Nube y modelos de servicio

Esta mañana seguramente has escuchado una canción en Spotify, has abierto Classroom y has guardado algo en Google Drive. Nada de eso se ejecutaba en tu móvil, no de verdad. ¿Dónde estaba la canción? ¿Dónde están tus deberes ahora mismo? ¿Y quién mantiene encendida a las tres de la madrugada la máquina que los guarda?

La respuesta es «en la nube», y es menos misteriosa de lo que parece. Un hotel de Calvià que recibe diez veces más reservas en agosto que en enero no puede comprar diez servidores para agosto y tirar nueve en septiembre. Alquila. Como desarrollador de DAM, casi todo lo que construyas acabará ejecutándose en el ordenador de otro, así que necesitas saber exactamente qué estás alquilando.

Qué aprenderás

- Definir la **computación en la nube** con sus cinco características y distinguirla del **on-premise**
- Distinguir los tres **modelos de servicio**: **IaaS**, **PaaS** y **SaaS**
- Decir quién gestiona qué en cada modelo, y qué parte es **siempre** tuya
- Clasificar servicios reales que usas, con pruebas de la propia web del proveedor
- Publicar tu propia página web con **GitHub Pages** y decir qué modelo es
- Evitar la trampa: «la nube» no está en el cielo, es un edificio lleno de servidores de otro

La nube es alquilar a la compañía de suministro

> La **computación en la nube** es el acceso bajo demanda, a través de la red, a recursos informáticos (servidores, almacenamiento, bases de datos, aplicaciones) que un proveedor comparte entre muchos clientes, que pueden crecer o reducirse rápidamente y que pagas según el uso.

En el juego de construir ciudades de este módulo, la nube es **alquilar el agua y la electricidad a una compañía de suministro** en lugar de construir tu propia central. Abres el grifo y pagas la factura. La alternativa se llama **on-premise** (en local): la empresa compra sus propios servidores y los tiene en su propio edificio.

El instituto de normalización de Estados Unidos, el NIST, escribió la definición que todo el mundo cita (SP 800-145, 2011). Enumera cinco características. Un servicio al que le faltan no es realmente nube:

Característica	Qué significa	Ejemplo
Autoservicio bajo demanda	Te das de alta y creas los recursos tú mismo, sin esperar a que nadie instale nada.	Crear un servidor desde un formulario web en un par de minutos
Acceso por la red	Llegas a ella desde cualquier lugar y dispositivo con conexión.	Abrir Classroom en el instituto, en casa o en el móvil

Característica	Qué significa	Ejemplo
Recursos compartidos	El mismo hardware da servicio a muchos clientes, separados lógicamente.	Miles de empresas en los mismos centros de datos
Elasticidad	Los recursos se pueden añadir o quitar rápidamente según cambia la demanda.	La web del hotel en agosto y en enero
Pago por uso	El uso se mide y pagas lo que consumes.	Como la factura de la luz

Volvamos al hotel. On-premise, compra un servidor, lo pone en una sala con aire acondicionado y paga a alguien para que lo cuide: una gran inversión el primer día y, si en agosto se queda corto, mala suerte. En la nube no hay inversión inicial, paga cada mes según el uso y en agosto pide más capacidad.

> **Cuidado: «la nube está en algún sitio ahí arriba»**
> La nube es física. Es un **centro de datos**: un edificio lleno de servidores, cables, fuentes de alimentación y refrigeración, propiedad de un proveedor como Amazon, Microsoft, Google o Hetzner. Un disco duro externo o un servidor en el armario de la oficina no es nube: nadie más lo comparte, no crece bajo demanda y nadie te lo factura por uso.

Tres modelos de servicio: IaaS, PaaS y SaaS

Puedes alquilar a tres alturas distintas. Cuanto más subes, más hace el proveedor por ti.

> Un **modelo de servicio** dice qué parte del sistema te da hecha el proveedor y qué parte te queda a ti para gestionar.

