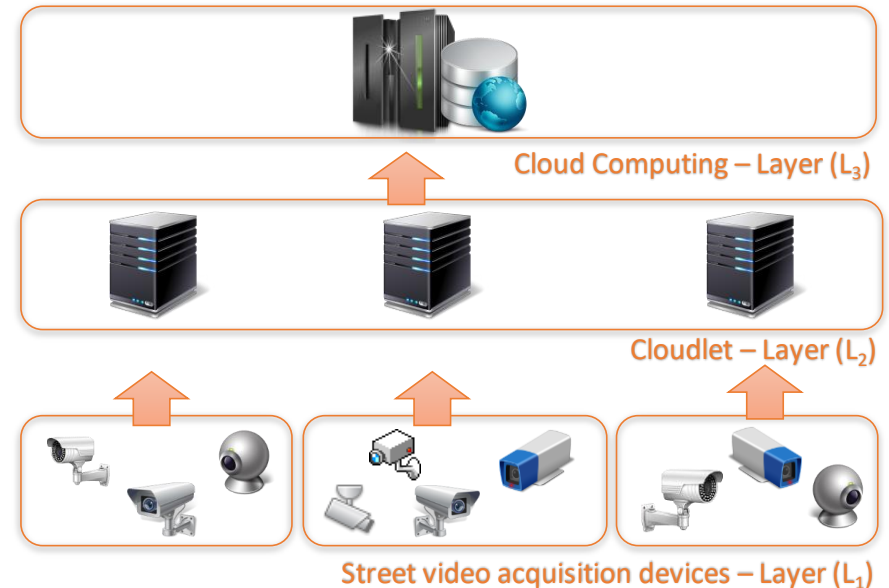
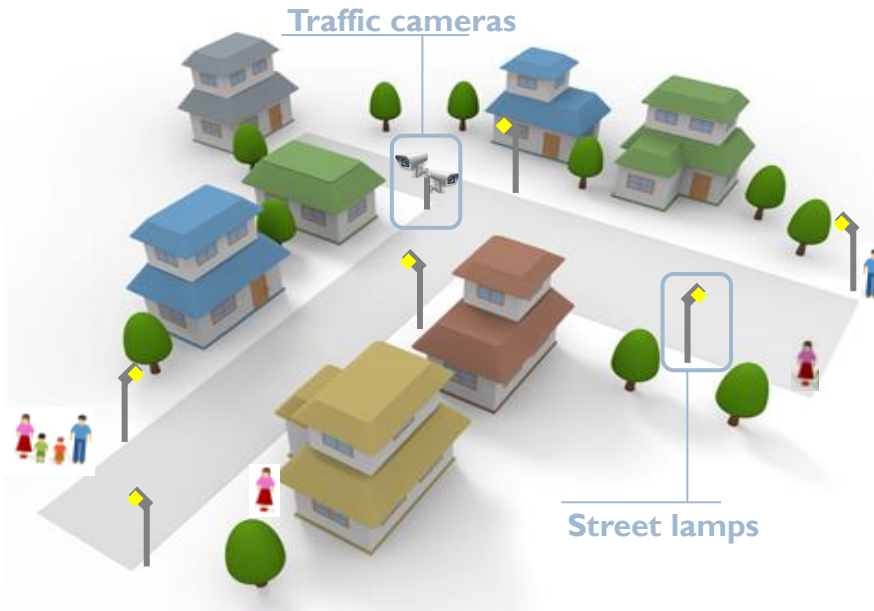
- ▶ **Arquitectura multinivel Cloud**
- ▶ **Concepto de Edge computing**

- Mora H. et al., (2019) Multilayer architecture model for mobile cloud computing paradigm, Complexity, 1-13.
<https://doi.org/10.1155/2019/3951495>
- Kong et al., (2022) Edge Computing for Internet of Everything: A Survey, IEEE Internet of Things Journal, Volume: 9, Issue: 23
<https://doi.org/10.1109/IJOT.2022.3200431>



