

Whitepaper do AWS

# Fundamentos da arquitetura SaaS



# Fundamentos da arquitetura SaaS: Whitepaper do AWS

Copyright © 2023 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

As marcas comerciais e imagens comerciais da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre os clientes ou que deprecie ou desprestigie a Amazon. Todas as outras marcas comerciais que não pertencem à Amazon pertencem a seus respectivos proprietários, que podem ou não ser afiliados, conectados ou patrocinados pela Amazon.

---

# Table of Contents

Resumo e introdução .....	i
Introdução .....	1
Você é Well-Architected? .....	2
SaaS é um modelo de negócios .....	3
Você é um serviço, não um produto .....	5
A motivação inicial .....	6
Mudando para uma experiência unificada .....	9
Ambiente de gerenciamento versus plano de aplicação .....	12
Serviços centrais .....	14
Redefinindo a multilocalização .....	15
O caso extremo .....	17
Removendo o termo inquilino único .....	19
Apresentando o silo e a reserva .....	19
Silo e reserva de pilha completa .....	21
SaaS versus Provedor de Serviços Gerenciado (MSP) .....	23
Migração SaaS .....	25
Identidade SaaS .....	29
Isolamento de inquilinos .....	30
Particionamento de dados .....	32
Medição, métricas e cobrança .....	34
SaaS B2B e B2C .....	36
Conclusão .....	37
Outras fontes de leitura .....	38
Colaboradores .....	39
Revisões do documento .....	40
Avisos .....	41
Glossário do AWS .....	42

# Fundamentos da arquitetura SaaS

Data de publicação: 3 de agosto de 2022 ([Revisões do documento](#))

O escopo, as metas e a natureza da administração de uma empresa em um modelo de software como serviço (SaaS) podem ser difíceis de definir. A terminologia e os padrões usados para caracterizar os SaaS variam de acordo com sua origem. O objetivo deste documento é definir melhor os elementos fundamentais do SaaS e criar uma imagem mais clara dos padrões, termos e sistemas de valores que são aplicados ao projetar e fornecer um sistema SaaS no AWS. O objetivo mais amplo é fornecer uma coleção de insights fundamentais que forneçam aos clientes uma visão mais clara das opções que eles devem considerar ao adotar um modelo de entrega SaaS.

Este artigo é destinado a construtores e arquitetos de SaaS que estão no início de sua jornada de SaaS, bem como a construtores mais experientes que desejam refinar sua compreensão dos principais conceitos de SaaS. Algumas dessas informações também podem ser úteis para proprietários e estrategistas de produtos SaaS que desejam se familiarizar mais com o cenário de SaaS.

## Introdução

O termo software como serviço (SaaS) é usado para descrever um modelo de negócios e entrega. O desafio, no entanto, é que o que significa ser SaaS não é universalmente compreendido.

Embora haja algum acordo sobre alguns dos principais pilares do SaaS, ainda há alguma confusão sobre o que significa ser SaaS. É natural ter alguma variação na forma como as equipes podem ver o SaaS. Ao mesmo tempo, a falta de clareza sobre os conceitos e termos de SaaS pode criar alguma confusão para aqueles que exploram um modelo de entrega de SaaS.

Este documento se concentra em descrever a terminologia usada para descrever os principais conceitos de SaaS. Ter uma mentalidade compartilhada em torno desses conceitos cria uma imagem clara dos elementos fundamentais de uma arquitetura SaaS, equipando você com um vocabulário compartilhado para descrever estruturas de arquitetura SaaS. Isso é especialmente útil quando você se aprofunda em conteúdo adicional baseado nesses temas.

Este whitepaper se afasta dos detalhes da arquitetura da multilocação e explora como definimos os fundamentos do que significa ser SaaS. Idealmente, isso também fornecerá um conjunto mais claro de terminologia que permita que as organizações se alinhem mais rapidamente ao sabor e à natureza de suas soluções SaaS.

## Você é Well-Architected?

O [AWS Well-Architected Framework](#) ajuda você a entender os prós e os contras das decisões que você toma ao criar sistemas na nuvem. Os seis pilares do framework permitem a você conhecer as melhores práticas de arquitetura para criar e operar sistemas confiáveis, seguros, econômicos e sustentáveis na nuvem. Usando o [AWS Well-Architected Tool](#), disponível gratuitamente no [AWS Management Console](#), você pode analisar suas cargas de trabalho em relação a essas melhores práticas respondendo a um conjunto de perguntas para cada pilar.

No [SaaS Lens](#), nos concentramos nas melhores práticas para arquitetar suas cargas de trabalho de software como serviço (SaaS) no AWS.

Para obter orientações especializadas e melhores práticas adicionais para a arquitetura de sua nuvem (implantações de arquitetura de referência, diagramas e whitepapers), consulte o [AWS Architecture Center](#).

# SaaS é um modelo de negócios

Definir o que significa ser SaaS começa com a concordância com um princípio fundamental: SaaS é um modelo de negócios. Isso significa que, acima de tudo, a adoção de um modelo de entrega SaaS é diretamente orientada por um conjunto de objetivos de negócios. Sim, a tecnologia será usada para atingir algumas dessas metas, mas o SaaS consiste em implementar uma mentalidade e um modelo que visem um conjunto específico de objetivos de negócios.

Vamos examinar mais de perto alguns dos principais objetivos de negócios associados à adoção de um modelo de entrega SaaS.

