

Documento técnico de AWS

Fundamentos de la arquitectura SaaS



Fundamentos de la arquitectura SaaS: Documento técnico de AWS

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas comerciales que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

Resumen e introducción	i
Introducción	1
¿Tiene una buena arquitectura?	2
SaaS como modelo de negocio	3
Se trata de un servicio, no de un producto	5
La motivación inicial	6
Transición a una experiencia unificada	9
Plano de control frente a plano de aplicación	12
Servicios principales	14
Redefinición del concepto de multitenencia	16
Un caso extremo	18
Eliminación del término “inquilino único”	20
Introducción de los conceptos de silo y grupo	20
Silo y grupo de pila completa	22
SaaS frente a proveedor de servicios administrados (MSP)	24
Migración a SaaS	26
Identidad SaaS	30
Aislamiento de inquilinos	31
Particiones de datos	33
Medición, métricas y facturación	35
SaaS B2B y B2C	37
Conclusión	38
Documentación adicional	39
Colaboradores	40
Revisiones del documento	41
Avisos	42
Glosario de AWS	43

Fundamentos de la arquitectura SaaS

Fecha de publicación: 3 de agosto de 2022 ([Revisiones del documento](#))

El alcance, los objetivos y la naturaleza de la administración de una empresa en un modelo de software como servicio (SaaS) pueden ser difíciles de definir. La terminología y los patrones que se utilizan para caracterizar el modelo SaaS varían según su origen. El objetivo de este documento es definir de forma más precisa los elementos fundamentales del software como servicio (SaaS) y describir detalladamente los patrones, términos y sistemas de valores que se aplican al diseño y la entrega de un sistema SaaS en AWS. El objetivo más amplio es proporcionar información básica que ofrezca a los clientes una visión más clara de las opciones que deben tener en cuenta cuando se planteen adoptar un modelo de entrega basado en SaaS.

Este documento está dirigido a desarrolladores y arquitectos de SaaS que están iniciándose en el tema, así como a desarrolladores más experimentados que deseen perfeccionar su comprensión de los conceptos básicos de SaaS. Parte de esta información también puede resultar útil para los propietarios de productos y estrategias de SaaS que deseen familiarizarse más con el panorama del modelo SaaS.

Introducción

El término “software como servicio (SaaS)” se utiliza para describir un modelo de negocio y de entrega. Sin embargo, no todo el mundo entiende qué significa exactamente ser SaaS.

Aunque existe cierto consenso sobre algunos de los pilares fundamentales del SaaS, sigue habiendo algo de confusión en torno a lo que significa ser SaaS. Es normal que haya algunas diferencias en la forma en que los equipos consideran el SaaS. Al mismo tiempo, la falta de claridad sobre los conceptos y términos del SaaS puede crear cierta confusión para aquellas personas que están explorando un modelo de entrega basado en SaaS.

Este documento se centra en explicar la terminología que se utiliza para describir los conceptos básicos de SaaS. Partir de unas ideas comunes en torno a estos conceptos permite hacerse una imagen más precisa de los elementos fundamentales de una arquitectura SaaS, con lo que se utilizará un mismo vocabulario para describir los elementos de dicha arquitectura. Esto resulta especialmente útil cuando se profundiza en otros contenidos basados en estos temas.

Este documento técnico se aparta de los detalles de la arquitectura multitenencia y analiza cómo hemos definido los fundamentos de lo que significa ser SaaS. Idealmente, esto también

proporcionará un conjunto de terminología más preciso, que permita a las organizaciones alinearse más rápidamente con el tipo y la naturaleza de sus soluciones SaaS.

¿Usa Well-Architected?

El [marco de buena arquitectura de AWS](#) le ayuda a entender las ventajas y desventajas de las decisiones que toma al crear sistemas en la nube. Los seis pilares del marco le permitirán aprender las prácticas recomendadas de arquitectura para diseñar y utilizar sistemas fiables, seguros, eficientes, rentables y sostenibles. Mediante [AWS Well-Architected Tool](#), disponible sin costo alguno en la [Consola de administración de AWS](#), puede comparar las cargas de trabajo con estas prácticas recomendadas respondiendo a una serie de preguntas para cada pilar.

En el [enfoque SaaS](#), nos centramos en las prácticas recomendadas para diseñar la arquitectura de las cargas de trabajo de software como servicio (SaaS) en AWS

Para obtener asesoramiento más experto y conocer las prácticas recomendadas para la arquitectura en la nube (implementaciones de arquitectura de referencia, diagramas y documentos técnicos), consulte el [Centro de arquitectura de AWS](#).

SaaS como modelo de negocio

La definición de lo que significa ser SaaS comienza por ponerse de acuerdo sobre un principio clave: SaaS es un modelo de negocio. Esto significa que, por encima de todo, la adopción de un modelo de entrega SaaS está impulsada directamente por un conjunto de objetivos empresariales. Efectivamente, la tecnología se utilizará para alcanzar algunos de esos objetivos, pero la misión de SaaS es establecer un planteamiento y un modelo orientados a un conjunto específico de objetivos empresariales.

Analicemos más detenidamente algunos de los objetivos empresariales clave asociados a la adopción de un modelo de entrega SaaS.

- **Agilidad:** este término resume el objetivo más amplio de SaaS. Las empresas de SaaS de más éxito parten de la idea de que deben estar preparadas para adaptarse continuamente al mercado, a los clientes y a la dinámica competitiva. Las grandes empresas de SaaS están diseñadas para adaptarse continuamente a nuevos modelos de precios, nuevos segmentos de mercado y nuevas necesidades de los clientes.
- **Eficiencia operativa:** las empresas de SaaS necesitan una óptima eficiencia operativa para favorecer la escalabilidad y la agilidad. Esto significa establecer una cultura y unas herramientas que se centren en crear una huella operativa que promueva el lanzamiento frecuente y rápido de nuevas funciones. También significa tener una experiencia única y unificada que permita administrar, controlar e implementar todos los entornos de los clientes de forma colectiva. El modelo SaaS no admite las personalizaciones ni las versiones únicas. Las empresas de SaaS consideran la eficiencia operativa un pilar fundamental para poder impulsar y escalar el negocio con éxito.
- **Incorporación sencilla:** para poder ser más ágiles y favorecer el crecimiento, también es muy importante reducir cualquier complicación en el proceso de incorporación de clientes inquilinos. Esto se aplica a todos los clientes, ya sean de empresa a empresa (B2B) o de empresa a cliente (B2C). Cualquiera que sea el segmento y el tipo de cliente al que se dirija, deberá centrarse en generar valor para sus clientes. El cambio a un modelo centrado en el servicio requiere que la empresa de SaaS se centre en todos los aspectos de la experiencia del cliente, con un énfasis específico en la repetibilidad y la eficiencia del ciclo de vida general de la incorporación.
- **Innovación:** pasarse al SaaS no es simplemente responder a las necesidades de los clientes actuales; se trata de establecer los elementos fundamentales que facilitan la innovación. Desea tener un modelo SaaS que le permita reaccionar y responder a las necesidades de sus clientes.

Sin embargo, también le interesará aprovechar esta agilidad para impulsar futuras innovaciones que le ayuden a abrir nuevos mercados, oportunidades y eficiencias para sus clientes.

- **Respuesta del mercado:** el modelo SaaS se aleja de la noción tradicional de lanzamientos trimestrales y planes a dos años. Se basa en su agilidad para dar a la organización la capacidad de reaccionar y responder a la dinámica del mercado prácticamente en tiempo real. La inversión en los elementos organizativos, técnicos y culturales del SaaS brinda la oportunidad de cambiar la estrategia empresarial en función de las nuevas dinámicas de los clientes y los mercados.
- **Crecimiento:** SaaS es una estrategia empresarial centrada en el crecimiento. Al alinear todos los componentes de la organización en torno a los principios de agilidad y eficiencia, las organizaciones de SaaS pueden orientarse a un modelo de crecimiento. Esto significa establecer los mecanismos que posibiliten y faciliten la rápida adopción de su oferta de SaaS.

Observará que cada uno de estos elementos se centra en un resultado empresarial. Existe una amplia gama de estrategias técnicas y patrones que se pueden utilizar para desarrollar un sistema SaaS. Sin embargo, estas estrategias técnicas no cambian la naturaleza general de la empresa.

Cuando nos reunimos con los responsables de las organizaciones y les preguntamos qué objetivos quieren conseguir con su adopción del SaaS, siempre empezamos por explicarles que se trata de un modelo centrado en la empresa. Las opciones tecnológicas son importantes, pero deben plantearse en el contexto de estos objetivos empresariales. Adoptar un modelo multiinquilino sin contar con una agilidad y una eficiencia operativa adecuadas y una incorporación fluida, por ejemplo, pondría en riesgo el éxito de su empresa de SaaS.

Tras esta introducción, tratemos de concluir con una definición más concisa de SaaS que se ajuste a los principios que hemos descrito:

SaaS es un modelo de empresa y de entrega de software que permite a las organizaciones ofrecer sus soluciones en un modelo centrado en el servicio y de funcionamiento fluido que maximiza el valor para los clientes y los proveedores. Se basa en la agilidad y la eficiencia operativa como pilares de una estrategia empresarial que favorece el crecimiento, el alcance y la innovación.

Los objetivos empresariales deben responder al planteamiento de que la experiencia del servicio es compartida por todos los clientes. Una parte importante del cambio a un modelo SaaS implica dejar de lado las personalizaciones únicas que podrían formar parte de un modelo de software convencional. Cualquier esfuerzo por ofrecer especialización a los clientes generalmente nos aleja de los valores fundamentales que intentamos lograr con el modelo SaaS.

Se trata de un servicio, no de un producto

Adoptar un modelo de “servicio” es algo más que una cuestión de marketing o terminología. Al poner el foco en el servicio, nos alejamos del planteamiento convencional de desarrollo basado en el producto. Mientras que las características y las funciones son muy importantes para todos los productos, SaaS pone más énfasis en la experiencia que los clientes tendrán con el servicio.