1. Paradigma Cloud Computing

► Ejemplo ciudades inteligentes

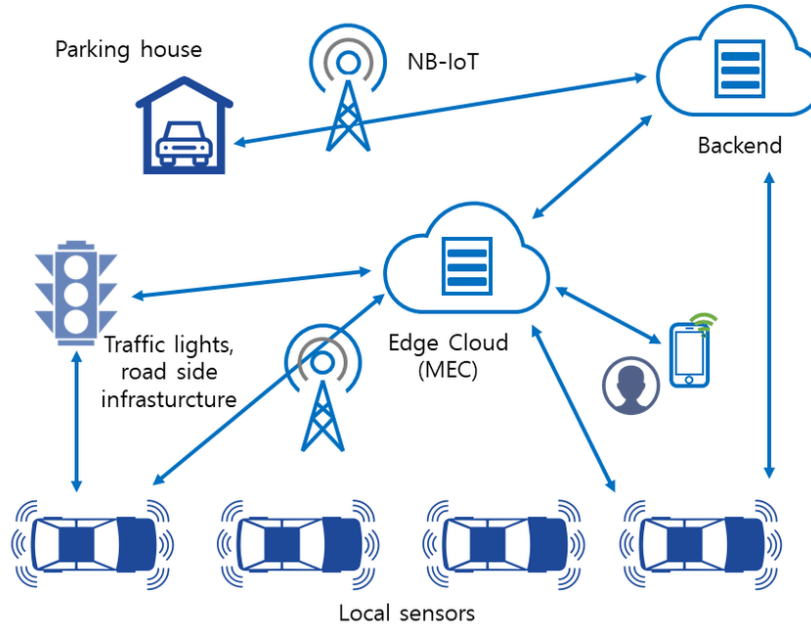


- Mora H. et al., (2019) Distributed Architectures for Intensive Urban Computing: A Case Study on Smart Lighting for Sustainable Cities. IEEE Access, Vol. 7.

<https://doi.org/10.1109/ACCESS.2019.2914613>

1. Paradigma Cloud Computing

► Ejemplo coche autónomo



- Yeruva V. (2021) Edge Computing's Applications In Autonomous Driving And Business At Large <https://www.forbes.com/sites/forbestechcouncil/2021/09/27/edge-computings-applications-in-autonomous-driving-and-business-at-large/?sh=5d470bd996aa>
- Cui et al., (2020) Offloading Autonomous Driving Services via Edge Computing, IEEE Internet of Things Journal (Volume: 7, Issue: 10). <https://doi.org/10.1109/JIOT.2020.3001218>

1. Paradigma Cloud Computing

▶ Ventajas del paradigma Cloud

- ▶ Capacidad de abstracción de los recursos
- ▶ Escalabilidad
- ▶ Modelo de servicio pago por uso (as-a-service)
- ▶ Acceso ubicuo
- ▶ Seguridad

▶ Inconvenientes del paradigma Cloud

- ▶ Infraestructuras compartidas / problemas de rendimiento
- ▶ Dependencia tecnológica
- ▶ Deslocalización de la información y pérdida de control
- ▶ Mayor riesgo de fuga de información
- ▶ Se diluye la ventaja competitiva

2. Modelos de despliegue Cloud

- ▶ Modelos de despliegue Cloud
 - ▶ Private Cloud
 - ▶ Public Cloud
 - ▶ Hybrid Cloud
 - ▶ Esquema Multicloud
 - ▶ Modelo Clouds Federadas

2. Modelos de despliegue Cloud

► Public Cloud

- La infraestructura de la nube está preparada para su uso abierto por el público en general. Puede ser propiedad de, administrada y operada por una organización empresarial, académica o gubernamental, o alguna combinación de ellas. Existe en las instalaciones del proveedor de la nube.

2. Modelos de despliegue Cloud

► Private Cloud

- La infraestructura de la nube se suministra para uso exclusivo de una sola organización que comprende múltiples consumidores (por ejemplo, unidades de negocios). Puede ser propiedad de la organización, de un tercero o de una combinación de ellos, y puede existir dentro (on-premise) o fuera de las instalaciones.

2. Modelos de despliegue Cloud

► Hybrid Cloud

- La infraestructura de la nube es una composición de dos o más infraestructuras de nubes distintas (privadas o públicas) que siguen siendo entidades únicas, pero que están unidas entre sí por una tecnología que permite la portabilidad de los datos y las aplicaciones (por ejemplo, para el equilibrio de la carga entre las nubes).