	IaaS · infraestructura	PaaS · plataforma	SaaS · software
Qué obtienes	Máquinas virtuales, red y almacenamiento	Un entorno listo donde despliegas tu código	Una aplicación terminada, lista para usar
Qué gestionas tú	Sistema operativo, software, aplicación y datos	La aplicación y los datos	Solo tus datos y tu configuración
Para quién es	Administradores de sistemas	Desarrolladores	Usuarios finales
Ejemplos	Amazon EC2, Azure Virtual Machines, Google Compute Engine, Hetzner Cloud	Google App Engine, Heroku, Azure App Service, GitHub Pages	Gmail, Google Classroom, Microsoft 365, Spotify, Kahoot

- **IaaS** (*Infrastructure as a Service*, infraestructura como servicio): alquilas una **máquina virtual**, un ordenador simulado por software sobre el hardware del proveedor. Es como alquilar un solar vacío con el agua y la luz conectadas: la casa la construyes tú.
- **PaaS** (*Platform as a Service*, plataforma como servicio): tú traes tu código y la plataforma lo ejecuta. Como alquilar un piso amueblado: tú traes tus cosas.
- **SaaS** (*Software as a Service*, software como servicio): simplemente usas la aplicación. Como alojarte en un hotel.

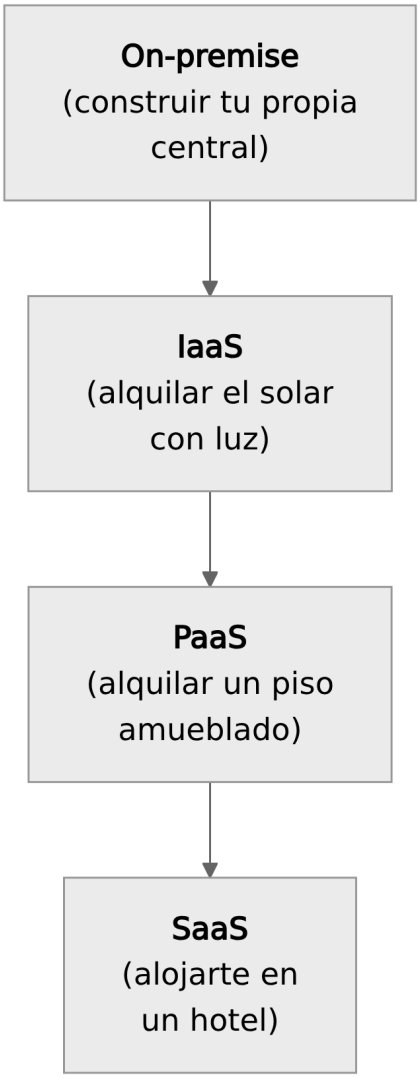
Quién gestiona qué

Apila las capas de cualquier sistema, desde el edificio hasta tus datos. Para cada modelo, marca qué es tarea del **cliente** (C) y qué es del **proveedor** (P):

Capa	On-premise	IaaS	PaaS	SaaS
Datos	C	C	C	C
Aplicación	C	C	C	P
Entorno de ejecución (Java, Node, base de datos…)	C	C	P	P
Sistema operativo	C	C	P	P
Virtualización	C	P	P	P
Servidores, discos y red	C	P	P	P
Edificio, electricidad y refrigeración	C	P	P	P

Mira la primera fila. **Tus datos son siempre responsabilidad tuya**, sea cual sea el modelo. Cuanto más subes, menos administras, pero menos control tienes. Los proveedores lo llaman **modelo de responsabilidad compartida**.

> **Cuidado: las fronteras no son muros**



> Algunos servicios están entre dos modelos. **Firebase**, por ejemplo, es una plataforma sobre la que construyes (PaaS), pero también ofrece piezas terminadas como el inicio de sesión de usuarios. Lo que se evalúa es el **motivo**: di qué gestiona el cliente y qué gestiona el proveedor, y la etiqueta sale sola.

GitHub Pages: una plataforma que puedes usar hoy

> **GitHub Pages** es un servicio gratuito de GitHub que publica una **web estática** (HTML, CSS e imágenes que no cambian según el visitante) directamente desde un repositorio de código.

Pones los archivos en un repositorio; GitHub los guarda, los sirve al mundo y mantiene los servidores en marcha. Nunca ves un servidor. Esa es la definición de **PaaS**, y lo usarás toda la unidad: en la Entrega 3 tu grupo publica su dosier como una web de GitHub Pages.