¿Qué significa esto? En un modelo centrado en el servicio, se piensa más en la forma en que los clientes se incorporan a su servicio, en la rapidez con la que obtienen valor y en la agilidad con la que se pueden introducir nuevas funciones que satisfagan las necesidades de los clientes. Los detalles relacionados con la forma en que se crea, controla y administra el servicio están fuera de la vista de los clientes.

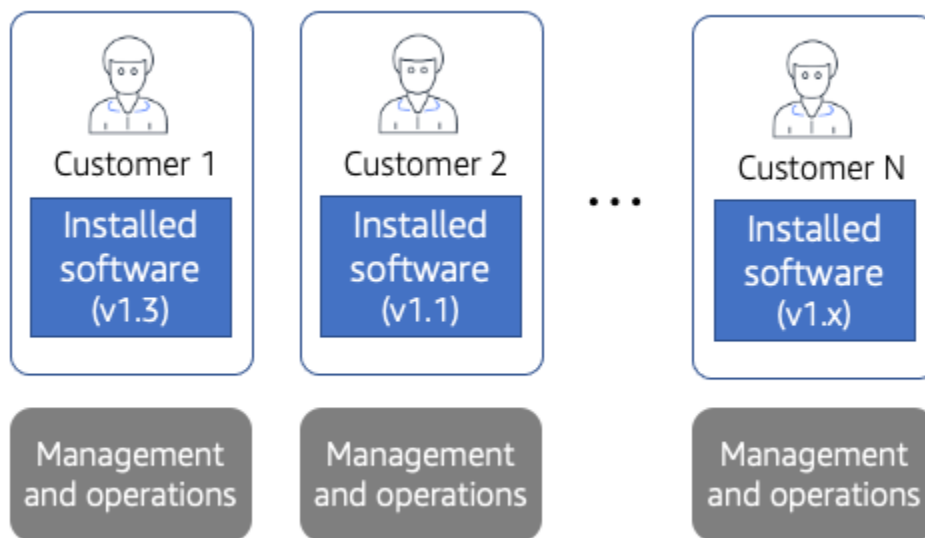
De este modo, pensamos en este servicio SaaS tal como lo haríamos con cualquier otro servicio que pudiéramos consumir. Si vamos a un restaurante, por supuesto que nos importa la comida, pero también nos importa el servicio. La rapidez con la que nos atienden, la frecuencia con la que nos rellenan la copa de agua y la prontitud con la que llega la comida a nuestra mesa: todo sirve para medir la experiencia del servicio. Esta es la misma mentalidad y sistema de valores que deberían guiarnos a la hora de desarrollar un servicio SaaS.

Este modelo de software como servicio debería ser clave a la hora de diseñar los equipos y el servicio de la empresa. Cuando trabaje para implementar este modelo, la experiencia debería tener una importancia igual o superior a las características y funciones. También para la empresa la experiencia deberá ser un elemento clave para garantizar el crecimiento y el éxito a largo plazo de su oferta de SaaS.

La motivación inicial

Para entender el concepto de SaaS, empecemos con una idea bastante simple de lo que intentamos lograr al crear una empresa de SaaS. El mejor punto de partida es analizar cómo se ha creado, administrado y utilizado el software convencional (no SaaS).

El siguiente diagrama proporciona una visión conceptual de cómo varios proveedores han empaquetado y entregado sus soluciones.



Modelo clásico de empaquetado y entrega de soluciones de software

En este diagrama, describimos un conjunto de entornos de cliente. Estos clientes representan las diferentes empresas o entidades que han comprado el software de un proveedor. Básicamente, cada uno de estos clientes funciona en un entorno independiente donde se ha instalado un producto de un proveedor de software.

De este modo, la instalación de cada cliente se trata como un entorno independiente dedicado a ese cliente. Esto significa que los clientes se consideran a sí mismos los propietarios de estos entornos y podrían solicitar una personalización o configuraciones únicas que se adapten a sus necesidades.

Algo habitual en este planteamiento es que los clientes controlan la versión del producto que utilizan. Esto puede ocurrir por varios motivos. Es posible que los clientes se muestren reacios a usar nuevas funciones o estén preocupados por las interrupciones asociadas a la adopción de una nueva versión.

Puede imaginarse cómo puede afectar esta dinámica a la infraestructura operativa del proveedor de software. Cuanto más permita a los clientes tener entornos únicos, más difícil será administrar, actualizar y dar soporte a las distintas configuraciones de cada cliente.

Esta necesidad de entornos únicos a menudo requiere que las organizaciones creen equipos dedicados que proporcionen una experiencia de administración y operaciones separada para cada cliente. Aunque algunos de estos recursos pueden compartirse entre los clientes, este modelo suele generar gastos incrementales por cada nuevo cliente que se incorpora.

Este tipo de modelo en el que cada cliente ejecuta sus soluciones en su propio entorno (en la nube o en las instalaciones) también repercute en los costes. Aunque puede intentar escalar estos entornos, el escalado se limitará a la actividad de un solo cliente. Básicamente, la optimización de los costos se limita a lo que puede lograr en el entorno de un cliente individual. También significa que es posible que necesite estrategias de escalado independientes para adaptarse a las diferencias en la actividad de los distintos clientes.

Inicialmente, algunas empresas de software considerarán que este modelo es eficaz. Ofrecer la personalización única puede servirles como herramienta de ventas, permitiendo a los nuevos clientes imponer requisitos únicos en su entorno. Mientras la empresa no tiene muchos clientes y su crecimiento es moderado, este modelo parece perfectamente sostenible.

Sin embargo, a medida que las empresas comienzan a tener más éxito, las limitaciones de este modelo empiezan a plantear serios problemas. Imagine, por ejemplo, un escenario en el que su empresa alcance un pico de crecimiento significativo y tenga que añadir muchos clientes nuevos a un ritmo rápido. Este crecimiento empezará a aumentar la sobrecarga operativa, la complejidad de la administración y los costos, entre otros muchos problemas.

En última instancia, la sobrecarga y el impacto generales de este modelo pueden poner seriamente en peligro el éxito de una empresa de software. El primer problema podría ser la eficiencia operativa. El aumento del personal y los costos asociados con la captación de clientes comienzan a reducir los márgenes de la empresa.

Sin embargo, los temas operativos son solo una parte del problema. El verdadero problema de este modelo es que, a medida que se amplía, comienza a afectar directamente a la capacidad de la empresa de lanzar nuevas funciones y seguir el ritmo del mercado. Cuando cada cliente tiene su propio entorno, los proveedores deben hacer frente a una serie de requisitos sobre actualización, migración y clientes para tratar de introducir nuevas funciones en su sistema.

Por lo general, esto lleva a ciclos de lanzamiento más largos y complejos, lo que tiende a reducir la cantidad de lanzamientos que se realizan cada año. Y lo que es más importante, esta complejidad

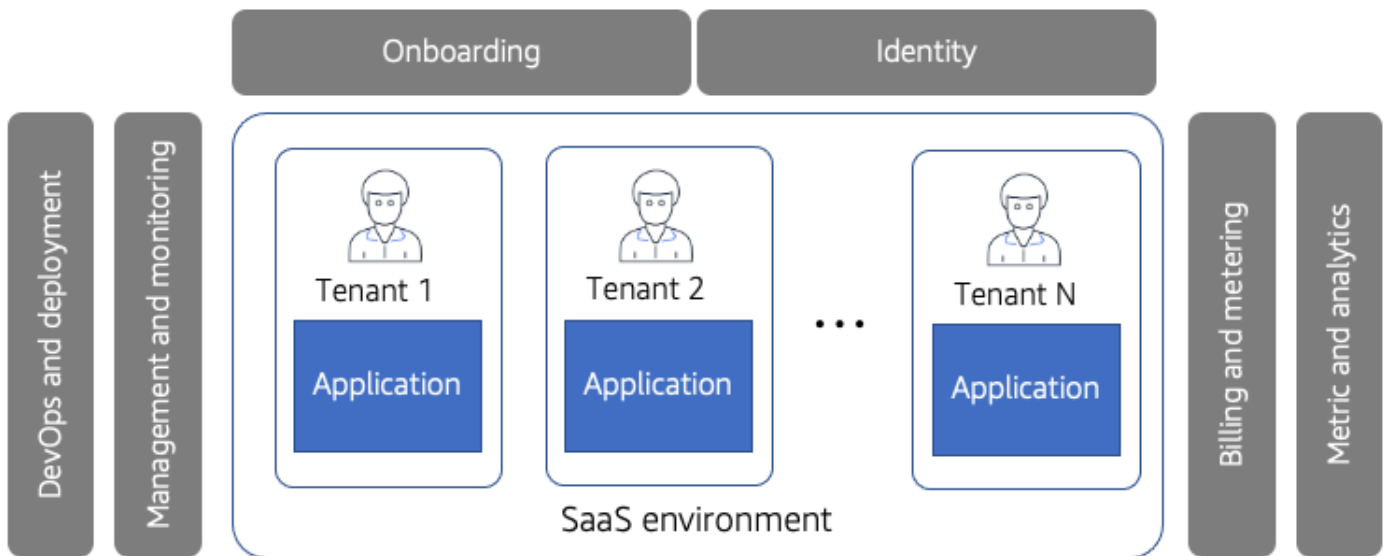
hace que los equipos dediquen más tiempo a analizar cada nueva función mucho antes de que se lance al cliente. Los equipos comienzan a centrarse más en validar las nuevas funciones y dedican menos tiempo a agilizar la entrega. La sobrecarga que supone el lanzamiento de nuevas funciones aumenta tanto que los equipos acaban centrándose más en la mecánica de las pruebas y menos en la incorporación de funciones que aportan innovación a la oferta.

En este modo más lento y cauteloso, los equipos suelen tener ciclos largos, lo que aumenta cada vez más el tiempo que pasa entre el inicio de una idea y el momento en que llega a las manos de los clientes. En general, esto puede dificultar la capacidad de reacción ante la dinámica del mercado y las presiones competitivas.