2. Modelos de despliegue Cloud

- ▶ **Comparativa despliegue Cloud en las propias instalaciones (on-premise) e instalaciones del proveedor:**
 - ▶ **Personalización y escalabilidad**
 - ▶ On-Premise: permitirá un uso exclusivo y personalizado de los datos y del hardware, pero tiene una capacidad limitada.
 - ▶ Proveedor: no se tiene control absoluto sobre los datos, pero es potencialmente ilimitado.
 - ▶ **Seguridad**
 - ▶ On-premise: garantiza que las personas que tienen acceso a los datos han sido autorizadas previamente por la empresa.
 - ▶ Proveedor: la empresa confiará sus datos a un tercero.
 - ▶ **Costes:**
 - ▶ On-premise: es necesaria una elevada inversión inicial.
 - ▶ Proveedor: pago por uso desde el primer día.

2. Modelos de despliegue Cloud

► Esquema de despliegue MultiCloud

- Multicloud describe el uso de más de un proveedor de servicios de Cloud.
- Es necesario un orquestador para gestionar los recursos Cloud de diferentes orígenes.
- Ventajas:
 - Reduce la dependencia tecnológica.
 - Mejora la flexibilidad en el uso de recursos.
- Inconvenientes:
 - Mayor coste y complejidad de mantenimiento.
 - Mayor riesgo de seguridad.



2. Modelos de despliegue Cloud

► Modelo Cloud Federadas

- Ecosistema en el que varios proveedores Cloud comparten servicios de forma transparente al usuario.
- Ventajas:
 - Mejora la eficiencia y coste en el uso de los recursos.
 - Incrementa la fiabilidad y escalabilidad.
- Inconvenientes:
 - Mayor complejidad de mantenimiento.
 - Mayor riesgo de seguridad.
 - Mayores necesidades de interoperabilidad.
- Proyecto Gaia-X: El Cloud Federado Europeo
 - <https://www.gaia-x.eu/>
 - <https://www.gaiax.es/>



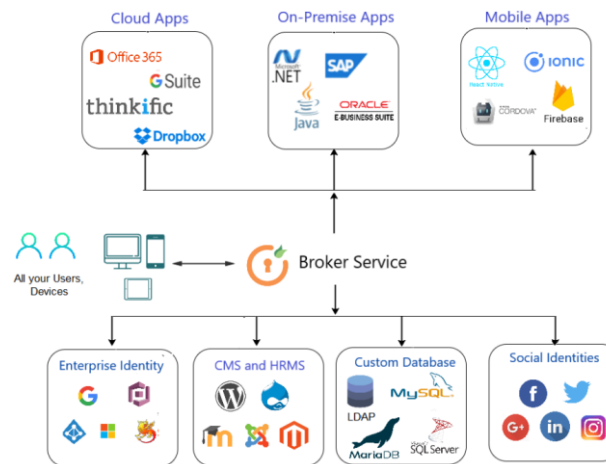
- Kar B. Et al. (2023) Offloading using Traditional Optimization and Machine Learning in Federated Cloud-Edge-Fog Systems: A Survey. IEEE Communications Surveys & Tutorials. <https://doi.org/10.1109/COMST.2023.3239579>
- Li Y. et al. (2022) Federated Clouds for Efficient Multitasking in Distributed Artificial Intelligence Applications. IEEE Transactions on Cloud Computing. <https://doi.org/10.1109/TCC.2022.3184157>