- **Agilidade** — Esse termo resume o objetivo mais amplo do SaaS. As empresas SaaS bem-sucedidas são construídas com a ideia de que devem estar prontas para se adaptar continuamente ao mercado, aos clientes e à dinâmica competitiva. As grandes empresas de SaaS são estruturadas para adotar continuamente novos modelos de preços, novos segmentos de mercado e novas necessidades dos clientes.
- **Eficiência operacional** — as empresas de SaaS confiam na eficiência operacional para promover escala e agilidade. Isso significa implementar uma cultura e ferramentas focadas na criação de uma pegada operacional que promova o lançamento frequente e rápido de novos recursos. Isso também significa ter uma experiência única e unificada que permite gerenciar, operar e implantar todos os ambientes do cliente coletivamente. A ideia de oferecer suporte a versões e personalizações únicas acabou. As empresas de SaaS valorizam a eficiência operacional como um pilar central de sua capacidade de crescer e escalar os negócios com sucesso.
- **Integração sem atrito** — Como parte de ser mais ágil e abraçar o crescimento, você também deve priorizar a diminuição de qualquer atrito no processo de integração do cliente inquilino. Isso se aplica universalmente aos clientes business to business (B2B) e business-to-customer (B2C). Não importa qual segmento ou tipo de cliente você atende, você ainda precisa se concentrar no tempo de valorização para seus clientes. A mudança para um modelo centrado em serviços exige que o negócio de SaaS se concentre em todos os aspectos da experiência do cliente, com ênfase específica na repetibilidade e eficiência do ciclo de vida geral de integração.
- **Inovação** — migrar para o SaaS não significa apenas atender às necessidades dos clientes atuais; trata-se de implementar os elementos fundamentais que permitem inovar. Você quer reagir e responder às necessidades do cliente em seu modelo SaaS. No entanto, você também quer usar essa agilidade para impulsionar inovações futuras que permitam desbloquear novos mercados, oportunidades e eficiências para seus clientes.

- Resposta do mercado — O SaaS se afasta da noção tradicional de lançamentos trimestrais e planos bienais. Ele se baseia em sua agilidade para dar à organização a capacidade de reagir e responder à dinâmica do mercado quase em tempo real. O investimento nos elementos organizacionais, técnicos e culturais do SaaS cria a oportunidade de dinamizar a estratégia de negócios com base na dinâmica emergente do cliente e do mercado.
- Crescimento — O SaaS é uma estratégia de negócios centrada no crescimento. O alinhamento de todas as partes móveis da organização em torno da agilidade e eficiência dá às organizações de SaaS a capacidade de atingir um modelo de crescimento. Isso significa implementar os mecanismos que abrangem e acolhem a rápida adoção de sua oferta de SaaS.

Você notará que cada um desses itens está focado em um resultado comercial. Há uma grande variedade de estratégias e padrões técnicos que podem ser usados para criar um sistema SaaS. No entanto, nada sobre essas estratégias técnicas muda a história mais ampla dos negócios.

Quando conversamos com as organizações e perguntamos o que elas estão tentando alcançar como parte da adoção do SaaS, sempre começamos com essa discussão focada nos negócios. As escolhas tecnológicas são importantes, mas devem ser realizadas no contexto dessas metas de negócios. Ser multi-inquilino sem obter agilidade, eficiência operacional ou integração sem atrito, por exemplo, prejudicaria o sucesso de seus negócios de SaaS.

Com isso como pano de fundo, vamos tentar formalizar isso em uma definição mais concisa de SaaS que esteja em conformidade com os princípios descritos anteriormente:

O SaaS é um modelo de negócios e entrega de software que oferece às organizações a capacidade de oferecer suas soluções em um modelo centrado em serviços e de baixo atrito que maximiza o valor para clientes e fornecedores. Ela se baseia na agilidade e na eficiência operacional como pilares de uma estratégia de negócios que promove crescimento, alcance e inovação.

Você deve ver o alinhamento entre os objetivos de negócios e como eles dependem de uma experiência compartilhada para todos os clientes. Uma grande parte da mudança para o SaaS significa se afastar das personalizações únicas que podem fazer parte de um modelo de software tradicional. Qualquer esforço para oferecer especialização aos clientes geralmente nos afasta dos valores fundamentais que estamos tentando alcançar com o SaaS.

# Você é um serviço, não um produto

Adotar um modelo de “serviço” é mais do que apenas marketing ou terminologia. Em uma mentalidade de serviço, você se afastará de aspectos da abordagem tradicional de desenvolvimento baseada em produtos. Embora os recursos e a funcionalidade sejam certamente importantes para cada produto, o SaaS dá mais ênfase à experiência que os clientes terão com seu serviço.

O que isso significa? Em um modelo centrado em serviços, você pensa mais sobre como os clientes são integrados ao seu serviço, com que rapidez eles obtêm valor e com que rapidez você pode introduzir recursos que atendam às necessidades do cliente. Os detalhes associados à forma como seu serviço é criado, operado e gerenciado estão fora da visão do cliente.