Transición a una experiencia unificada

Para dar respuesta a los problemas del modelo de software clásico, las organizaciones recurren a un modelo que les permite crear una experiencia única y unificada para gestionar y controlar sus clientes de forma colectiva.

El siguiente diagrama muestra los conceptos de un entorno en el que todos los clientes se administran, incorporan, facturan y controlan mediante un modelo compartido.



Conceptos de un entorno en el que todos los clientes se administran, incorporan, facturan y controlan mediante un modelo compartido

A primera vista, esto puede parecer muy similar al modelo anterior. Sin embargo, al profundizar un poco más, vemos que existen diferencias fundamentales y significativas entre estos dos enfoques.

En primer lugar, puede observar que los entornos de los clientes se denominan ahora "inquilinos". Esta noción de inquilino es fundamental para el SaaS. La idea básica es que tiene un único entorno SaaS y que cada uno de sus clientes viene a ser como un inquilino de ese entorno, donde consume los recursos que necesita. Un inquilino puede ser una empresa con muchos usuarios o un usuario individual.

Para entender mejor la idea de un inquilino, imagínese un edificio de viviendas o un centro comercial. El espacio de estos edificios se alquila a inquilinos individuales. Los inquilinos utilizan algunos de los recursos compartidos del edificio (agua, electricidad, etc.) y pagan por el consumo que realizan.

Los inquilinos de SaaS siguen un patrón similar. Usted cuenta con la infraestructura de su entorno SaaS y tiene unos inquilinos que consumen la infraestructura de ese entorno. La cantidad de recursos que consume cada inquilino puede variar. Estos inquilinos también se administran, facturan y controlan de forma colectiva.

Si vuelve a examinar el diagrama, verá cómo funciona en la práctica la noción de tenencia. Aquí, los inquilinos ya no tienen su propio entorno. En lugar de eso, todos los inquilinos se alojan y administran dentro de los muros de un entorno SaaS colectivo.

El diagrama también incluye una gama de servicios compartidos situados alrededor de su entorno SaaS. Estos servicios son globales para todos los usuarios del entorno SaaS. Esto significa que todos los usuarios de este entorno comparten, por ejemplo, los servicios de incorporación e identidad. Lo mismo ocurre con la administración, las operaciones, el despliegue, la facturación y las métricas.

Esta idea de un conjunto unificado de servicios que se aplican a todos los inquilinos es fundamental en el modelo SaaS. Al compartir estos servicios, puede solucionar algunos de los problemas asociados al modelo clásico descrito anteriormente.

Otro elemento clave de este diagrama, aunque quizá no sea tan obvio, es que todos los inquilinos de este entorno ejecutan la misma versión de la aplicación. En este modelo no se ejecutan versiones independientes y únicas para cada cliente. Hacer que todos los inquilinos ejecuten la misma versión del software es uno de los atributos distintivos fundamentales de un entorno SaaS.

Como todos los clientes utilizan la misma versión del producto, ya no tiene que hacer frente a muchos de los problemas que plantean los modelos clásicos de software instalado. En el modelo unificado, las nuevas funciones se pueden implementar en todos los inquilinos mediante un único proceso compartido.

Este enfoque le permite utilizar un único panel operativo para administrar y controlar a todos los inquilinos. De esta forma, puede administrar y monitorizar a sus inquilinos a través de una experiencia operativa común, lo que permite agregar nuevos inquilinos sin aumentar los gastos operativos. Esta es una parte fundamental de la propuesta de valor de SaaS, que brinda a los equipos la capacidad de reducir los gastos operativos y mejorar la agilidad de la organización a nivel general.

Imagínese lo sencillo que sería añadir 100 o 1000 clientes nuevos en este modelo. En lugar de preocuparse por cómo podría la incorporación de estos nuevos clientes reducir los márgenes y añadir complejidad, puede ver este crecimiento como una oportunidad.

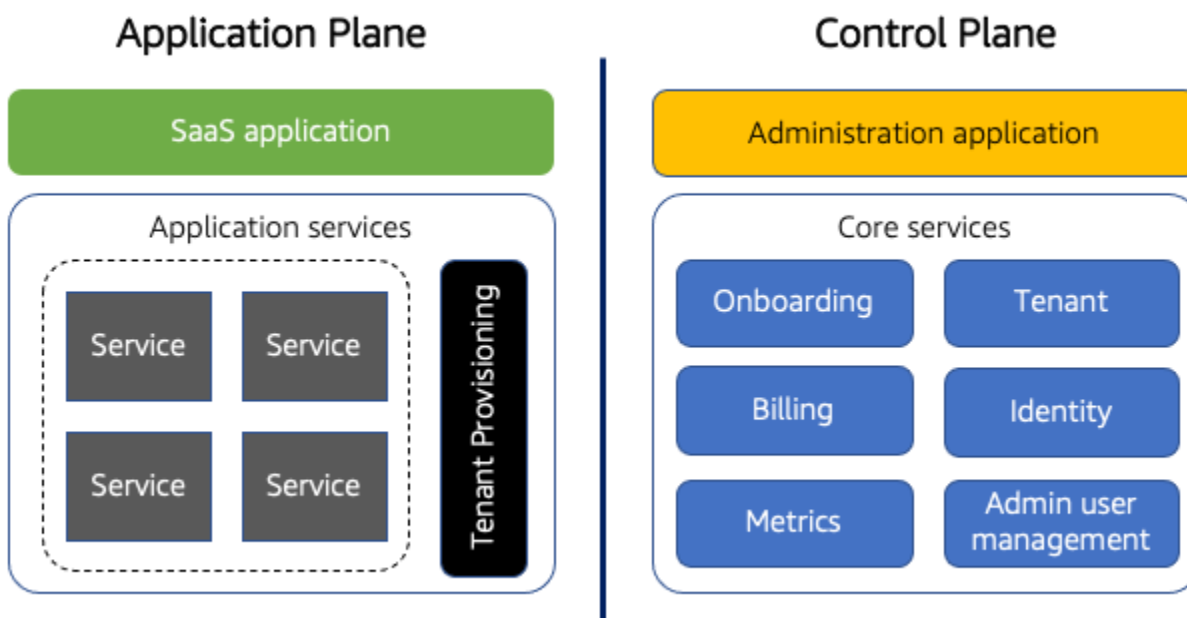
En general, SaaS se centra en la forma en que se implementa la aplicación que se halla en el medio de este modelo. Las empresas suelen dar más importancia a cómo se almacenan los datos, cómo se comparten los recursos, etc. Sin embargo, la realidad es que, si bien estos detalles son muy relevantes, hay muchas maneras de diseñar una aplicación de forma que pueda presentarse como una solución SaaS para los clientes.

Lo fundamental es el objetivo más amplio de tener una experiencia única y unificada que englobe los entornos de los inquilinos. Contar con esta experiencia compartida es lo que le permite impulsar el crecimiento, la agilidad y la eficiencia operativa que se asocian a los objetivos generales de una empresa de SaaS.

Plano de control frente a plano de aplicación

El diagrama anterior proporciona una vista conceptual de las ideas clave de la arquitectura SaaS. Ahora profundizaremos en estos conceptos y definiremos de forma precisa las capas que componen un entorno de SaaS. Tener esta imagen más clara de los límites entre los conceptos de SaaS ayudará a entender mejor los componentes de una solución SaaS.

En el siguiente diagrama el entorno SaaS se divide en dos planos distintos. A la derecha está el plano de control. Esta parte del diagrama incluye todas las funciones y los servicios que se utilizan para incorporar, autenticar, administrar, controlar y analizar un entorno multiinquilino.



Plano de control frente a plano de aplicación

Este plano de control es fundamental en cualquier modelo de SaaS multiinquilino. Todas las soluciones SaaS, independientemente de la implementación y el esquema de aislamiento de las aplicaciones, deben incluir aquellos servicios que le permitan administrar y controlar sus inquilinos mediante una experiencia única y unificada.

Hemos dividido el plano de control en dos elementos distintos. Los servicios principales representan el conjunto de los servicios que se utilizan para organizar su experiencia multiinquilino. Hemos incluido algunos ejemplos de los servicios principales más comunes, aunque pueden variar para cada solución SaaS.

También puede observar que mostramos una aplicación de administración independiente. Se trata de la aplicación (una aplicación web, una interfaz de línea de comandos o una API) que podría usar un proveedor de SaaS para administrar su entorno multiinquilino.

Es importante tener en cuenta que el plano de control y sus servicios no son en realidad multiinquilino. Su objetivo no consiste en proporcionar los atributos funcionales reales de la aplicación SaaS (que sí debe ser multiinquilino). Si analiza cualquiera de los servicios principales, por ejemplo, no encontrará el aislamiento de inquilinos ni los demás componentes que forman parte de la funcionalidad de una aplicación multiinquilino. Estos servicios son globales para todos los inquilinos.

En la parte izquierda del diagrama se muestra el plano de la aplicación de un entorno SaaS. Aquí es donde reside la funcionalidad multiinquilino de su aplicación. Los elementos del diagrama deben ser algo imprecisos, ya que cada solución se puede implementar y descomponer de forma diferente según las necesidades del dominio, la tecnología empleada, etc.

El dominio de la aplicación se divide en dos elementos. Por un lado, la aplicación SaaS, que representa la experiencia o la aplicación de inquilino de su solución. Se trata de los elementos mediante los que los inquilinos interactúan con su aplicación SaaS. Por otro lado, están los servicios de backend que representan la lógica empresarial y los elementos funcionales de una solución SaaS. Podrían ser microservicios o algún otro paquete de servicios de su aplicación.

También puede observar que hemos puesto aparte el aprovisionamiento. El objetivo es resaltar el hecho de que cualquier aprovisionamiento de recursos para los inquilinos durante la incorporación formaría parte de este dominio de aplicación. Algunos podrían argumentar que esto pertenece al plano de control. Sin embargo, lo hemos colocado en el dominio de la aplicación, ya que los recursos que debe aprovisionar y configurar están conectados más directamente con los servicios que se crean y configuran en el plano de la aplicación.