2. Modelos de despliegue Cloud

► Modelo Cloud Federadas

► Es necesario un servicio de Broker Cloud

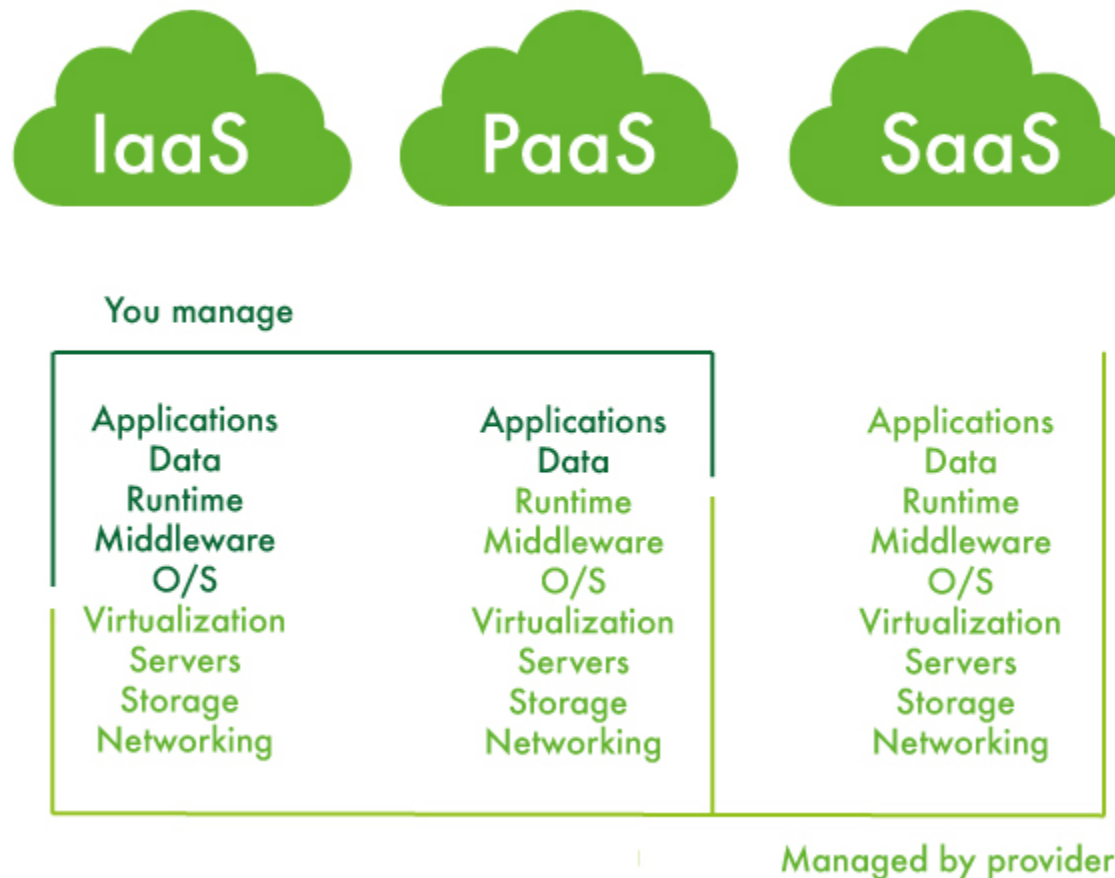
- Negocia condiciones con los distintos Cloud
- SLA (Service Level Agreements): Acuerdos de nivel de servicio



- C. Muralidharan, R. Anitha, (2022) Trusted cloud broker for estimating the reputation of cloud providers in federated cloud environment, Concurrency and Computation, Vol. 34, Issue 11, <https://doi.org/10.1002/cpe.6537>

3. Modelos de Servicio

► Modelos de Servicio



3. Modelos de Servicio

▶ IaaS: Infrastructure as a Service

- ▶ La capacidad proporcionada consiste en proporcionar procesamiento, almacenamiento, redes y otros recursos informáticos fundamentales en los que se puede desplegar y ejecutar software.
- ▶ El consumidor no gestiona ni controla la nube subyacente pero tiene control sobre los sistemas operativos, el almacenamiento y las aplicaciones desplegadas y posiblemente un control limitado de determinados componentes de la red.
- ▶ Ventajas:
 - ▶ Mejoras en la gestión de la infraestructura de TI.
 - ▶ Ahorro de costes y mayor eficiencia de los recursos de TI.
- ▶ Inconvenientes:
 - ▶ Mayor dependencia sobre el proveedor cloud.
- ▶ Ejemplo: Microsoft Azure, Amazon EC2

3. Modelos de Servicio

► PaaS: Platform as a Service

- La capacidad que se proporciona consiste en desplegar en la infraestructura de la nube aplicaciones creadas o adquiridas por el consumidor y creadas con lenguajes de programación, bibliotecas, servicios y herramientas apoyadas por el proveedor.
- El consumidor no gestiona ni controla la infraestructura de nube subyacente, incluidos la red, los servidores, los sistemas operativos o el almacenamiento, pero tiene control sobre las aplicaciones desplegadas y posiblemente sobre los parámetros de configuración del entorno de alojamiento de las aplicaciones.
- Ventajas:
 - Permite externalizar los procesos de desarrollo de software.
 - Se producen mejoras de la productividad y costes de desarrollo: no hay que mantener la plataforma: BD, servidores, etc.
- Inconvenientes:
 - Posible dependencia del proveedor en aspectos estratégicos.
- Ejemplo: Velneo (<http://velneo.es/>). Desarrollo de aplicaciones empresariales para móviles.