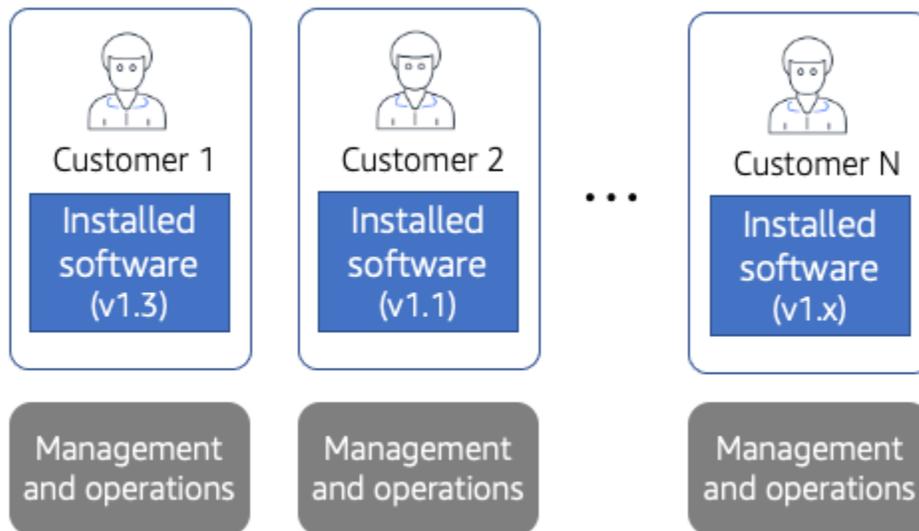
Nesse modo, pensamos nesse serviço SaaS como faríamos com qualquer outro serviço que possamos consumir. Se estamos em um restaurante, certamente nos preocupamos com a comida, mas também com o serviço. A rapidez com que o garçom chega à sua mesa, com que frequência ele reabastece a água, a rapidez com que a comida chega — todas essas são medidas da experiência do serviço. Essa é a mesma mentalidade e sistema de valores que devem moldar a forma como pensamos sobre a criação de um serviço SaaS.

Esse modelo como serviço deve ter uma forte influência na forma como você constrói suas equipes e seu serviço. Seu acúmulo de trabalho agora colocará esses atributos de experiência em pé de igualdade ou mais alto do que os recursos e funções. A empresa também os considerará fundamentais para o crescimento e o sucesso de longo prazo de sua oferta de SaaS.

## A motivação inicial

Para entender o SaaS, vamos começar com uma noção relativamente simples do que estamos tentando alcançar ao criar um negócio de SaaS. O melhor lugar para começar é observar como o software tradicional (não SaaS) foi criado, gerenciado e operado.

O diagrama a seguir fornece uma visão conceitual de como vários fornecedores empacotaram e entregaram suas soluções.



### O modelo clássico para empacotar e fornecer soluções de software

Neste diagrama, descrevemos uma coleção de ambientes de clientes. Esses clientes representam as diferentes empresas ou entidades que compraram o software de um fornecedor. Cada um desses clientes está basicamente trabalhando em um ambiente autônomo em que instalou o produto de um fornecedor de software.

Nesse modo, a instalação de cada cliente é tratada como um ambiente independente dedicado a esse cliente. Isso significa que os clientes se veem como proprietários desses ambientes, potencialmente solicitando uma personalização única ou configurações exclusivas que atendam às suas necessidades.

Um efeito colateral comum dessa mentalidade é que os clientes controlarão qual versão do produto estão executando. Isso pode acontecer por vários motivos. Os clientes podem estar apreensivos com os novos recursos ou preocupados com as interrupções associadas à adoção de uma nova versão.

Você pode imaginar como essa dinâmica pode impactar a pegada operacional do fornecedor de software. Quanto mais você permite que os clientes tenham ambientes únicos, mais difícil se torna gerenciar, atualizar e oferecer suporte às diferentes configurações de cada cliente.

Essa necessidade de ambientes únicos geralmente exige que as organizações criem equipes dedicadas que forneçam experiência separada de gerenciamento e operações para cada cliente. Embora alguns desses recursos possam ser compartilhados entre clientes, esse modelo geralmente introduz despesas incrementais para cada novo cliente incorporado.

Fazer com que cada cliente execute suas soluções em seu próprio ambiente (na nuvem ou on-premises) também afeta o custo. Embora você possa tentar escalar esses ambientes, o escalonamento será limitado à atividade de um único cliente. Essencialmente, sua otimização de custos é limitada ao que você pode alcançar dentro do escopo de um ambiente individual do cliente. Isso também significa que você pode precisar de estratégias de escalonamento separadas para acomodar as variações de atividade entre os clientes.

Inicialmente, algumas empresas de software verão esse modelo como uma estrutura poderosa. Eles usam a capacidade de fornecer personalização única como ferramenta de vendas, permitindo que novos clientes imponham requisitos exclusivos de seu ambiente. Embora a contagem de clientes e o crescimento do negócio permaneçam modestos, esse modelo parece perfeitamente sustentável.

No entanto, à medida que as empresas começam a ter um sucesso mais amplo, as restrições desse modelo começam a criar desafios reais. Imagine, por exemplo, um cenário em que sua empresa atinge um pico significativo de crescimento que faz com que você adicione muitos novos clientes em um ritmo acelerado. Esse crescimento começará a aumentar a sobrecarga operacional, a complexidade do gerenciamento, o custo e uma série de outros problemas.

Em última análise, a sobrecarga e o impacto coletivos desse modelo podem começar a minar fundamentalmente o sucesso de uma empresa de software. O primeiro ponto problemático pode ser a eficiência operacional. O aumento de pessoal e os custos associados à contratação de clientes começam a corroer as margens dos negócios.

No entanto, os problemas operacionais são apenas parte do desafio. O verdadeiro problema é que esse modelo, à medida que se expande, começa a impactar diretamente a capacidade da empresa de lançar novos recursos e acompanhar o ritmo do mercado. Quando cada cliente tem seu próprio ambiente, os fornecedores devem equilibrar uma série de requisitos de atualização, migração e clientes à medida que tentam introduzir novos recursos em seu sistema.

