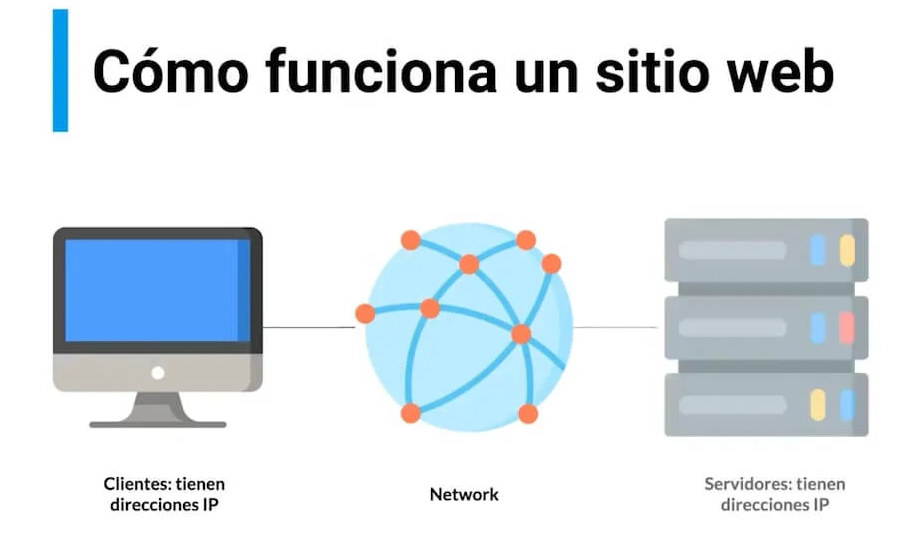
# **Visión general de las TI tradicionales**

Entendamos primero como funciona la web en términos simples. Tenemos un **cliente** con una dirección IP que se conecta a una **red** para hacer una **petición** a un **servidor** con otra dirección IP. Este servidor devuelve una respuesta al cliente.

Si la web fuera un servicio postal, el **cliente** seríamos nosotros con la **petición** o paquete que queremos enviar, la **red** sería el servicio postal en sí y el **servidor** representaría al destinatario al que queremos enviar el paquete.



## **¿Cómo está compuesto un servidor?**

Un servidor posee los siguientes componentes:

* **Cómputo/CPU**: Realiza las operaciones que necesitamos.
* **Memoria RAM**: Contiene la información a procesar mediante la CPU. Es como un cerebro
* **Almacenamiento**: Archiva datos, a modo de archivos de texto plano.
* **Bases de datos**: Información almacenada de manera estructurada
* **Redes**: Cables, routers y servidores conectados unos a otros. Servidores DNS

## **Terminología de IT (redes)**

En términos generales, un cliente envía un paquete a un **router**, el cual reenvía este paquete al **switch**, y este se encarga de distribuirlo.

* **Router**: Dispositivo que reenvía paquetes de datos entre redes informáticas
* **Switch**: Dispositivo que toma paquetes y los envía al servidor/cliente correcto en la red

## **Diseño tradicional de infraestructura**

Las grandes empresas de IT empezaron comprando servidores y montándolos en sus garajes. Se encontraron con problemas al tratar de expandir su infraestructura, como los costos de mover estos servidores, comprar nuevos, y más…

## **Problemas del enfoque de IT tradicional**

A continuación, conocerás algunas dificultades del enfoque de tecnología de la información habitual:

* **Renta**: Los costos de rentar espacios para mantener servidores son altos.
* **Mantenimiento**: El funcionamiento de los servidores es difícil de asegurar.
* **Remplazar y agregar hardware**: Pueden existir largos tiempos de espera para conseguir el hardware necesario.
* **Escalamiento limitado**: Podemos vernos limitados por el espacio donde almacenamos los servidores.
* **Monitoreo 24/7**: Debemos contratar gente para monitorear los servidores.
* **Desastres naturales**: ¿Cómo evitamos que se caigan nuestros servicios si ocurre un imprevisto?