3. Modelos de Servicio

► SaaS: Software as a Service

- La capacidad que se le da al consumidor consiste en utilizar las aplicaciones del proveedor que se ejecutan en una infraestructura de nube. Se puede acceder a las aplicaciones desde varios dispositivos cliente a través de una interfaz de cliente ligero, como un navegador web o una interfaz de programa.
- El consumidor no gestiona ni controla la infraestructura subyacente de la nube, incluida la red, los servidores, los sistemas operativos, el almacenamiento o incluso las capacidades de las aplicaciones individuales, con la posible excepción de los ajustes de configuración de las aplicaciones específicas de los usuarios.
- Ventajas:
 - Software constantemente actualizado y común a todos los nodos.
 - Prestación de servicios a nivel global.
- Inconvenientes:
 - Dependencia del proveedor externo y de conexión a Internet.
 - Seguridad. (sol. Criptografía homomórfica).
 - Escalabilidad a largo plazo.

3. Modelos de Servicio

► FaaS: Function as a Service (Serverless computing)

- Aplicaciones nativas Cloud.
- Modelo de servicio en la nube que permite a los usuarios crear y ejecutar aplicaciones y procesos sin entrar en contacto con el servidor subyacente.
- Los recursos que se requieren se aportan de manera puntual y en tiempo real durante el tiempo de ejecución de una única llamada de función.
- Los recursos de computación son gestionados por el proveedor Cloud.
- Ventajas:
 - Suministro ágil de recursos. Mayor escalabilidad de recursos.
 - Solo se cobran gastos por recursos que realmente se han usado
- Inconvenientes:
 - El diseño y la implementación de las estructuras “serverless” requiere de más esfuerzo.
 - Mayor dependencia del proveedor.
- Ejemplo: Amazon AWS Lambda, MS Azure Functions, Google Cloud Functions

- Li Z. et al., (2022) The Serverless Computing Survey: A Technical Primer for Design Architecture, ACM Computing Surveys, Vol. 54, Issue 10, <https://doi.org/10.1145/3508360>