Tres datos que debes saber antes de empezar (comprobados el 25 de octubre de 2026 en la documentación de GitHub):

Dato	Qué significa para ti
Las condiciones de GitHub exigen tener **13 años o más**	Todo el mundo en este curso puede abrir una cuenta; es gratuita y no pide tarjeta
Con una cuenta gratuita, el repositorio debe ser **público**	Todo lo que subas lo puede ver cualquiera: nada de datos personales ni teléfonos; basta con tu nombre de pila
Pages no es para tiendas ni para recoger contraseñas	Un dosier, un portafolio o la página de un proyecto están bien

En clase: ¿a qué nivel juega?

Parte 1 · Clasifica servicios reales. En la plantilla de Classroom:

Paso	Haz esto	Deberías obtener
1	Anota **cuatro servicios que hayas usado esta semana** (móvil u ordenador) y añade estos cuatro: Amazon EC2, Heroku, Google App Engine, Hetzner Cloud	Una lista de ocho servicios
2	Clasifica cada uno como **IaaS, PaaS o SaaS** y escribe una frase con el porqué: ¿qué gestiona el cliente y qué el proveedor?	Ocho etiquetas con su motivo
3	Para cada uno, pega un enlace a la página **oficial** (página del producto o de precios, no un blog ni una IA) y cita la frase que lo demuestra	Ocho pruebas
4	Para la empresa de tu proyecto, enumera **tres sistemas** que tiene o que probablemente tiene (web, reservas, ERP, correo, control de planta…) y, para cada uno, el modelo que propondrías y un **proveedor real** que lo ofrezca	El borrador de la sección A de la D3

Parte 2 · Tu primera web en GitHub Pages.

Paso	Haz esto	Deberías ver
1	Inicia sesión en GitHub, o crea una cuenta gratuita en github.com/signup (correo, contraseña, nombre de usuario, una comprobación y un código enviado a tu correo)	Tu página de inicio de GitHub
2	Haz clic en **+** (arriba a la derecha) → **New repository**. Llámalo exactamente `yourusername.github.io`. Visibilidad **Public**. Activa **Add README**. **Create repository**	Tu nuevo repositorio con un `README.md`

Paso	Haz esto	Deberías ver
3	**Add file** → **Create new file**. Llámalo `index.html`. Escribe una página mínima: un título con tu nombre de pila y la frase «Esta web es un servicio … porque …». **Commit changes**	`index.html` en el repositorio
4	**Settings** → **Pages**. En **Build and deployment**, **Source**: **Deploy from a branch**; **Branch**: `main` y `/ (root)`. **Save**	Un mensaje de que tu web se está publicando
5	Espera (puede tardar hasta 10 minutos) y abre `https://yourusername.github.io`	Tu página, visible desde cualquier ordenador del mundo

Si no puedes o no quieres abrir una cuenta de GitHub, haz la página con **Google Sites** y tu cuenta del instituto, y di en la frase que Google Sites es SaaS, no PaaS. No afecta a tu nota individual: en la entrega, la web es trabajo de grupo.

Entrega en Classroom: la tabla de la parte 1 con sus pruebas, los tres sistemas de tu empresa y el enlace a tu web. **Sin IA generativa**: esta actividad se evalúa (trabajo de clase, 30 %).

Para tu proyecto

Esta lección arranca la Entrega 3 · Arquitectura en la nube, que se entrega el **lunes 8 de febrero de 2027 a las 23:59**, con las defensas individuales en la sesión 14 de ese mismo día.

Trabajo de hoy	Se convierte en	Criterio
Tres sistemas de la empresa con un modelo y un proveedor real	Sección A: modelos de servicio	a
Tu cuenta de GitHub y tu primera web	La web del grupo, publicada por el Técnico	Parte de grupo

Ideas clave

- La **computación en la nube** es alquilar recursos informáticos bajo demanda, a través de la red, compartidos, elásticos y pagados por uso. Funciona en **centros de datos** reales. Lo contrario es el **on-premise**.
- **IaaS** te da máquinas, **PaaS** te da un lugar donde desplegar tu código, **SaaS** te da la aplicación terminada.
- Cuanto más alto el modelo, menos administras y menos controlas. **Tus datos son siempre responsabilidad tuya** (el **modelo de responsabilidad compartida**).
- Los usuarios finales se encuentran sobre todo con **SaaS**; los desarrolladores viven sobre todo en **PaaS** e **IaaS**.
- **GitHub Pages** publica una **web estática** desde un repositorio: un PaaS que puedes usar gratis, con un repositorio público.

