

# Modelo de responsabilidades

DigitalHouse>



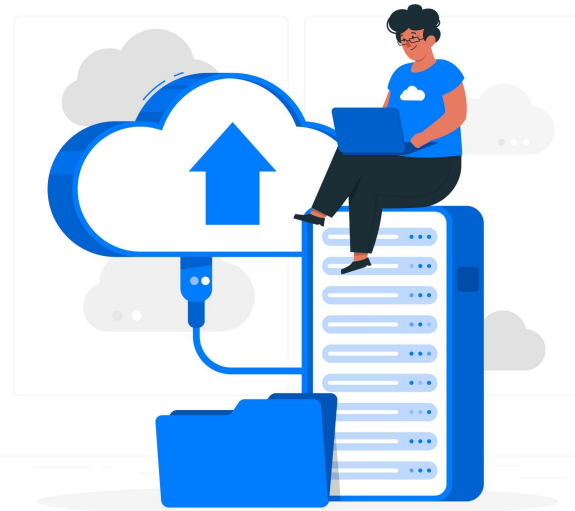
**Certified Tech  
Developer**

The Ultimate Degree

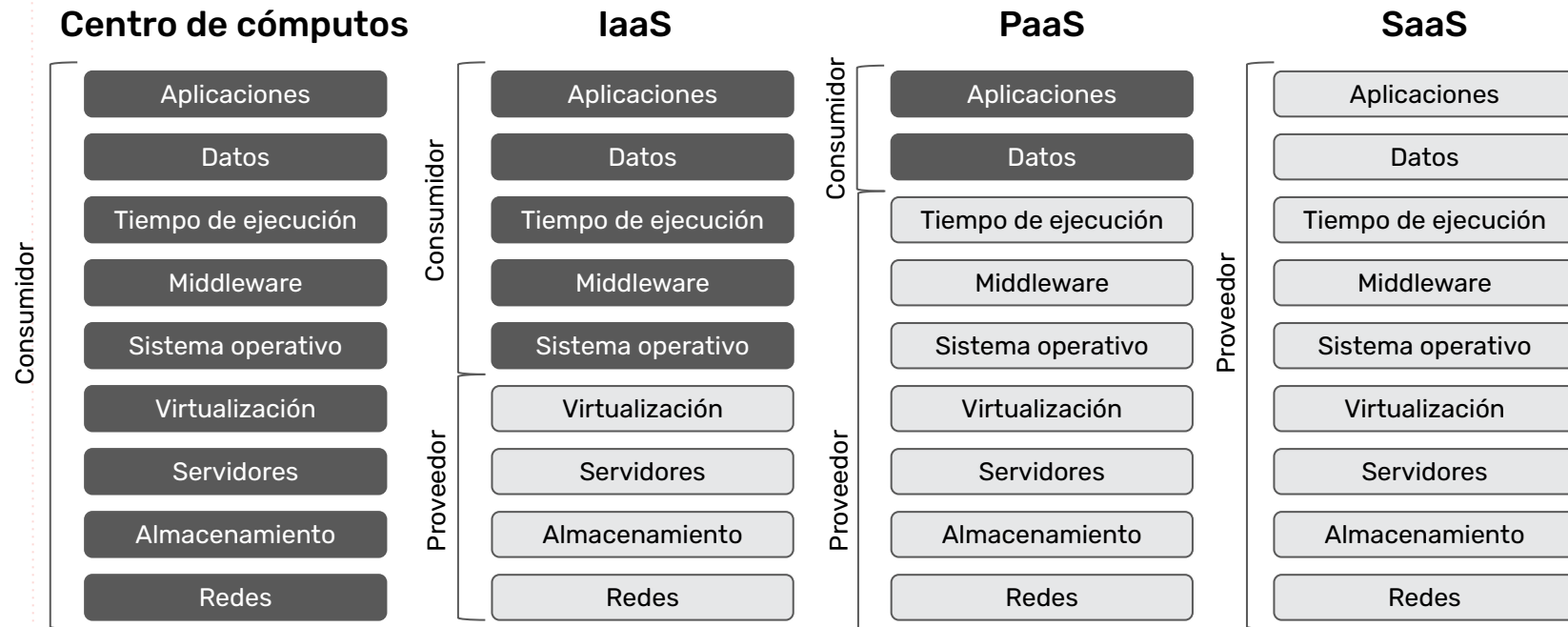
## ¿Por qué es necesario?