4. Virtualización

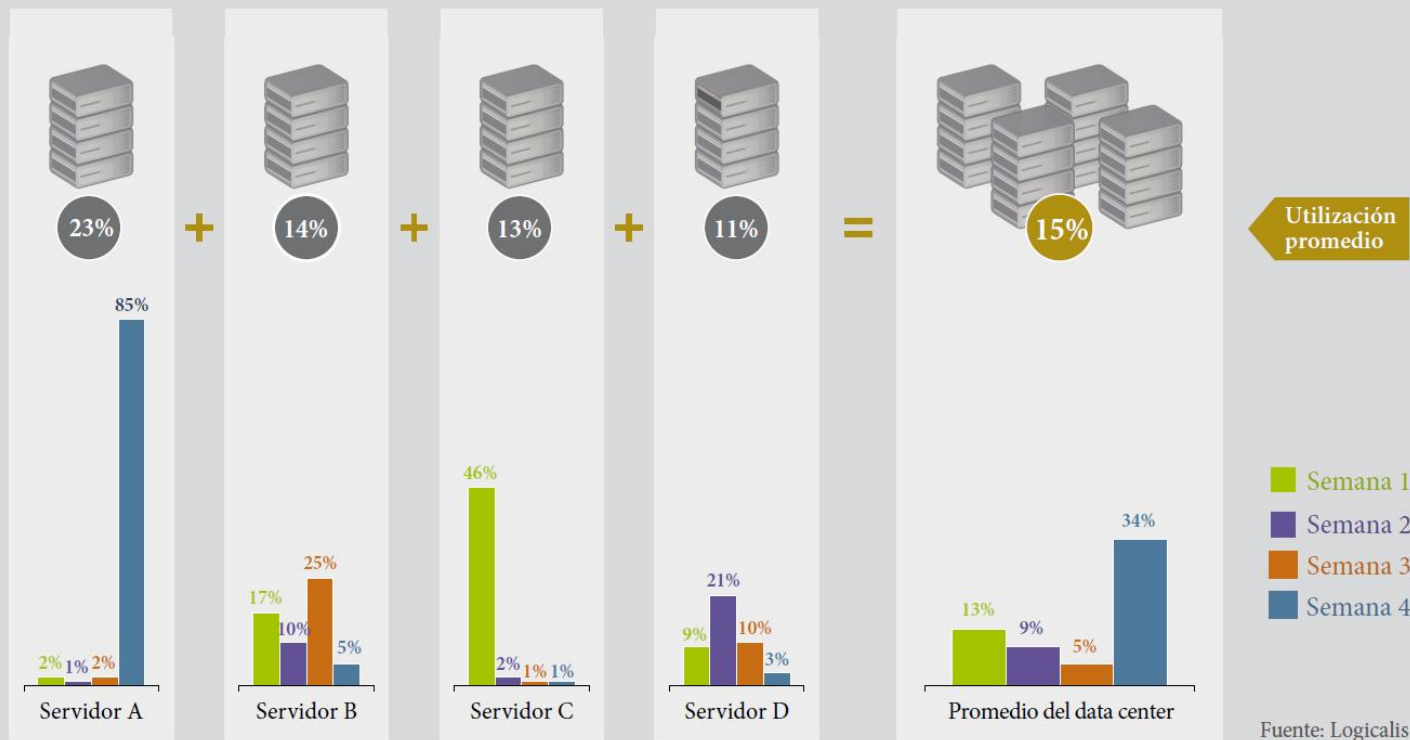
▶ Virtualización

- ▶ La virtualización permite presentar un conjunto de recursos de cómputo sin restricciones de configuración o ubicación.
- ▶ Se permite heterogeneidad de máquinas y sistemas operativos.
- ▶ La mejora del tiempo de procesamiento de los servidores permite reducir el número de máquinas para el mismo trabajo.
- ▶ Se reduce el coste en equipos y en energía eléctrica.
- ▶ Facilita la centralización de la administración del hardware.