Isso normalmente leva a ciclos de lançamento mais longos e complexos, o que tende a reduzir o número de lançamentos feitos a cada ano. Mais importante ainda, essa complexidade faz com que

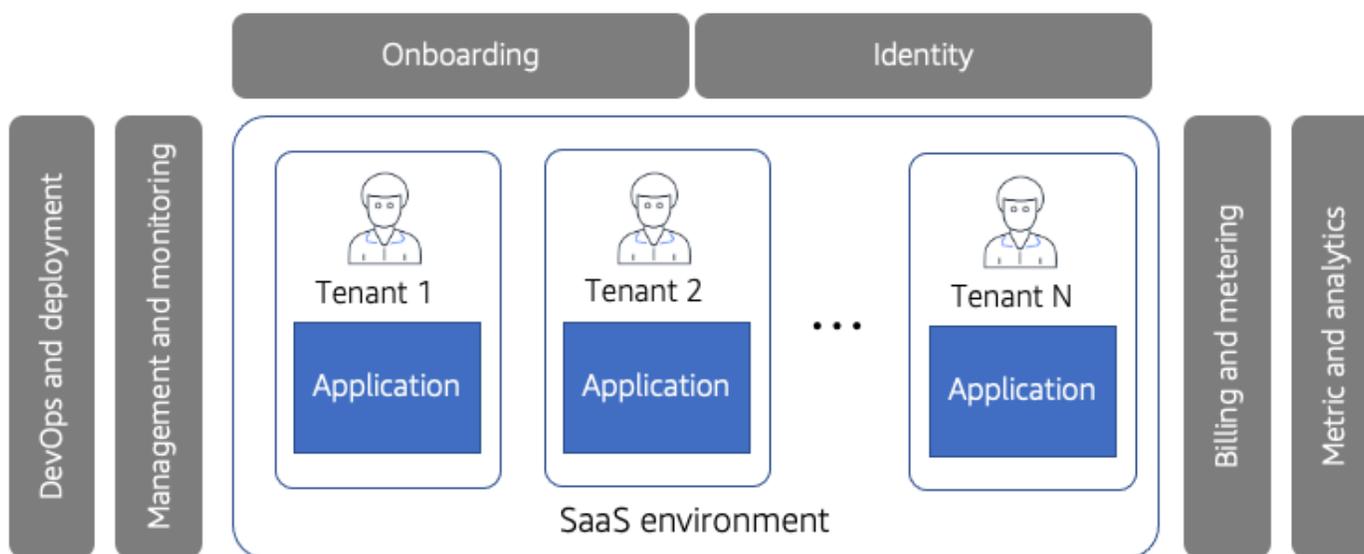
as equipes dediquem mais tempo à análise de cada novo atributo muito antes de ser lançado para o cliente. As equipes começam a se concentrar mais na validação de novos recursos e menos na velocidade da entrega. A sobrecarga de lançar novos recursos se torna tão significativa que as equipes ficam mais focadas na mecânica dos testes e menos na nova funcionalidade que impulsiona a inovação de suas ofertas.

Nesse modo mais lento e cauteloso, as equipes tendem a ter longos tempos de ciclo que colocam lacunas cada vez maiores entre o início de uma ideia e o momento em que ela chega às mãos de um cliente. No geral, isso pode prejudicar sua capacidade de reagir à dinâmica do mercado e às pressões competitivas.

## Mudando para uma experiência unificada

Para atender às necessidades desse dilema clássico do software, as organizações recorrem a um modelo que permite criar uma experiência única e unificada que permite que os clientes sejam gerenciados e operados coletivamente.

O diagrama a seguir fornece uma visão conceitual de um ambiente em que todos os clientes são gerenciados, integrados, faturados e operados por meio de um modelo compartilhado.



Uma visão conceitual de um ambiente em que todos os clientes são gerenciados, integrados, faturados e operados por meio de um modelo compartilhado

À primeira vista, isso pode não parecer muito diferente do modelo anterior. No entanto, à medida que nos aprofundarmos um pouco mais, você verá que existem diferenças fundamentais e significativas nessas duas abordagens.

Primeiro, você notará que os ambientes do cliente foram renomeados para inquilinos. Essa noção de inquilino é fundamental para o SaaS. A ideia básica é que você tenha um único ambiente SaaS e cada um de seus clientes seja visto como inquilino desse ambiente, consumindo os recursos de que precisam. Um inquilino pode ser uma empresa com muitos usuários ou pode se correlacionar diretamente com um usuário individual.

Para entender melhor a ideia de um inquilino, considere a ideia de apartamentos ou prédios comerciais. O espaço em cada um desses edifícios é alugado para inquilinos individuais. Os

inquilinos dependem de alguns dos recursos compartilhados do prédio (água, energia e assim por diante), pagando pelo que consomem.

Os inquilinos de SaaS seguem um padrão semelhante. Você tem a infraestrutura do seu ambiente SaaS e inquilinos que consomem a infraestrutura desse ambiente. A quantidade de recursos consumidos por cada inquilino pode variar. Esses inquilinos também são gerenciados, cobrados e operados coletivamente.

Se você voltar ao diagrama, verá a noção de locação ganhando vida. Aqui, os inquilinos não têm mais seu próprio ambiente. Em vez disso, todos os inquilinos são alojados e gerenciados dentro das paredes de um ambiente SaaS coletivo.

O diagrama também inclui uma variedade de serviços compartilhados que cercam seu ambiente SaaS. Esses serviços são globais para todos os inquilinos do seu ambiente SaaS. Isso significa que a integração e a identidade, por exemplo, são compartilhadas por todos os inquilinos desse ambiente. O mesmo vale para gerenciamento, operações, implantação, cobrança e métricas.