Como ya vimos, en cloud computing lo que hacemos es utilizar recursos de un tercero. Básicamente nos convertimos en inquilinos de aquel que es propietario de los activos informáticos.

El modelo de responsabilidades es un estándar que nos permite definir diferentes tipos de contratación de los servicios al dueño de la nube, qué tareas quedan delegadas a la nube y cuáles serán nuestra responsabilidad.



# El modelo

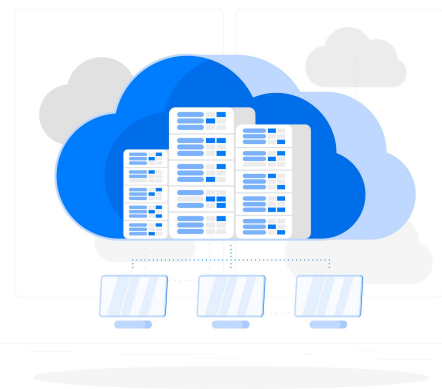


# Analicemos el modelo

Si miramos el modelo podemos identificar:

- Columnas: centro de cómputos, IaaS, PaaS y SaaS.
- Roles: consumidor y proveedor.
- Cajas que definen un conjunto de tareas, tecnologías y disciplinas con las que ya estamos familiarizados, como pueden ser: redes, virtualización o middleware.
- Y, finalmente, colores que tiñen cada una de ellas, dependiendo la columna en la que nos encontremos.

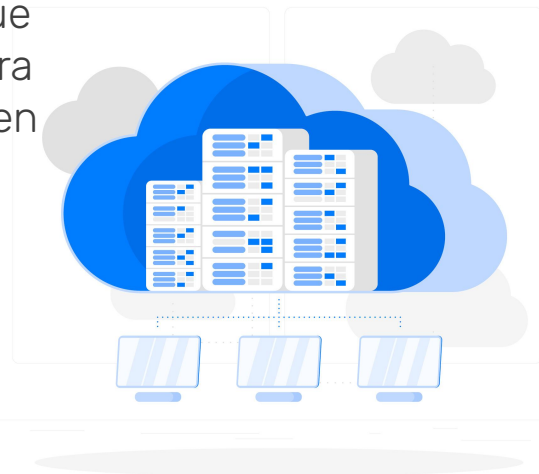
Pero, ¿qué significan?



# Los roles: consumidor y proveedor

Utilizamos estos dos roles para definir quién es responsable. Ya vimos que en cloud computing lo que hacemos es en realidad contratar recursos de manera flexible y con interfaces bien definidas –como pueden ser APIs– para la administración de los mismos por un tercero.

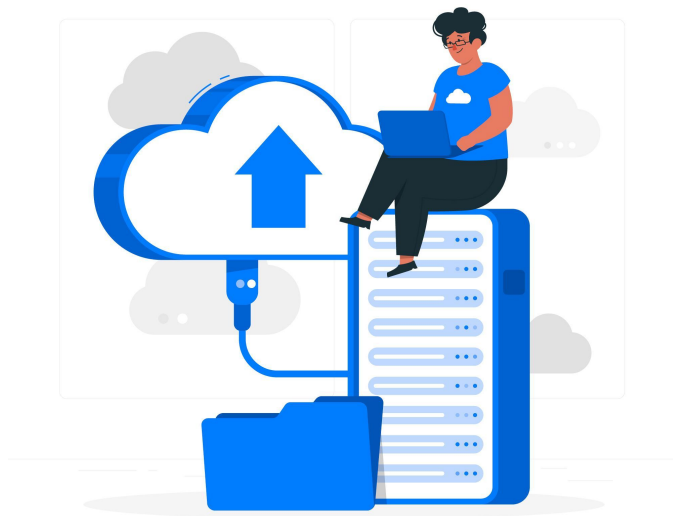
Ese tercero, dueño de la nube o de los centros de cómputos en los que decidamos correr nuestras aplicaciones, es a quien vamos a llamar **proveedor**.



# Los roles: consumidor y proveedor

## ¿Y quién es el consumidor?

La respuesta es sencilla, ¡nosotros!  
Que decidimos contratar los servicios  
del proveedor para ejecutar nuestras  
aplicaciones en sus centros de cómputos.