4. Virtualización

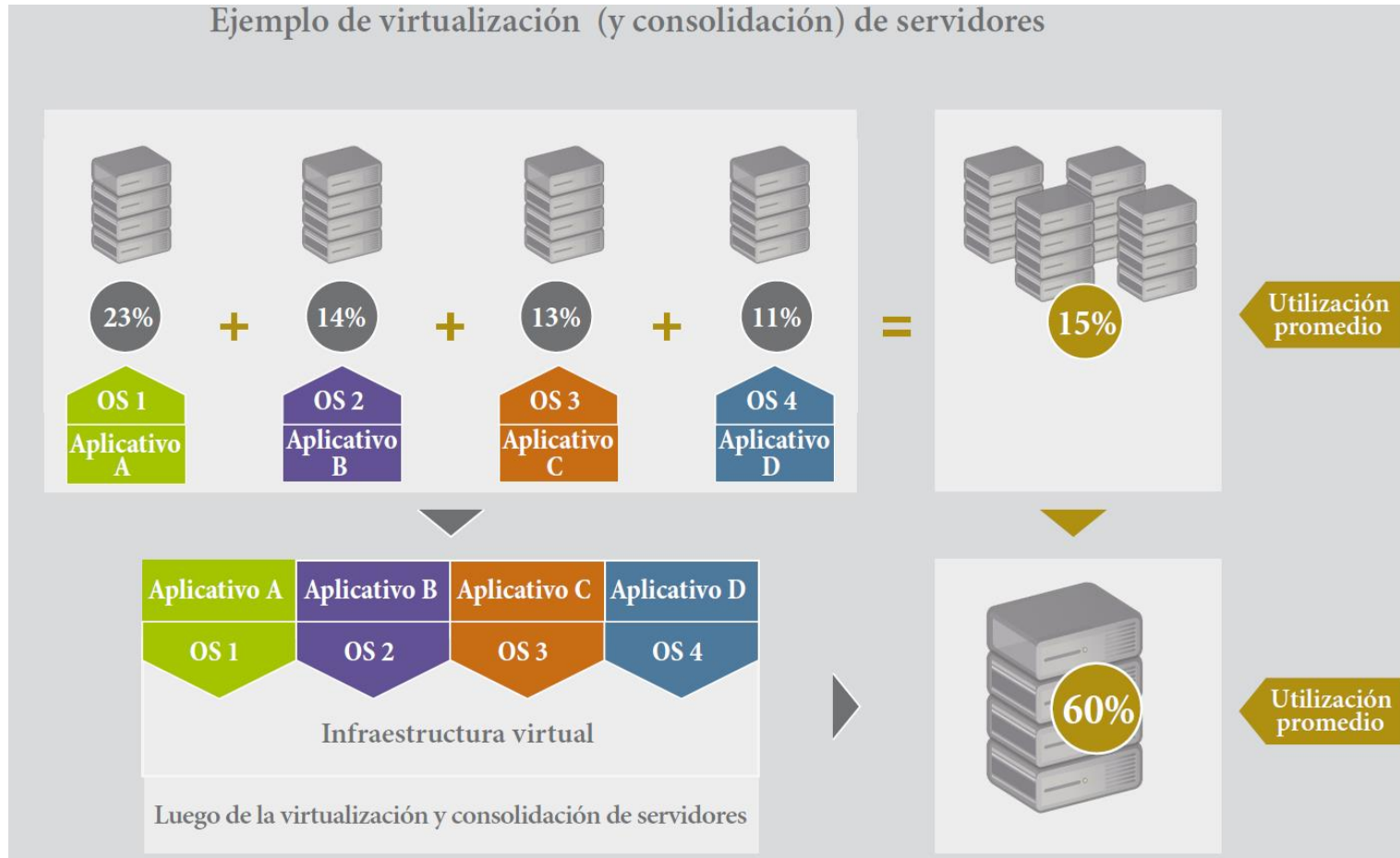
► Virtualización

Figura 1 Ejemplo de tasa de uso de servidores físicos en un *data center* durante un mes



4. Virtualización

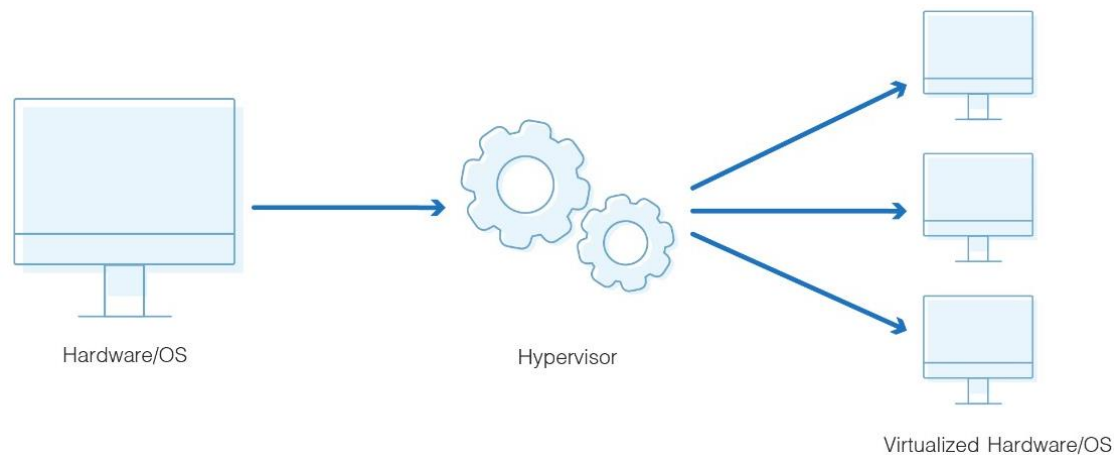
► Virtualización



4. Virtualización

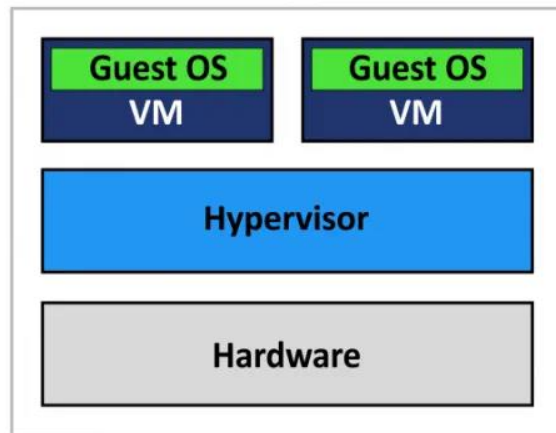
► Hypervisor

Es un sistema que crea y ejecuta máquinas virtuales mediante el uso compartido virtual de sus recursos.