Essa ideia de um conjunto unificado de serviços que são aplicados universalmente a todos os seus inquilinos é fundamental para o SaaS. Ao compartilhar esses conceitos, você pode enfrentar vários desafios associados ao modelo clássico descrito acima.

Outro elemento importante e um tanto sutil desse diagrama é que todos os inquilinos desse ambiente estão executando a mesma versão do seu aplicativo. Já se foi a ideia de ter versões separadas e únicas em execução para cada cliente. Ter todos os inquilinos executando a mesma versão representa um dos atributos distintivos fundamentais de um ambiente SaaS.

Ao ter todos os clientes executando a mesma versão do seu produto, você não enfrenta mais muitos dos desafios de um modelo clássico de software instalado. No modelo unificado, novos recursos podem ser implantados em todos os inquilinos por meio de um processo único e compartilhado.

Essa abordagem oferece a capacidade de empregar um único painel operacional que pode gerenciar e operar todos os inquilinos. Isso permite gerenciar e monitorar seus inquilinos por meio de uma experiência operacional comum, permitindo que novos inquilinos sejam adicionados sem adicionar sobrecarga operacional incremental. Essa é uma parte essencial da proposta de valor do SaaS, que dá às equipes a capacidade de reduzir despesas operacionais e melhorar a agilidade organizacional geral.

Imagine o que significaria adicionar 100 ou 1.000 novos clientes nesse modelo. Em vez de se preocupar com a forma como esses novos clientes podem reduzir as margens e aumentar a complexidade, você pode ver esse crescimento como a oportunidade que ele representa.

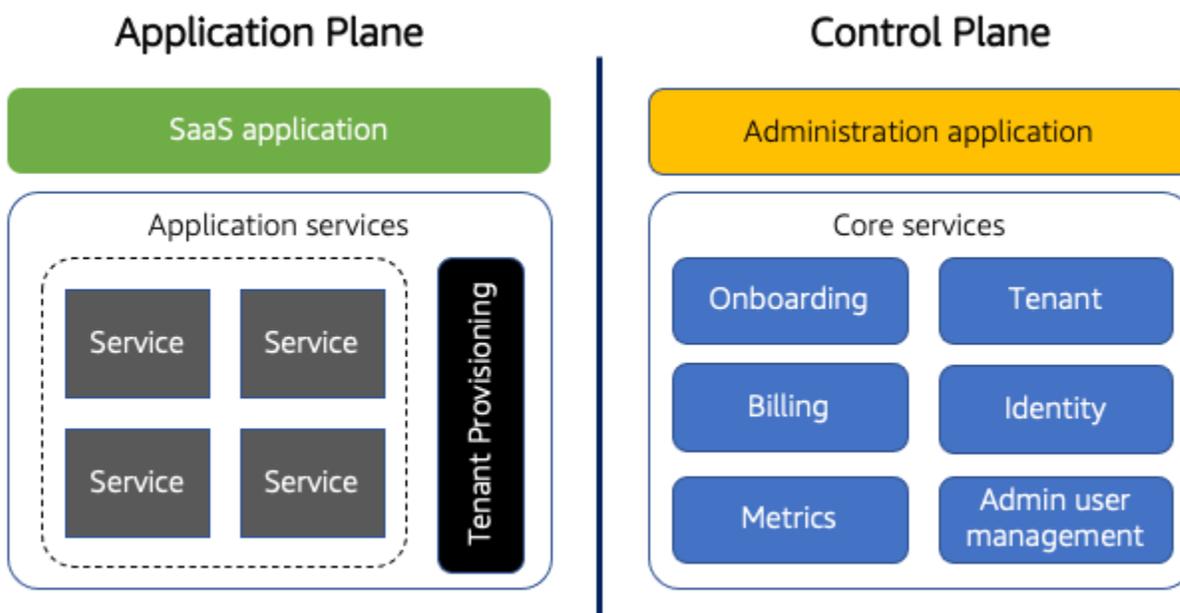
Geralmente, o foco do SaaS é colocado em como o aplicativo no meio desse modelo é implementado. As empresas querem se concentrar em como os dados são armazenados, como os recursos são compartilhados e assim por diante. No entanto, a realidade é que, embora esses detalhes sejam definitivamente importantes, há muitas maneiras pelas quais seu aplicativo pode ser construído e ainda se apresentar como uma solução SaaS para seus clientes.

O que é fundamental é a meta mais ampla de ter uma experiência única e unificada que envolva seus ambientes de inquilinos. Ter essa experiência compartilhada é o que permite impulsionar o crescimento, a agilidade e a eficiência operacional que estão conectados aos objetivos gerais de uma empresa SaaS.

## Ambiente de gerenciamento versus plano de aplicação

O diagrama anterior fornece uma visão conceitual dos principais conceitos da arquitetura SaaS. Agora, vamos nos aprofundar nisso e definir melhor como seu ambiente SaaS se decompõe em camadas distintas. Ter essa imagem mais clara dos limites entre os conceitos de SaaS facilitará a descrição das partes móveis de uma solução SaaS.

O diagrama a seguir divide seu ambiente SaaS em dois planos distintos. Do lado direito, fica o ambiente de gerenciamento. Esse lado do diagrama inclui todas as funcionalidades e serviços usados para integrar, autenticar, gerenciar, operar e analisar um ambiente multi-inquilino.



### Ambiente de gerenciamento versus plano de aplicação

Esse ambiente de gerenciamento é fundamental para qualquer modelo SaaS multi-inquilino. Toda solução SaaS, independentemente do esquema de implantação e isolamento do aplicativo, deve incluir os serviços que oferecem a capacidade de gerenciar e operar seus inquilinos por meio de uma experiência única e unificada.