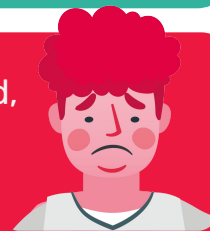
# Las columnas: centro de cómputos

Cuando hablamos de **centro de cómputos** lo que hacemos es definir de qué tareas nosotros seríamos responsables si somos quienes alojamos en su totalidad los servicios de tecnologías. En otras palabras, que no contratamos los servicios de ningún proveedor para que albergue o administre nuestros sistemas.



**Pros:** Tenemos control total sobre los activos, podemos decidir cómo, para qué, dónde y cuándo los utilizamos.

**Contras:** Tenemos responsabilidad total por la salud, utilización y mantenimiento del parque informático. No podemos escalar de manera flexible.



# Las columnas: IaaS

Cuando hablamos de **IaaS** nos referimos a un acrónimo en inglés: *Infrastructure as a Service* (infraestructura como servicio, en castellano). En este modelo de responsabilidad, el proveedor nos da la posibilidad de instanciar máquinas virtuales en su centro de cómputos sin que nosotros tengamos la necesidad de administrar la infraestructura subyacente.



**Pros:** Seguimos teniendo control sobre el sistema operativo, un ambiente ideal para instalar aplicaciones legacy. Nos permite escalar la infraestructura de forma más dinámica.

**Contras:** Seguimos siendo responsables por la administración y salud del sistema operativo y todos los servicios y aplicaciones que se ejecuten encima de este.





# Las columnas: PaaS

Cuando hablamos de **PaaS** nos referimos a un acrónimo en inglés: *Platform as a Service* (en español, plataforma como servicio). En este modelo de responsabilidad, el proveedor nos ofrece una tecnología específica, expuesta por medio de interfaces definidas, ya sean gráficas o APIs, y nos abstrae de toda la gestión de los recursos subyacentes.



**Pros:** Nos abstraemos completamente de la administración de los recursos de las capas inferiores. Delegamos en el proveedor la gestión, monitoreo y escalabilidad de la tecnología. Permittiéndonos a nosotros enfocarnos plenamente en la entrega de valor.

**Cons:** Tiende a ser una opción costosa, ya que nos abstrae de todo el músculo requerido para administración y monitoreo de la infraestructura subyacente. Las configuraciones disponibles están limitadas a aquellas que el proveedor haya decidido exponer o disponibilizar.



## Ejemplo de columna PaaS

El modelo PaaS tiende a ser difícil de visualizar cuando no tenemos experiencia. Tomemos un ejemplo para ayudarnos: **Amazon DynamoDB**.

DynamoDB es una base de datos no estructurada, similar a **MongoDB**. Pero mientras que si queremos utilizar MongoDB, debemos instanciar la infraestructura necesaria y luego instalarla; Amazon DynamoDB la podemos consumir, sencillamente, como un servicio.



## Ejemplo de columna PaaS

¿Qué significa esto? Que una vez suscrito al servicio mediante el sitio web de Amazon AWS, nos van a proveer:



- Una interfaz web a través de la cual podremos hacer algunas configuraciones del servicio de Dynamo.
- Una API que nos permitirá configurar los mismos parámetros que la interfaz web, pero de manera programática.
- Y una cadena de conexión que utilizaremos en nuestra aplicación para conectarnos contra la base de datos de Dynamo.

De esta manera, no tuvimos la necesidad de instalar ningún recurso. Simplemente suscribirnos al servicio y la plataforma o tecnología está disponible para nuestro uso, Internet mediante.

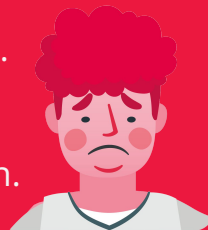
# Las columnas: SaaS

Cuando hablamos de **SaaS** nos referimos a un acrónimo en inglés: *Software as a Service* (en castellano, software como servicio). En este modelo de responsabilidad, el proveedor nos ofrece una aplicación. La mejor manera de entender este modelo de responsabilidad es pensar en ejemplos: **Trello**, **Salesforce** o **Gmail** son ejemplos de SaaS.



**Pros:** Con tan solo una tarjeta de crédito podemos acceder a una aplicación lista para utilizar y resolver una necesidad. Y sin tener que cubrir el costo de la infraestructura y operación de la misma.

**Cons:** Poco a nulo control sobre la configuración del sistema contratado. Dependiendo la flexibilidad de software puede que tengamos que adaptar nuestros procesos a lo que el software permite hacer. No hay control sobre las nuevas funcionalidades del sistema y cuándo se liberan.



DigitalHouse>