4. Virtualización

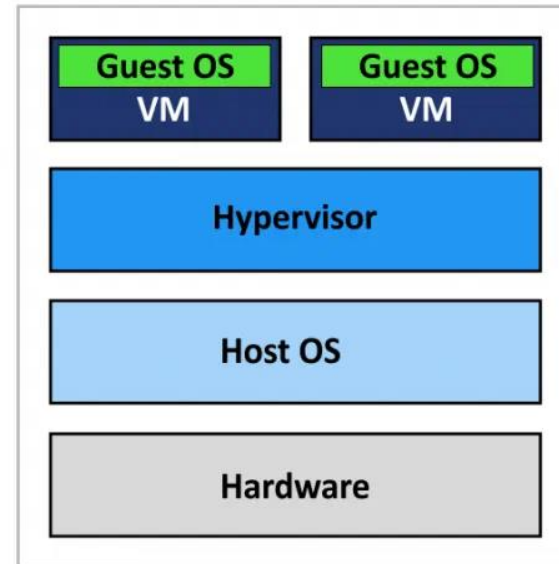
► Opciones de Virtualización



**Type 1 Hypervisor
(Bare-Metal Architecture)**

Ejemplos:

- Xen Server
- MS Hyper-V



**Type 2 Hypervisor
(Hosted Architecture)**

Ejemplos:

- VMware Workstation
- Oracle Virtual Box

5. Contenedores

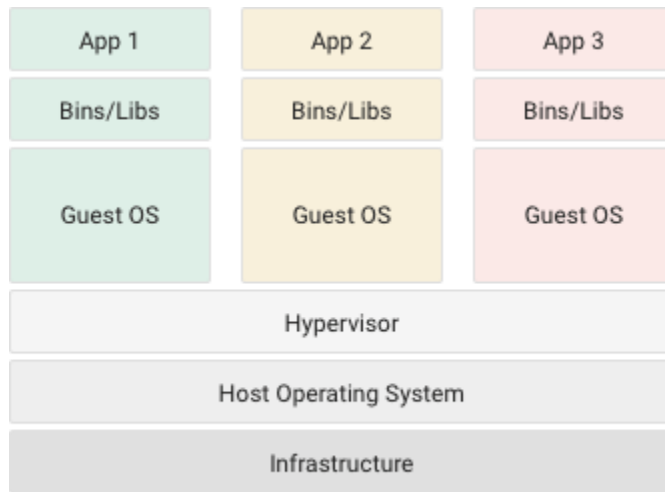
► Contenedores

- Un contenedor es una unidad estándar de software que empaqueta el código y todas sus dependencias para que la aplicación se ejecute de forma rápida y fiable de un entorno informático a otro. Una imagen de un contenedor es un paquete de software ligero, autónomo y ejecutable que incluye todo lo necesario para ejecutar una aplicación: código, tiempo de ejecución, herramientas del sistema, bibliotecas del sistema y configuraciones. Las imágenes de contenedores se convierten en contenedores en tiempo de ejecución.
- De este modo, el software en contenedores siempre se ejecutará de la misma manera, independientemente de la infraestructura. Los contenedores aíslan el software de su entorno y garantizan que funcione de manera uniforme a pesar de las diferencias, por ejemplo, entre el desarrollo y la puesta en escena.

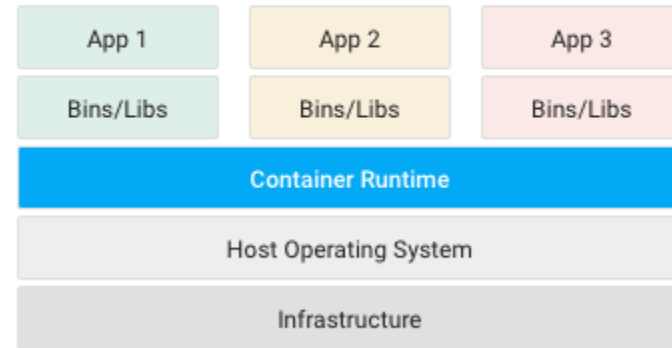
5. Contenedores

► Contenedores

- Los contenedores ofrecen una solución mucho más ligera que las máquinas virtuales con la que pueden trabajar los desarrolladores y equipos de operaciones de TI



Virtual Machines



Containers