Dentro do ambiente de gerenciamento, dividimos ainda mais isso em dois elementos distintos. Os principais serviços aqui representam o conjunto de serviços usados para orquestrar sua experiência multilocatária. Incluímos alguns exemplos comuns de serviços que normalmente fazem parte do núcleo, reconhecendo que os serviços principais podem variar um pouco para cada solução SaaS.

Você também notará que mostramos um aplicativo de administração separado. Isso representa o aplicativo (um aplicativo web, uma interface de linha de comando ou uma API) que pode ser usado por um provedor de SaaS para gerenciar seu ambiente multi-inquilino.

Uma ressalva importante é que o ambiente de gerenciamento e seus serviços não são, na verdade, multi-inquilinos. A funcionalidade não está fornecendo os atributos funcionais reais do seu aplicativo SaaS (que precisa ser multi-inquilino). Se você examinar qualquer um dos serviços principais, por exemplo, não encontrará o isolamento do inquilino e as outras estruturas que fazem parte da funcionalidade do seu aplicativo multi-inquilino. Esses serviços são globais para todos os inquilinos.

O lado esquerdo do diagrama faz referência ao plano de aplicação de um ambiente SaaS. É aqui que reside a funcionalidade multilocatária do seu aplicativo. O que aparece no diagrama precisa permanecer um pouco vago, porque cada solução pode ser implantada e decomposta de forma diferente com base nas necessidades de seu domínio, na área de cobertura de sua tecnologia e assim por diante.

O domínio do aplicativo é separado em dois elementos. Existe o aplicativo SaaS que representa a experiência/aplicativo do inquilino para sua solução. Essa é a superfície que os inquilinos tocam para interagir com seu aplicativo SaaS. Depois, há os serviços internos que representam a lógica de negócios e os elementos funcionais de uma solução SaaS. Eles podem ser microsserviços ou algum outro pacote de seus serviços de aplicativos.

Você também notará que apresentamos o provisionamento. Isso é feito para destacar o fato de que qualquer provisionamento de recursos para inquilinos durante a integração faria parte desse domínio do aplicativo. Alguns poderiam argumentar que isso pertence ao ambiente de gerenciamento. No entanto, nós o colocamos no domínio do aplicativo, porque os recursos que ele deve provisionar e configurar estão mais diretamente conectados aos serviços criados e configurados no plano do aplicativo.

Dividir isso em planos distintos torna mais fácil pensar no cenário geral de uma arquitetura SaaS. Mais importante ainda, ele destaca a necessidade de um conjunto de serviços que estão totalmente fora do escopo da funcionalidade do seu aplicativo.

## Serviços centrais

O ambiente de gerenciamento mencionado anteriormente menciona uma série de serviços principais que representam os serviços típicos usados para integrar, gerenciar e operar um ambiente SaaS. Pode ser útil destacar ainda mais o papel de alguns desses serviços para destacar seu escopo e propósito em um ambiente SaaS. Veja a seguir um breve resumo de cada um desses serviços:

- **Integração** — Toda solução SaaS deve fornecer um mecanismo fácil para introduzir novos inquilinos em seu ambiente SaaS. Isso pode ser uma página de inscrição de autoatendimento ou uma experiência gerenciada internamente. De qualquer forma, uma solução SaaS deve fazer todo o possível para remover o atrito interno e externo dessa experiência e garantir estabilidade, eficiência e repetibilidade desse processo. Ele desempenha um papel essencial no apoio ao crescimento e à escala de um negócio SaaS. Geralmente, esse serviço orquestra outros serviços para criar usuários, inquilinos, políticas de isolamento, provisionamento e recursos por inquilino.
- **Inquilino** — O serviço de inquilino fornece uma maneira de centralizar as políticas, os atributos e o estado dos inquilinos. A chave é que os inquilinos não são usuários individuais. Na verdade, um inquilino provavelmente está associado a muitos usuários.
- **Identidade** — Os sistemas SaaS precisam de uma maneira clara de conectar os usuários aos inquilinos que traga o contexto do inquilino à experiência de autenticação e autorização de suas soluções. Isso influencia tanto a experiência de integração quanto o gerenciamento geral dos perfis de usuário.
- **Faturamento** — Como parte da adoção do SaaS, as organizações geralmente adotam novos modelos de cobrança. Eles também podem explorar a integração com provedores de cobrança terceirizados. Esse serviço principal é amplamente focado em apoiar a integração de novos inquilinos e coletar dados de consumo e atividades que são usados para gerar faturas para os inquilinos.
- **Métricas** — As equipes de SaaS dependem muito de sua capacidade de capturar e analisar dados métricos ricos que trazem mais visibilidade sobre como os inquilinos usam seu sistema, como consomem recursos e como seus inquilinos interagem com seus sistemas. Esses dados são usados para moldar estratégias operacionais, de produtos e de negócios.
- **Gerenciamento de usuários administrativos** — os sistemas SaaS devem oferecer suporte tanto aos usuários inquilinos quanto aos usuários administradores. Os usuários administradores representam os administradores de um provedor de SaaS. Eles entrarão em sua experiência operacional para monitorar e gerenciar seu ambiente SaaS.

# Redefinindo a multilocação

Os termos multilocação e SaaS geralmente estão intimamente conectados. Em alguns casos, as organizações descrevem SaaS e multilocação como a mesma coisa. Embora isso possa parecer natural, equiparar SaaS e multilocação tende a levar as equipes a adotar uma visão puramente técnica do SaaS quando, na realidade, o SaaS é mais um modelo de negócios do que uma estratégia de arquitetura.