Al presentar dos planos distintos, es más fácil hacerse una idea general de una arquitectura SaaS. Y lo que es más importante, ayuda a destacar la necesidad de un conjunto de servicios que estén completamente fuera del ámbito de la funcionalidad de la aplicación.

Servicios principales

En el plano de control que indicamos anteriormente se mencionan una serie de servicios principales que representan los servicios más comunes que se suelen utilizar para incorporar, administrar y operar un entorno SaaS. Puede ser útil describir en más detalle el papel de algunos de estos servicios para entender su alcance y objetivo en un entorno SaaS. A continuación, incluimos un breve resumen de cada uno de estos servicios:

- **Incorporación:** cada solución de SaaS debe proporcionar un mecanismo sencillo para incluir nuevos inquilinos en el entorno de SaaS. Puede ser una página de registro en modo de autoservicio o una experiencia administrada internamente. Cualquiera que sea la opción escogida, una solución de SaaS debería hacer todo lo posible por eliminar cualquier dificultad interna o externa de esta experiencia y garantizar la estabilidad, la eficiencia y la repetibilidad de este proceso. Desempeña un papel esencial para favorecer el crecimiento y la escalabilidad de una empresa de SaaS. Por lo general, en torno a este servicio se organizan otros servicios para crear usuarios, inquilinos, políticas de aislamiento, aprovisionamiento y recursos por inquilino.
- **Inquilino:** el servicio de inquilino proporciona una forma de centralizar las políticas, los atributos y el estado de los inquilinos. La idea principal es que los inquilinos no son usuarios individuales. De hecho, es probable que un inquilino esté asociado a muchos usuarios.
- **Identidad:** los sistemas SaaS necesitan una forma precisa de conectar a los usuarios con los inquilinos que tenga en cuenta el contexto de los inquilinos en la experiencia de autenticación y autorización de sus soluciones. Esto influye tanto en la experiencia de incorporación como en la administración general de los perfiles de usuario.
- **Facturación:** con la adopción de SaaS, las organizaciones a menudo cambian a nuevos modelos de facturación. También pueden evaluar la integración con proveedores de facturación externos. Este servicio principal se centra fundamentalmente en facilitar la incorporación de nuevos inquilinos y en recoger datos sobre consumo y actividad, que se utilizan para generar las facturas para los inquilinos.
- **Métricas:** los equipos de SaaS dependen en gran medida de su capacidad para capturar y analizar métricas detalladas, que permiten conocer mejor cómo usan los inquilinos el sistema, cómo consumen los recursos y cómo conectan sus propios sistemas. Estos datos se utilizan para diseñar las estrategias operativas, empresariales y de producto.
- **Gestión de usuarios administradores:** los sistemas SaaS deben admitir tanto usuarios inquilinos como usuarios administradores. Los usuarios administradores son los encargados de la

administración de un proveedor de SaaS. Se conectan a su experiencia operativa para monitorizar y administrar su entorno SaaS.

Redefinición del concepto de multitenencia

Los términos multitenencia y SaaS suelen estar estrechamente relacionados. En algunos casos, las organizaciones describen el SaaS y la multitenencia como una misma cosa. Si bien esto puede parecer natural, equiparar el SaaS con la multitenencia suele llevar a los equipos a adoptar una visión puramente técnica del SaaS cuando, en realidad, el SaaS es más un modelo de negocio que una estrategia de arquitectura.

Para entender mejor este concepto, empezaremos explicando la visión clásica de la multitenencia. En esta visión centrada exclusivamente en la infraestructura, la multitenencia se utiliza para describir la forma en que los inquilinos comparten los recursos con el fin de aumentar la agilidad y la rentabilidad.

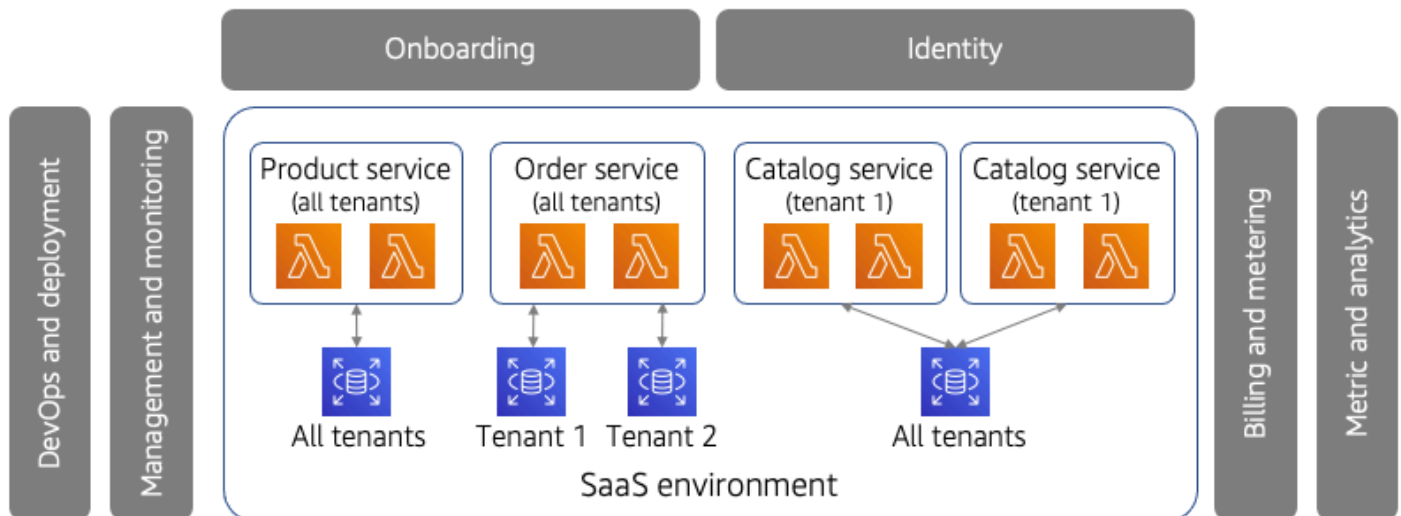
Supongamos, por ejemplo, que tiene un microservicio o una instancia de [Amazon Elastic Compute Cloud](#) (Amazon EC2) que consumen varios inquilinos de un sistema SaaS. Podríamos considerar que este servicio se ejecuta en un modelo de multitenencia, ya que los inquilinos comparten el uso de la infraestructura del servicio.

El problema de esta definición es que vincula demasiado directamente la noción técnica de multitenencia al SaaS. En esta definición se asume que la característica que define un modelo como SaaS es tener una infraestructura compartida multitenencia. Esta visión del SaaS comienza a fallar cuando observamos las diversas formas en que el SaaS se implementa en diferentes entornos.

En el siguiente diagrama se muestra un sistema SaaS que expone algunos de los desafíos a los que nos enfrentamos a la hora de definir el concepto de multitenencia.

Aquí puede ver el modelo SaaS clásico descrito anteriormente, con una serie de servicios de aplicaciones rodeados de servicios compartidos que le permiten administrar y controlar sus inquilinos de forma colectiva.

La novedad son los microservicios que hemos incluido. El diagrama incluye tres ejemplos de microservicios: productos, pedidos y catálogos. Si observa detenidamente el modelo de tenencia de cada uno de estos servicios, observará que todos emplean patrones de tenencia ligeramente diferentes.



SaaS y multitenencia

El servicio de productos comparte todos sus recursos (cómputo y almacenamiento) con todos los inquilinos. Esto se ajusta a la definición clásica de multitenencia. Sin embargo, si observa el servicio de pedidos, verá que tiene un recurso de cómputo compartido, pero un almacenamiento independiente para cada inquilino.

El servicio de catálogos añade otra variante en la que el recurso de cómputo es independiente para cada inquilino (una implementación de microservicio independiente para cada uno), pero el almacenamiento es compartido por todos los inquilinos.

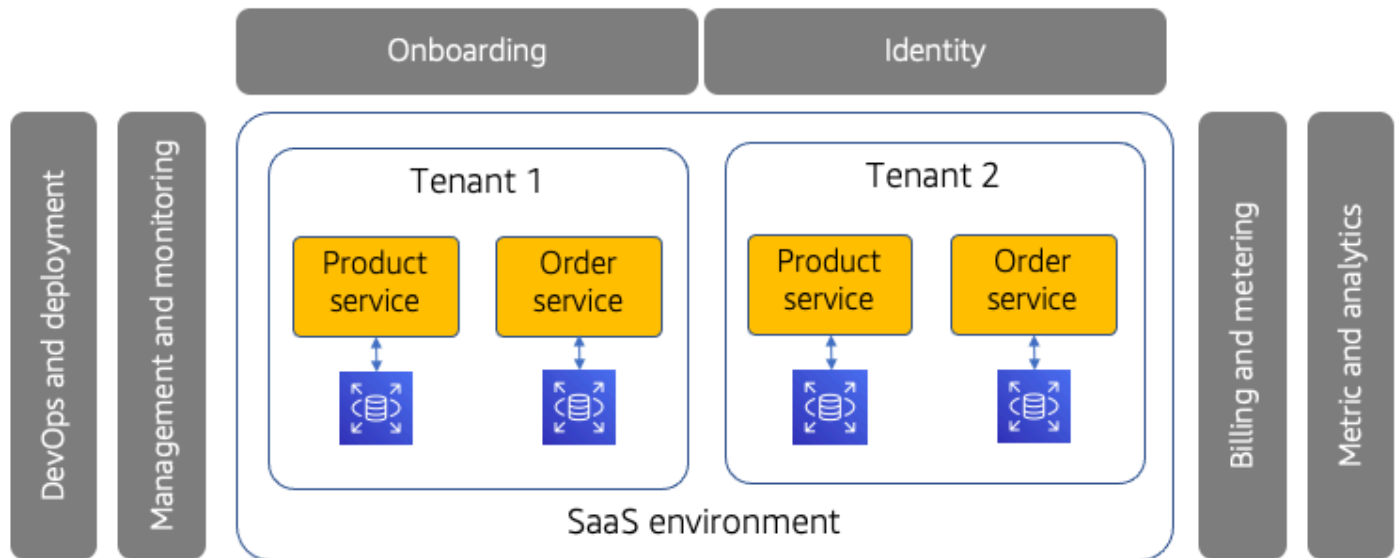
Las variaciones de este tipo son comunes en los entornos SaaS. [Vecinos ruidosos](#), modelos organizados en niveles o necesidades de aislamiento son algunas de las razones por las que podría decidir compartir o aislar partes de su solución SaaS de forma selectiva.