Comprobación rápida

1 ¿Cuál de estos es computación en la nube?

A Un disco duro externo donde la empresa copia sus archivos cada viernes
B Un servidor comprado por la empresa y guardado en su propia oficina
C Almacenamiento alquilado a un proveedor, al que se accede por internet y que se factura por GB usado
D Un pendrive que comparte toda la oficina

2 Una empresa alquila una máquina virtual e instala en ella el sistema operativo, la base de datos y su aplicación. ¿Qué modelo es?

- **A** IaaS
- **B** PaaS
- **C** SaaS
- **D** On-premise

3 ¿En qué modelo el cliente gestiona solo sus datos y su configuración?

- **A** IaaS
- **B** PaaS
- **C** SaaS
- **D** En ninguno: el cliente siempre gestiona el sistema operativo

4 Subes los archivos de tu web a un repositorio y GitHub Pages la publica sin que gestiones ningún servidor. ¿Qué modelo es?

- **A** IaaS
- **B** PaaS
- **C** SaaS
- **D** No es nube, porque es gratis

5 ¿Qué afirmación sobre el modelo de responsabilidad compartida es cierta?

- **A** En SaaS, el proveedor es responsable de los datos del cliente
- **B** En IaaS, el proveedor actualiza el sistema operativo de tus máquinas virtuales
- **C** Sea cual sea el modelo, el cliente es responsable de sus datos
- **D** On-premise, el proveedor gestiona el edificio y la electricidad

6 Un compañero dice: «La nube es solo internet; mis archivos están en el aire». ¿Cuál es la mejor respuesta?

- **A** Cierto: la nube no tiene ubicación física
- **B** Los archivos están en centros de datos reales propiedad de un proveedor; internet es solo el camino para llegar a ellos
- **C** Cierto, pero solo para los servicios SaaS
- **D** Falso: los archivos de la nube siempre se guardan en tu propio dispositivo

7 Una panadería de Inca quiere pedidos en línea sin contratar a un programador, y una start-up de Palma quiere desplegar su propia app sin gestionar servidores. ¿Qué modelos encajan mejor con cada una?

- **A** IaaS para la panadería, SaaS para la start-up
- **B** SaaS para la panadería, PaaS para la start-up
- **C** PaaS para las dos
- **D** On-premise para las dos, para ahorrar dinero

Ya sabes qué puedes alquilar. Ahora verás de quién es la central, para qué se usa la nube y cuánto sube la factura: Despliegue y funciones.

Soluciones

1. C · La nube es **bajo demanda**, **por la red**, **compartida**, **elástica** y **de pago por uso**. El disco, el servidor de la oficina y el pendrive no tienen ninguna de esas características; el servidor de la oficina es **on-premise**.

2. A · El proveedor solo da la **infraestructura** (la máquina virtual, la red y el almacenamiento). Todo lo que hay del sistema operativo hacia arriba es tarea del cliente: eso es **IaaS**.

3. C · En **SaaS** la aplicación está terminada y el proveedor gestiona todo lo que hay por debajo. El cliente sigue siendo dueño de los **datos**, que son responsabilidad suya en todos los modelos.

4. B · Tú traes el código; la plataforma lo guarda y lo sirve. Eso es **PaaS**. Ser gratis no impide que sea nube: el proveedor sigue gestionando el centro de datos.

5. C · La fila de los **datos** es siempre del cliente. En IaaS el sistema operativo también es tuyo; on-premise ni siquiera hay proveedor.

6. B · La nube es física: edificios llenos de servidores, cables y refrigeración. La **red** es cómo llegas a ellos. Por eso, en la lección 4, importará **en qué país** están.

7. B · La panadería necesita una **aplicación terminada** (SaaS). La start-up tiene su propio código y quiere que otro gestione los servidores: eso es **PaaS**.