Para entender melhor esse conceito, vamos começar com a visão clássica da multilocação. Nessa visão puramente focada na infraestrutura, a multilocação é usada para descrever como os recursos são compartilhados pelos inquilinos para promover agilidade e eficiência de custos.

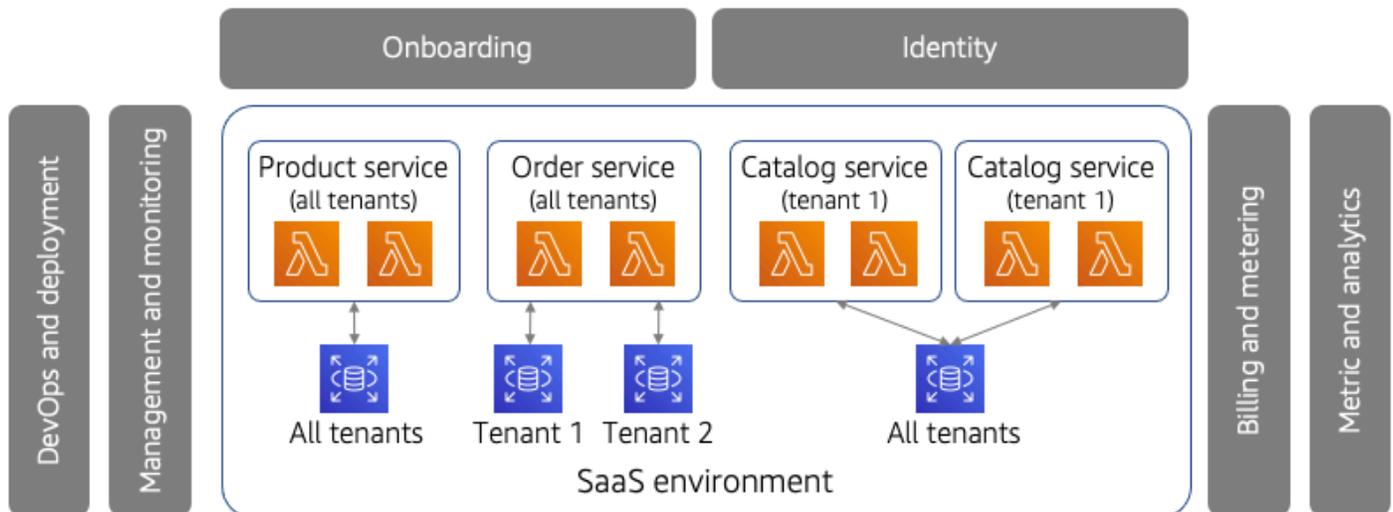
Suponha, por exemplo, que você tenha um microsserviço ou uma instância do [Amazon Elastic Compute Cloud](#) (Amazon EC2) que seja consumida por vários inquilinos do seu sistema SaaS. Esse serviço seria visto como executado em um modelo multi-inquilino, porque os inquilinos estão compartilhando o uso da infraestrutura que executa esse serviço.

O desafio dessa definição é que ela também vincula diretamente a noção técnica de multilocação ao SaaS. Ele presume que a característica definidora do SaaS é que ele deve ter uma infraestrutura compartilhada e multilocatária. Essa visão do SaaS começa a se desfazer à medida que observamos as várias maneiras pelas quais o SaaS é realizado em diferentes ambientes.

O diagrama a seguir fornece uma visão de um sistema SaaS que expõe alguns dos desafios que temos com a definição de multilocação.

Aqui você verá o modelo SaaS clássico descrito anteriormente, com uma série de serviços de aplicativos cercados por serviços compartilhados que permitem gerenciar e operar seus inquilinos coletivamente.

A novidade são os microsserviços que incluímos. O diagrama inclui três exemplos de microsserviços: produto, pedido e catálogo. Se você observar atentamente o modelo de locação de cada um desses serviços, notará que todos eles empregam padrões de locação ligeiramente diferentes.



## SaaS e multilocação

O serviço do produto compartilha todos os seus recursos (computação e armazenamento) com todos os inquilinos. Isso se alinha à definição clássica de multilocação. No entanto, se você observar o serviço de pedidos, verá que ele tem computação compartilhada, mas armazenamento separado para cada inquilino.

O serviço de catálogo adiciona outra variação em que a computação é separada para cada inquilino (uma implantação de microsserviço separada para cada inquilino), mas compartilha o armazenamento para todos os inquilinos.