Teniendo en cuenta estas variaciones (y muchas otras posibilidades), resulta más difícil determinar cómo utilizar el término multiinquilino para definir este entorno. En general, en lo que respecta al cliente, se trata de un entorno multiinquilino. Sin embargo, si utilizamos la definición más técnica, algunas partes de este entorno son multiinquilino y otras no.

Por eso se hace necesario dejar de utilizar el término multiinquilino para definir los entornos SaaS. Aunque podemos hablar de cómo se implementa la multitenencia en una aplicación, debemos evitar usarla para definir una solución como SaaS. Si se va a usar el término multiinquilino, tiene más sentido emplearlo para describir todo el entorno SaaS como multiinquilino, sabiendo que es posible que algunas partes de la arquitectura se compartan y otras no. En general, este entorno se administra y controla según un modelo multiinquilino.

Un caso extremo

Para entender mejor el concepto de tenencia, veamos un modelo SaaS que tiene inquilinos que comparten cero recursos. El siguiente diagrama es un ejemplo de entorno SaaS que utilizan algunos proveedores de SaaS.



Pila por inquilino

En este diagrama, puede ver que todavía tenemos un entorno común alrededor de estos inquilinos. Sin embargo, cada inquilino se implementa con una colección de recursos dedicada. Los inquilinos de este modelo no comparten nada.

Este ejemplo pone en cuestión qué significa adoptar un modelo multiinquilino. ¿Se puede decir que es un entorno multiinquilino aunque no se comparta ninguno de los recursos? Los inquilinos que consumen este sistema tienen las mismas expectativas que tendría cualquier inquilino de un entorno SaaS con recursos compartidos. De hecho, es posible que no sepan cómo se despliegan sus recursos en el entorno SaaS.

Aunque estos inquilinos operan en una infraestructura organizada en silos, se administran y controlan de forma colectiva. Comparten una experiencia unificada de incorporación, identidad, métricas, facturación y operaciones. Además, cuando se lanza una nueva versión, se implementa para todos los usuarios. Para que este sistema funcione, no puede permitir personalizaciones únicas para inquilinos individuales.

Este ejemplo extremo proporciona un buen modelo para probar el concepto de SaaS multiinquilino. Aunque es posible que no consiga el nivel de eficiencia de una infraestructura compartida, se trata de un entorno SaaS multiinquilino totalmente válido. En el caso de algunos clientes, el ámbito de su empresa puede requerir que algunos de sus clientes o todos ellos utilicen este modelo. Eso no significa que no sean SaaS. Si utilizan estos servicios compartidos y todos los inquilinos usan la misma versión, este modelo sigue cumpliendo los principios fundamentales del SaaS.

Teniendo en cuenta estos parámetros y esta definición más amplia de SaaS, es obvio que hace falta reformular el uso del término multiinquilino. Tiene más sentido definir como multiinquilino cualquier sistema SaaS que se administre y controle de forma colectiva. Después, se puede recurrir a una terminología más específica para describir cómo se comparten o se dedican los recursos en la implementación de una solución SaaS.

Eliminación del término “inquilino único”

Por asociación con el término multiinquilino, es comprensible que algunas personas quieran usar el término inquilino único para describir los entornos SaaS. Sin embargo, teniendo en cuenta el contexto descrito anteriormente, el término inquilino único crea confusión.

¿El diagrama anterior muestra un entorno de inquilino único? Si bien técnicamente cada inquilino tiene su propia pila, siguen estando controlados y administrados en un modelo multiinquilino. Esta es la razón por la que generalmente se evita el término inquilino único. Todos los entornos se caracterizan por ser multiinquilinos, ya que todos son variaciones de un modelo de tenencia, donde algunos o todos los recursos son compartidos o dedicados.

Introducción de los conceptos de silo y grupo

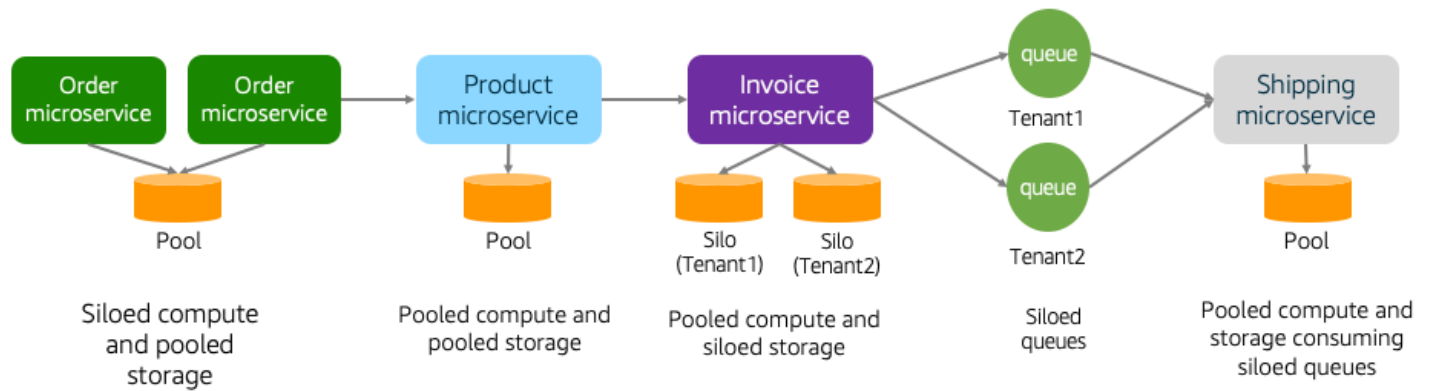
En respuesta a esta gran variedad de modelos y dados los desafíos que plantea el término multitenencia, hemos introducido una terminología que nos permite capturar y describir con mayor precisión los diferentes modelos que se utilizan a la hora de diseñar un sistema SaaS.

Dos términos que utilizamos para describir el uso de los recursos en un entorno SaaS son silo y grupo. Estos términos nos permiten etiquetar la naturaleza de los entornos SaaS, utilizando el término multiinquilino como descripción general que se puede aplicar a cualquier número de modelos subyacentes.

En el nivel más básico, el término silo se utiliza para describir situaciones en las que un recurso se dedica a un inquilino determinado. Por el contrario, el modelo de grupo se usa para describir escenarios en los que los inquilinos comparten un recurso.

Al analizar cómo se utilizan los términos silo y grupo, es importante dejar claro que no expresan conceptos absolutos. El silo y el grupo podrían aplicarse a una pila completa de recursos de un inquilino o solo a partes concretas de un entorno SaaS general. Por lo tanto, si decimos que algún recurso utiliza un modelo de silos, no significa que todos los recursos de ese entorno estén aislados. Lo mismo ocurre con la forma en que usaríamos el término agrupado.

El siguiente diagrama es un ejemplo de cómo los modelos agrupados y distribuidos en silos se utilizan de forma más detallada en un entorno SaaS:



Modelos de silos y grupos

Este diagrama incluye una serie de muestras que ilustran la naturaleza más específica de los modelos de silos y grupos. Si lo examina de izquierda a derecha, puede ver que empezamos con un microservicio de pedidos. Este microservicio tiene el servicio de cómputo en silo y el almacenamiento en grupo. Interactúa con un servicio de productos que tiene cómputo y almacenamiento en grupo.

A continuación, el servicio de productos interactúa con un microservicio de facturas que tiene el cómputo en grupo y el almacenamiento en silo. Este servicio envía los mensajes mediante colas al servicio de envío. Las colas se despliegan en un modelo de silos.

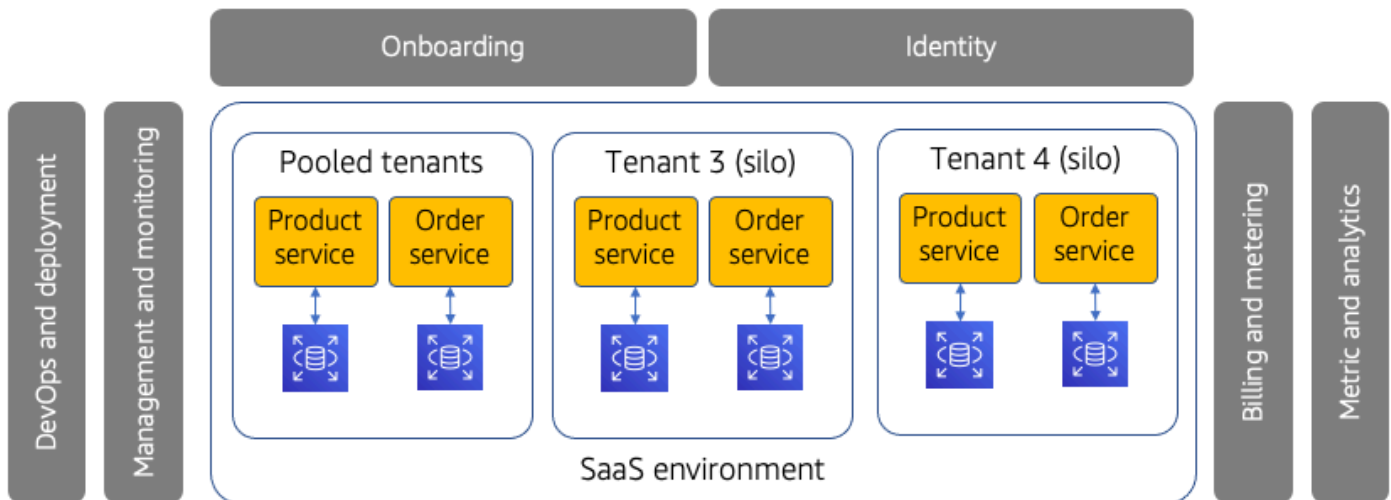
Por último, el microservicio de envío adquiere los mensajes de las colas distribuidas en silo. Utiliza cómputo y almacenamiento en grupo.