# **Qué es la computación en la nube**

La computación en la nube es la entrega bajo demanda de recursos de IT como computación, almacenamiento y otros servicios a través de internet. En pocas palabras, es como si alquiláramos la computadora de otra persona.

Esta tecnología permite acceso instantáneo a los recursos que necesites, así como la adquisición del tipo y tamaño exacto de estos recursos. Algunos servicios que seguramente has usado son Gmail (proveedor de email), Dropbox (proveedor de almacenamiento) y Netflix (proveedor de video bajo demanda).

## **Modelos de computación en la nube**

A continuación, conocerás las distintas plataformas en la nube que utilizamos cuando trabajamos en proyectos personales o en nuestra empresa.

## **Nube pública**

La nube pública se refiere a los recursos de proveedores que utilizamos a través de internet y algunos ejemplos son Google Cloud Platform (GCP), Azure y AWS.

Además, posee estas ventajas:

* Elimina los gastos de capital comercial (CapEx) y reduce el gasto operativo (OpEx).
* Reduce los precios en economías de escala.
* Despliega aplicaciones a nivel global en cuestión de minutos.

## **Nube privada**

La nube privada es un servicio empleado por una organización que no está abierto al público. Permite un control total de la infraestructura y es útil para aplicaciones con requerimientos específicos de seguridad o comerciales.

## **Nube híbrida**

La nube híbrida consiste en mantener nuestra infraestructura y extender sus capacidades mediante la nube pública. Posibilita el control sobre activos sensibles en tu infraestructura privada, aprovechando la flexibilidad y rentabilidad de la nube pública.

## **Características de la computación en la nube**

Ahora que conoces los distintos modelos de tecnología en la nube, es importante que hablar sobre sus propiedades de computación.

* Este modelo genera un autoservicio en demanda (con registros en la plataforma ya se pueden proveer recursos)
* Tiene un amplio acceso a la red.
* Proporciona un espacio donde los clientes pueden compartir infraestructura y recursos de manera segura.

## **Problemas resueltos por la nube**

Por último, es crucial que conozcas las cualidades que trae implementar un sistema de computación en la nube.

* La nube aporta flexibilidad (puedes cambiar los tipos de recursos cuando sea necesario)
* Brinda rentabilidad y un servicio medido pues pagas solo por lo que usas.
* Trae escalabilidad al agregar capacidad para hardware o equipos que necesitan acomodar cargas grandes.
* Ofrece elasticidad al dar capacidad de escalar automáticamente cuando sea necesario.
* Tiene alta disponibilidad y tolerancia a fallos.
* Proporciona agilidad (puedes desarrollar, probar y ejecutar rápidamente aplicaciones en la nube)

# **Los diferentes tipos de cómputo: IaaS vs. PaaS vs. SaaS**

Ahora que conoces más sobre la tecnología en la nube, es importante introducir sus **distintos tipos de servicio** en la industria para identificar sus diferencias.

Estos modelos varían de acuerdo al tipo de servicio informático que pueda ofrecer, como servidores, almacenamiento, software o bases de datos.

## **Infrastructure as a Service (IAAS)**

La infraestructura como servicio (IAAS) proporciona componentes básicos de IT en la nube, es decir, **redes, computación**, **almacenamiento, etc**. A su vez, provee el máximo nivel de flexibilidad para adaptarlo a tus necesidades.

Ejemplos:

* Azure Virtual Machines.
* Linode.
* Digital ocean.
* S2 AWS.

## **Platform as a Service (PAAS)**

Los modelos que ofrecen una plataforma como servicio (PAAS) eliminan la necesidad de que administremos la infraestructura y proveen una plataforma para gestionar aplicaciones.

Ejemplos:

* Heroku.
* Google App Engine.
* AWS Elastic Beanstalk.

## **Software as a Service (SAAS)**

El Software como servicio (SAAS) brinda un producto de software terminado que es ejecutado y administrado por el proveedor del servicio.