Aunque esto puede parecer un poco complicado, el objetivo es resaltar la naturaleza granular de los conceptos de silo y grupo. Al diseñar y desarrollar su solución SaaS, debería tomar estas decisiones sobre silos y grupos en función de las necesidades de su ámbito y de sus clientes.

Los vecinos ruidosos, el aislamiento, la organización en niveles y muchos otros motivos pueden influir en la forma y el momento de adoptar un modelo de silos o de grupos.

Silo y grupo de pila completa

El silo y el grupo también se pueden usar para describir una pila completa de SaaS. En este enfoque, todos los recursos de un inquilino se despliegan de forma dedicada o compartida. El siguiente diagrama proporciona un ejemplo de cómo podría implementarse en un entorno SaaS.



Modelos de silo y grupo de pila completa

En este diagrama, se puede ver que hay tres modelos diferentes para los despliegues de inquilinos de pila completa. En primer lugar, se ve que hay un entorno de grupo de pila completa. Los inquilinos de este grupo comparten todos los recursos (cómputo, almacenamiento, etc.).

Las otras dos pilas que se muestran representan entornos de inquilinos de pila completa distribuidos en silos. En este caso, el inquilino 3 y el inquilino 4 tienen cada uno sus propias pilas dedicadas, y ninguno de los recursos se comparte con otros inquilinos.

Esta combinación de modelos de silos y grupos en el mismo entorno de SaaS no es tan atípica. Imagine, por ejemplo, que tiene un conjunto de inquilinos de nivel básico que pagan un precio moderado por usar su sistema. Estos inquilinos se ubican en un entorno agrupado.

Además, es posible que tenga inquilinos de nivel premium que estén dispuestos a pagar más por el privilegio de ejecutar las operaciones en un silo. Estos clientes se despliegan con pilas separadas (como se muestra en el diagrama).

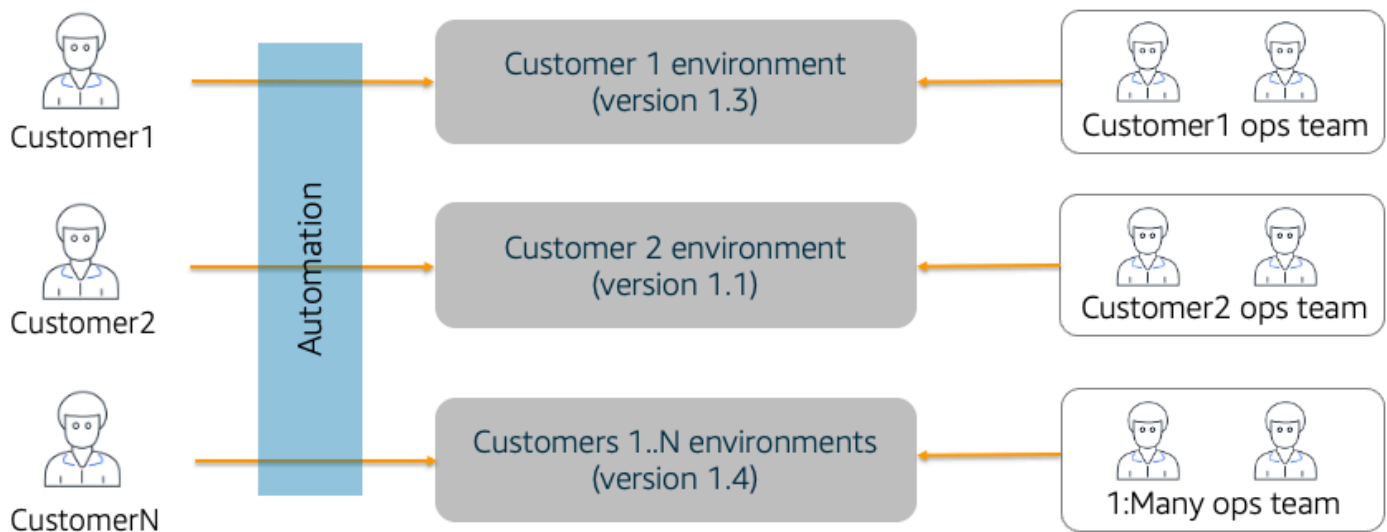
Incluso en este modelo, en el que es posible que se haya permitido a los inquilinos ejecutar las operaciones en su propio silo de pila completa, sería fundamental que en estos silos no se permitiera

ninguna variación única ni personalización para estos inquilinos. En todos los aspectos, cada una de estas pilas debería tener la misma configuración y ejecutar la misma versión del software. Cuando se lanza una nueva versión, esta se despliega en el entorno de los inquilinos agrupados y en cada uno de los entornos de silos.

SaaS frente a proveedor de servicios administrados (MSP)

También existe cierta confusión en torno a las líneas que separan los modelos de SaaS y de proveedor de servicios administrados (MSP). Si observamos un modelo de MSP, puede parecer que tiene algunos objetivos similares a los del modelo de SaaS.

Sin embargo, si profundizamos un poco más en el modelo de MSP, comprobaremos que MSP y SaaS son en realidad diferentes. El siguiente diagrama proporciona una vista conceptual de un entorno MSP.



Modelo de proveedor de servicios administrados (MSP)

Este diagrama representa un enfoque del modelo de MSP. A la izquierda, se muestran los clientes que utilizan el modelo de MSP. En general, el enfoque en este caso consistiría en utilizar la automatización disponible para aprovisionar el entorno de cada cliente e instalar el software para ese cliente.

A la derecha se muestra el esquema operativo que el MSP utilizaría para dar soporte a estos entornos de clientes.

Es importante tener en cuenta que el MSP suele instalar y administrar una versión del producto que un cliente determinado desea ejecutar. Es posible que todos los clientes utilicen la misma versión, pero esto no suele ser necesario en un modelo de MSP.

La estrategia general consiste en simplificar la vida del proveedor de software al hacerse cargo de la instalación y la administración de estos entornos. Si bien esto simplifica el trabajo del proveedor, no se relaciona directamente con los valores y planteamientos que son esenciales para una oferta de SaaS.

El objetivo es liberarse de la responsabilidad de la administración. Hacer esa transición no es simplemente conseguir que todos los clientes utilicen la misma versión con una experiencia única y unificada de administración y operaciones. Por el contrario, MSP suele permitir versiones independientes y suele tratar cada uno de estos entornos como sistemas con funcionamientos separados.

Sin duda hay áreas en las que el modelo de MSP podría empezar a solaparse con el de SaaS. Si el MSP básicamente requiriese que todos los clientes ejecutaran la misma versión y pudiera incorporar, administrar, controlar y facturar a todos los inquilinos de forma centralizada a través de una sola experiencia, podría empezar a ser más SaaS que MSP.

La cuestión general es que automatizar la instalación de los entornos no es lo mismo que tener un entorno SaaS. Solo si se agregan todas las demás advertencias abordadas anteriormente, este modelo será auténticamente SaaS.

Si nos alejamos de los aspectos tecnológicos y operativos, la línea que separa los modelos de MSP y SaaS se hace aún más clara. En general, como empresa de SaaS, el éxito de su oferta depende de su capacidad para implicarse profundamente en todos los componentes de la experiencia.

Por lo general, esto significa estar al tanto de la experiencia de incorporación, comprender cómo afectan los eventos operativos a los inquilinos, hacer un seguimiento de las métricas y los análisis clave y estar cerca del cliente. En un modelo de MSP, donde todo esto se transfiere a otra persona, es posible que acabe perdiéndose los detalles clave que son fundamentales para controlar una empresa SaaS.

Migración a SaaS

Muchos de los proveedores que adoptan SaaS migran desde un modelo de software instalado tradicional (descrito anteriormente). Para estos proveedores, es especialmente importante tener una buena alineación con los principios básicos del SaaS.

Aquí, nuevamente, es donde puede haber cierta confusión sobre lo que significa migrar a un modelo SaaS. Algunas personas, por ejemplo, ven la migración a la nube como una migración a SaaS. Otros consideran que incorporar la automatización a su proceso de instalación y aprovisionamiento es una forma de lograr la migración.

Debemos reconocer que cada organización puede tener un punto de partida diferente, contar con una base heredada distinta y probablemente deba hacer frente a sus propios retos en cuestión de competencia y mercado. Esto significa que cada migración será diferente.

Sin embargo, aunque cada ruta es diferente, hay algunas áreas en las que hay desacuerdo en torno a los principios básicos que rigen las estrategias de migración. Una buena alineación en torno a los conceptos y principios puede tener un impacto significativo en el éxito general de la migración a SaaS.

Teniendo en cuenta los conceptos descritos anteriormente, resulta obvio que la transición a SaaS comienza con la estrategia y los objetivos empresariales. Este aspecto puede quedar relegado en la fase de migración cuando existe presión por implantar un modelo SaaS lo más rápido posible.

En estos casos, las organizaciones suelen ver la migración principalmente como una cuestión técnica. Sin embargo, cada migración a SaaS debería comenzar con una visión clara de los clientes objetivo, la experiencia de servicio, los objetivos operativos, etc. Tener claro cómo debe ser su empresa de SaaS tendrá un profundo impacto en la forma, las prioridades y la ruta que tome para migrar su solución a SaaS.

Tener este enfoque bien definido desde el principio de la migración ayuda a sentar las bases para migrar la tecnología y la empresa como parte del proceso de transición a SaaS. Al diseñar su plan de ruta, concéntrese en aquellas cuestiones que puedan definir mejor hacia dónde quiere dirigirse.

La tabla siguiente muestra las diferencias entre un planteamiento de migración basado en cuestiones técnicas y uno de migración basado en la empresa.

Tabla 1: Migración técnica frente a migración centrada en la empresa

Planteamiento de migración que prioriza las cuestiones técnicas	Planteamiento de migración centrada en la empresa
¿Cómo aislamos los datos de los inquilinos?	¿Cómo puede el SaaS ayudarnos a hacer crecer nuestra empresa?
¿Cómo asociamos a los usuarios con los inquilinos?	¿A qué segmentos nos dirigimos?
¿Cómo evitamos las condiciones de vecino ruidoso?	¿Cuál es el tamaño y el perfil de estos segmentos?
¿Cómo hacemos las pruebas A/B?	¿Qué niveles tendremos que admitir?
¿Cómo escalamos en función de la carga de inquilinos?	¿A qué experiencia de servicio nos dirigimos?
¿Qué proveedor de facturación debemos utilizar?	¿Cuál es nuestra estrategia de precios y empaquetado?

El ejemplo de la izquierda muestra cómo sería una migración que prioriza las cuestiones técnicas. El equipo de ingeniería está muy centrado en abordar las cuestiones habituales de los sistemas multiinquilino que, sin duda, son importantes para cualquier arquitectura SaaS.

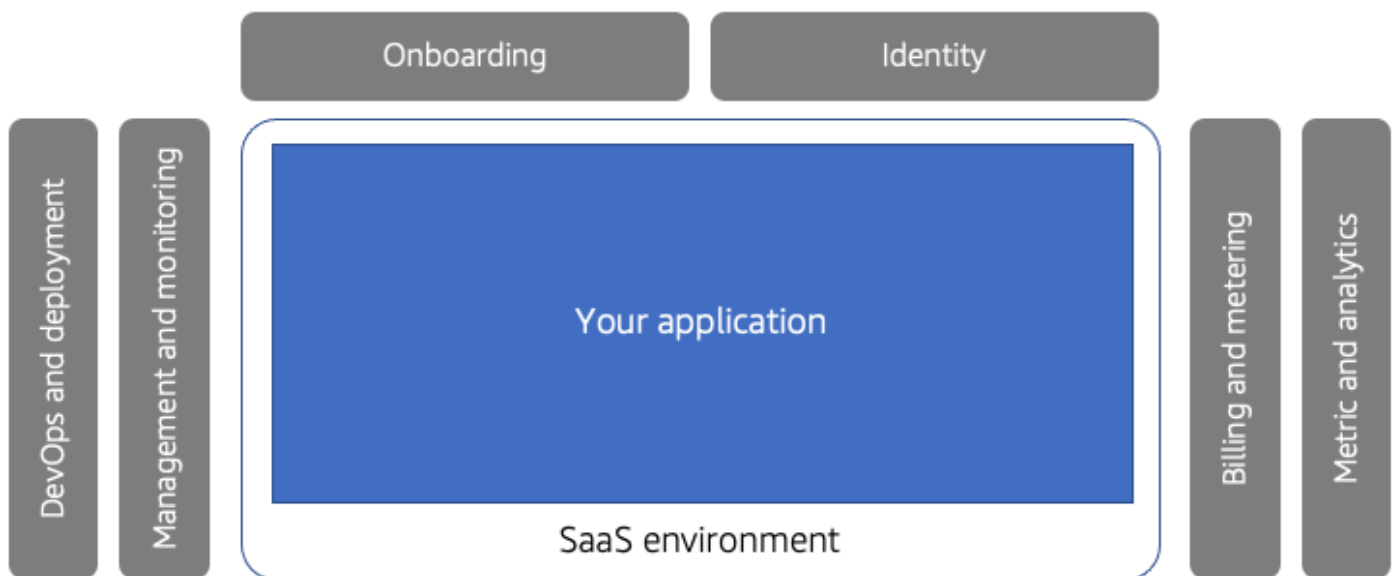
El problema es que las respuestas a muchas de las preguntas de la izquierda suelen estar directamente influenciadas por las respuestas a las preguntas de la derecha. Muy probablemente, cualquiera que esté analizando cómo realizar la migración estará al tanto de estas cuestiones. Sin embargo, la realidad es que muchas organizaciones comienzan por buscar la rentabilidad económica y operativa como primer paso, dando por sentado que las cuestiones relativas a la empresa se solucionarán por sí solas.

Dentro de esta estrategia de migración, también puede haber confusión acerca de cómo se podría transformar el entorno heredado para adaptarlo a un modelo de SaaS. Aquí también hay una gran cantidad de opciones para migrar a SaaS. Sin embargo, existe un sistema común de valores que generalmente defendemos para cualquier migración.

Cuando anteriormente abordamos los principios de SaaS, mostramos los diferentes patrones y términos que se utilizan para describir los entornos de SaaS. Un tema común en todas esas

soluciones es la idea de tener servicios compartidos en torno a la aplicación. La identidad, la incorporación, las métricas y la facturación son elementos comunes fundamentales en cualquier entorno SaaS.

Ahora, al analizar la migración, comprobará que estos mismos servicios compartidos desempeñan un papel clave en cualquier migración. El siguiente diagrama muestra una vista conceptual de la migración.



Migración a SaaS

Este diagrama representa la experiencia objetivo para cualquier ruta de migración. Incluye los mismos servicios compartidos que se describieron anteriormente. En el centro se encontraría la aplicación.

La idea clave es que puede colocar cualquier número de modelos de aplicación como elementos centrales de este entorno. El primer paso de la migración podría consistir en que cada inquilino opere en su propio silo. O bien, puede tener una arquitectura híbrida en la que algunos elementos se distribuyen en silos mientras que otras funciones se abordan mediante un conjunto de microservicios modernizados.

La medida en que se modernice inicialmente la aplicación dependerá de la naturaleza del entorno heredado, las necesidades del mercado, las cuestiones de costos, etc. Lo que no cambia es la introducción de estos servicios compartidos.

Cualquier migración a SaaS debe admitir estos servicios compartidos fundamentales para que su empresa pueda operar en un modelo SaaS. Por ejemplo, todas las variaciones de la arquitectura de la aplicación necesitan una identidad en SaaS. Necesitará operaciones orientadas a los inquilinos para administrar y monitorizar su solución SaaS.

Implementar estos servicios compartidos al principio de la migración le permite ofrecer una experiencia de SaaS a sus clientes, incluso si la aplicación subyacente sigue ejecutándose en un silo de pila completa para cada inquilino.

El objetivo general es hacer que la aplicación se ejecute en un modelo SaaS. Después, puede centrar su atención en una mayor modernización y perfeccionamiento de su aplicación. Este enfoque también le permite transferir los demás ámbitos de su empresa (marketing, ventas, soporte, etc.) a un ritmo más rápido. Y lo que es más importante, le permite empezar a captar y recoger comentarios de los clientes, que se pueden utilizar para ir modernizando su entorno.

Es importante tener en cuenta que es posible que los servicios compartidos que implemente no incluyan todas las funciones o mecanismos que necesitará en última instancia. El objetivo principal es crear los mecanismos compartidos que se necesitan al principio de la migración. Esto le permite centrarse en los elementos del sistema que son esenciales para la evolución de la arquitectura de sus aplicaciones y de sus procesos operativos.

Identidad SaaS

SaaS añade nuevas cuestiones a tener en cuenta en el modelo de identidad de su aplicación. Cuando se autentica a cada usuario, este debe estar asociado a un contexto de inquilino específico. Este contexto proporciona información básica sobre el inquilino que se utiliza en todo su entorno SaaS.

Esta vinculación de los inquilinos con los usuarios suele denominarse identidad SaaS de su aplicación. A medida que cada usuario se autentica, el proveedor de identidades suele generar un token que incluye tanto la identidad del usuario como la del inquilino.

Asociar los inquilinos con los usuarios es un aspecto fundamental de la arquitectura SaaS que tiene muchas implicaciones posteriores. El token de este proceso de identidad se transfiere a los microservicios de la aplicación y se utiliza para crear registros que tengan en cuenta los inquilinos, elaborar métricas, medir la facturación, imponer el aislamiento de los inquilinos, etc.

Es fundamental evitar situaciones que dependan de mecanismos separados e independientes para asociar usuarios con inquilinos. Esto puede poner en riesgo la seguridad del sistema y, a menudo, crea cuellos de botella en la arquitectura.

Aislamiento de inquilinos

Cuanto más avance en la adopción de un modelo multiinquilino, más les preocupará a sus inquilinos la posibilidad de que algunos puedan acceder a los recursos de otros. Los sistemas SaaS incluyen mecanismos explícitos que garantizan que los recursos de cada inquilino, incluso si se ejecutan en una infraestructura compartida, estén aislados.

Esto es lo que denominamos aislamiento de inquilinos. La idea detrás del aislamiento de inquilinos es que su arquitectura de SaaS introduzca mecanismos que controlen estrictamente el acceso a los recursos y bloqueen cualquier intento de acceder a los recursos de otro inquilino.

Tenga en cuenta que el aislamiento de los inquilinos es independiente de los mecanismos de seguridad generales. Aunque su sistema incluya funciones de autenticación y autorización, el hecho de que un usuario inquilino esté autenticado no significa que su sistema cuente con aislamiento. El aislamiento se aplica aparte de las funciones de autenticación y autorización básicas que pueden formar parte de su aplicación.

Para entenderlo mejor, imagine que ha utilizado un proveedor de identidades para autenticar el acceso a su sistema SaaS. El token de esta experiencia de autenticación también puede incluir información sobre el rol de un usuario, que podría utilizarse para controlar el acceso de ese usuario a una función específica de la aplicación. Estos elementos proporcionan seguridad, pero no aislamiento. De hecho, un usuario podría estar autenticado y autorizado y, aun así, acceder a los recursos de otro inquilino. Las funciones de autenticación y autorización podrían no impedir este acceso.

El aislamiento de inquilinos se centra exclusivamente en utilizar el contexto del inquilino para limitar el acceso a los recursos. Evalúa el contexto del inquilino actual y lo utiliza para determinar a qué recursos podrá acceder. A continuación, aplica este aislamiento a todos los usuarios de ese inquilino.