Ejemplos:

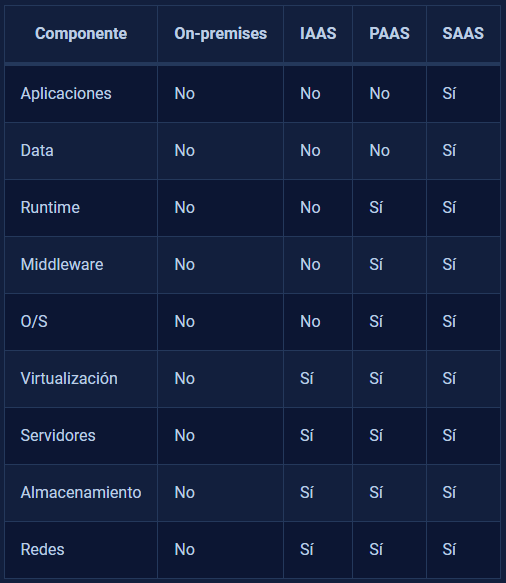
* Amazon Rekognition.
* Dropbox.
* Zoom.
* Gmail.

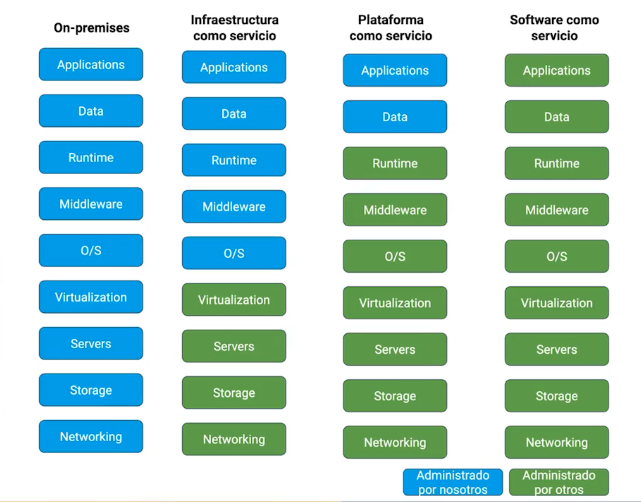
## **On -premises**

On-premises se refiere a una forma tradicional de cómputo en la cual nos encargamos de gestionar nuestra propia infraestructura.

## **Responsabilidades según el tipo de cómputo**

En la siguiente tabla se muestra qué componentes de IT están administrados según el tipo de cómputo en la nube. “Sí” indica que el componente está administrado por el proveedor de nube, “No” indica que nosotros somos responsables del componente.





# **Una pequeña historia de AWS**

**Benjamin Black** y **Chris Pinkham** son los principales desarrolladores de Amazon Web Services y crearon esta compañía a partir de la necesidad de impulsar nuevas tecnológicas en momentos de mayor tráfico y demanda.

La historia de AWS está repleta de hitos, pues es una de las plataformas más utilizadas en distintas startups y compañías que están transformando su industria. ¡No te preocupes! Aquí te resumimos fácil su línea del tiempo.

## **Línea del tiempo de AWS**

Hace veinte años nació esta promesa tecnológica y en la actualidad ¡tiene clientes en más de 245 países y territorios!

* 2002 → Se lanza internamente la plataforma.
* 2003 → Comienza a comercializarse la idea de AWS.
* 2004 → Llega al público el servicio SQS.
* 2006 → Vuelve a lanzarse al público SQS, S3 y EC2.
* 2007 → Abren operaciones en Europa.
* 2009 → Lanzan el servicio RDS (Relational Database).
* 2010 → Sale al mercado el servicio Route 53.
* 2012 → Lanzan DynamoDB (una base de datos no relacional).

## **AWS en números**

Quizás sean un gran seguidor y fiel cliente de esta compañía, pero… ¿Conocías estos datos?

* En 2019, AWS logró $35.02 mil millones de dólares en ingresos anuales.
* AWS representó el 47% del mercado en 2019.
* Esta plataforma posee más de un millón de usuarios activos.