Esto se hace más complejo a medida que observamos cómo se logra el aislamiento de inquilinos en los diferentes patrones de la arquitectura SaaS. En algunos casos, el aislamiento puede lograrse dedicando pilas completas de recursos a un inquilino al que se le apliquen políticas de red (u otras más generales) que impidan el acceso entre los inquilinos. En otros casos, puede tener recursos agrupados (elementos de una tabla de [Amazon DynamoDB](#)) que requieran políticas más granulares para controlar el acceso a los recursos.

Los intentos de acceder al recurso de un inquilino deben limitarse únicamente a aquellos recursos que pertenezcan a ese inquilino. El trabajo de los desarrolladores y arquitectos de SaaS es

determinar qué combinación de herramientas y tecnologías serán necesarias para cumplir con los requisitos de aislamiento de una aplicación específica.

Particiones de datos

Las particiones de datos se usan para describir diferentes estrategias empleadas para representar datos en un entorno multiinquilino. Este término se usa ampliamente para abarcar una variedad de enfoques y modelos diferentes que se pueden utilizar para asociar diferentes conjuntos de datos con inquilinos individuales.

Tenga en cuenta que a menudo se tiende a considerar las particiones de datos y el aislamiento de los inquilinos como intercambiables. Sin embargo, estos dos conceptos no son equivalentes. Cuando hablamos de particiones de datos, nos referimos a la forma en que se almacenan los datos de los inquilinos individuales. Las particiones de datos no garantizan que los datos estén aislados. El aislamiento debe aplicarse aparte para garantizar que un inquilino no pueda acceder a los recursos de otro.

Cada tecnología de almacenamiento de AWS aporta su propio conjunto de consideraciones a la estrategia de particiones de datos. Por ejemplo, el aislamiento de datos en Amazon DynamoDB tendrá un aspecto muy diferente al aislamiento de datos con Amazon [Relational Database Service \(Amazon RDS\)](#).

Por lo general, cuando se plantea la partición de datos, comienza por decidir si los datos se agruparán o estarán aislados. En un modelo de datos aislados, se tiene una estructura de almacenamiento distinta para cada inquilino, sin datos mezclados. En el caso de la partición agrupada, los datos se mezclan y se particionan en función de un identificador de arrendatario que determina qué datos están asociados a cada arrendatario.

Por ejemplo, con Amazon DynamoDB, un modelo de datos aislados utiliza una tabla independiente para cada inquilino. La agrupación de datos en Amazon DynamoDB se logra almacenando el identificador del inquilino en la clave de partición de cada tabla de Amazon DynamoDB que administra los datos de todos los inquilinos.

Imagínese cómo puede variar esto en la gama de servicios de AWS, donde cada uno de ellos presenta sus propias estructuras, lo que puede requerir un enfoque diferente para crear modelos de almacenamiento en silos y agrupados con cada servicio.

Si bien la partición de datos y el aislamiento de inquilinos son temas distintos, es probable que las estrategias de partición de datos que elija estén influenciadas por el modelo de aislamiento de los datos. Por ejemplo, puede distribuir parte del almacenamiento en silos porque este enfoque se adapte mejor a los requisitos de su ámbito de sus clientes. O bien, puede optar por silos porque es

posible que el modelo de grupos no le permita aplicar el aislamiento con el nivel de granularidad que requiere su solución.

Un vecino ruidoso también puede afectar a su planteamiento sobre el aislamiento. Es posible que algunas cargas de trabajo o casos de uso de su aplicación deban mantenerse separados para limitar los impactos de otros inquilinos o para cumplir con los acuerdos de nivel de servicio (SLA).

Medición, métricas y facturación

Al abordar el tema del SaaS, suelen mencionarse las nociones de medición, métricas y facturación. Estos conceptos a menudo se entremezclan en un solo concepto. Sin embargo, es importante distinguir las diferentes funciones que desempeñan la medición, las métricas y la facturación en un entorno SaaS.

El problema de estos conceptos es que a menudo la misma palabra puede usarse con diferentes significados. Por ejemplo, podemos hablar de la medición que se utiliza para generar una factura. Al mismo tiempo, también podemos hablar de la medición que se utiliza para rastrear el consumo interno de recursos y que no tiene ninguna relación con la facturación. También hablamos de métricas y SaaS en muchos contextos que pueden entremezclarse al abordar este tema.

Para ayudar a resolver este problema, vamos a asociar algunos conceptos específicos a cada uno de estos términos (teniendo en cuenta que no hay definiciones categóricas).

- **Medición:** este concepto, si bien tiene muchas definiciones, se adapta mejor al ámbito de la facturación de SaaS. La idea es medir la actividad de los inquilinos o el consumo de recursos con el fin de recoger los datos necesarios para generar una factura.
- **Métricas:** las métricas representan todos los datos que se capturan para analizar las tendencias en el ámbito de la empresa, las operaciones y la tecnología. Estos datos se utilizan en muchos contextos y funciones dentro del equipo de SaaS.

Aunque esta distinción no tiene una importancia fundamental, ayuda a entender mejor el papel de la medición y las métricas en un entorno SaaS.

Ahora, si ilustramos estos dos conceptos mediante ejemplos, podría pensar en equipar su aplicación con eventos de medición específicos que sirvan para detectar los datos necesarios para generar una factura. Podría ser el número de solicitudes o el número de usuarios activos, o bien podría asociarse a algún conjunto de datos de consumo (solicitudes, CPU, memoria) que se correlacione con alguna unidad que sea relevante para sus clientes.

En su entorno SaaS, publicará estos eventos de facturación desde su aplicación, y la estructura de facturación empleada por su sistema SaaS los incorporará y aplicará. Puede ser un sistema de facturación de terceros o uno personalizado.

A diferencia de esto, las métricas sirven para capturar las acciones, actividades, patrones de consumo, etc. que son esenciales para evaluar el buen estado del sistema y la huella operativa

de los distintos inquilinos. Las métricas que publica y agrega aquí vienen determinadas por las necesidades de diferentes personas (equipos operativos, propietarios de productos, arquitectos, etc.). Aquí, los datos de estas métricas se publican y se agregan a algunas herramientas de análisis que permiten a estos distintos usuarios crear vistas de la actividad del sistema que examinen aquellos aspectos que más se ajusten a las personas. Es posible que al propietario de un producto le interese saber el consumo que hacen de las funciones los diferentes inquilinos. Un arquitecto podría necesitar vistas que le ayuden a entender el consumo que hacen los inquilinos de los recursos de la infraestructura, etc.

SaaS B2B y B2C

Las ofertas de SaaS se orientan a los mercados B2B y B2C. Si bien las empresas y los clientes tienen dinámicas diferentes, los principios generales del SaaS no cambian en modo alguno estos mercados.

Si examinamos la incorporación, por ejemplo, los clientes de B2B y B2C pueden tener experiencias de incorporación diferentes. Es cierto que los sistemas B2C pueden centrarse más en un flujo de incorporación de modo autoservicio (aunque los sistemas B2B también pueden admitir este tipo de incorporación).

Aunque puede haber diferencias en la forma en que se ofrece la incorporación a los clientes, los valores fundamentales subyacentes son prácticamente los mismos. Incluso si una solución B2B se basa en un proceso de incorporación interno, este debería ser sea lo más sencillo y automatizado posible. Una solución B2B no implica que cambiemos nuestras expectativas sobre el tiempo de creación de valor para nuestros clientes.

Conclusión

El objetivo de este documento es describir los conceptos fundamentales de la arquitectura SaaS, proporcionando algunas aclaraciones sobre los modelos y la terminología que se emplea para definir los patrones y las estrategias de SaaS. Esperamos que esto ofrezca a las organizaciones una visión más clara del panorama general de SaaS.

Una gran parte de los temas que se abordan aquí se centran en lo que significa ser SaaS, poniendo especial énfasis en la creación de un entorno que permita administrar y controlar todos los inquilinos de SaaS mediante una experiencia unificada. Esto tiene relación con la idea central de que SaaS es, ante todo, un modelo de negocio. La arquitectura SaaS que cree debería promover estos objetivos empresariales fundamentales.

Documentación adicional

Hay una serie de recursos que ofrecen más detalles sobre los patrones de arquitectura SaaS que se ajustan a los descritos en este artículo.

Para obtener información adicional, consulte:

- [Estrategias de aislamiento de inquilinos de SaaS](#) (documento técnico de AWS)
- [Estrategias de almacenamiento de SaaS](#) (documento técnico de AWS)
- [Enfoque de SaaS Well-Architected](#) (documento técnico de AWS)

Colaboradores

Las siguientes personas y organizaciones han colaborado en este documento:

- Tod Golding, Principal Partner Solutions Architect, AWS SaaS Factory

Revisiones del documento

Para recibir notificaciones sobre las actualizaciones de este documento técnico, suscríbase a la fuente RSS.

Cambio	Descripción	Fecha
Publicación inicial	Documento técnico publicado.	3 de agosto de 2022

Avisos

Es responsabilidad de los clientes realizar su propia evaluación independiente de la información que contiene este documento. El presente documento: (a) tiene solo fines informativos, (b) representa las ofertas y prácticas actuales de los productos de AWS, que están sujetas a cambios sin previo aviso, y (c) no supone ningún compromiso ni garantía por parte de AWS y sus filiales, proveedores o licenciantes. Los productos o servicios de AWS se proporcionan “tal cual”, sin garantías, afirmaciones ni condiciones de ningún tipo, ya sean expresas o implícitas. Las responsabilidades y obligaciones de AWS con respecto a sus clientes se controlan mediante los acuerdos de AWS y este documento no forma parte ni modifica ningún acuerdo entre AWS y sus clientes.

© 2023 Amazon Web Services, Inc. o sus filiales. Todos los derechos reservados.

Glosario de AWS

Para ver la terminología más reciente de AWS, consulte el [Glosario de AWS](#) en la Referencia de Glosario de AWS.