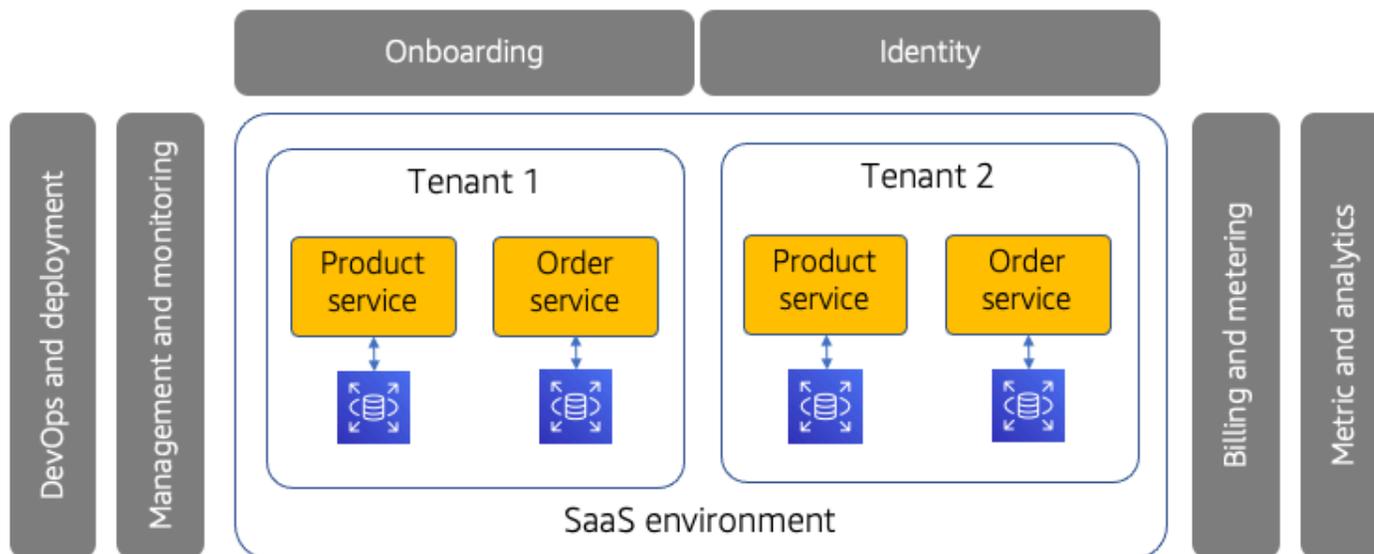
Variações dessa natureza são comuns em ambientes SaaS. [Vizinho barulhento](#), modelos hierárquicos, necessidades de isolamento — esses são alguns dos motivos pelos quais você pode compartilhar ou isolar seletivamente partes de sua solução SaaS.

Dadas essas variações — e muitas outras possibilidades — fica mais difícil descobrir como você deve usar o termo multi-inquilino para caracterizar esse ambiente. No geral, no que diz respeito ao cliente, esse é um ambiente multi-inquilino. No entanto, se usarmos a definição mais técnica, algumas partes desse ambiente são multilocatárias e outras não.

É por isso que se torna necessário deixar de usar o termo multi-inquilino para caracterizar ambientes SaaS. Em vez disso, podemos falar sobre como a multilocação é implementada em seu aplicativo, mas evite usá-la para caracterizar uma solução como SaaS. Se o termo multi-inquilino for usado, faz mais sentido usá-lo para descrever todo o ambiente SaaS como sendo multi-inquilino, sabendo que algumas partes da arquitetura podem ser compartilhadas e outras não. No geral, você ainda está operando e gerenciando esse ambiente em um modelo multi-inquilino.

## O caso extremo

Para destacar melhor essa noção de locação, vamos dar uma olhada em um modelo SaaS que tem inquilinos que compartilham zero recursos. O diagrama a seguir fornece um exemplo de ambiente SaaS que é empregado por alguns provedores de SaaS.



### Pilha por inquilino

Neste diagrama, você verá que ainda temos nosso ambiente comum em torno desses inquilinos. No entanto, cada inquilino é implantado com uma coleção dedicada de recursos. Nada é compartilhado pelos inquilinos neste modelo.

Este exemplo desafia o que significa ser multi-inquilino. Esse é um ambiente multi-inquilino, mesmo que nenhum dos recursos seja compartilhado? Os inquilinos que consomem esse sistema têm as mesmas expectativas que você teria de um ambiente SaaS com recursos compartilhados. Na verdade, eles podem não ter consciência de como seus recursos são implantados sob o capô do ambiente SaaS.

Embora esses inquilinos estejam operando em uma infraestrutura isolada, eles ainda são gerenciados e operados coletivamente. Eles compartilham uma experiência unificada de integração, identidade, métricas, faturamento e operação. Além disso, quando uma nova versão é lançada, ela é implantada em todos os inquilinos. Para que isso funcione, você não pode permitir uma personalização única para inquilinos individuais.

Esse exemplo extremo fornece um bom modelo para testar a noção de SaaS multi-inquilino. Embora possa não obter todas as eficiências da infraestrutura compartilhada, é um ambiente SaaS multi-inquilino totalmente válido. Para alguns clientes, seu domínio pode determinar que alguns ou todos os clientes usem esse modelo. Isso não significa que eles não sejam SaaS. Se eles usam esses serviços compartilhados e todos os inquilinos executam a mesma versão, isso ainda está em conformidade com os princípios fundamentais do SaaS.

Dados esses parâmetros e essa definição mais ampla de SaaS, você pode ver a necessidade de evoluir o uso do termo multi-inquilino. Faz mais sentido se referir a qualquer sistema SaaS gerenciado e operado coletivamente como sendo multi-inquilino. Em seguida, você pode usar uma terminologia mais granular para descrever como os recursos são compartilhados ou dedicados na implementação de uma solução SaaS.

## Removendo o termo inquilino único

Como parte do uso do termo multi-inquilino, é natural que as pessoas queiram usar o termo inquilino único para descrever ambientes SaaS. No entanto, considerando o pano de fundo descrito anteriormente, o termo inquilino único cria confusão.

O diagrama anterior é um ambiente de inquilino único? Embora cada inquilino tenha tecnicamente sua própria pilha, esses inquilinos ainda estão sendo operados e gerenciados em um modelo multi-inquilino. É por isso que o termo inquilino único geralmente é evitado. Em vez disso, todos os ambientes são caracterizados como multi-inquilinos, pois estão apenas implementando alguma variação de locação em que alguns ou todos os recursos são compartilhados ou dedicados.

## Apresentando o silo e a reserva

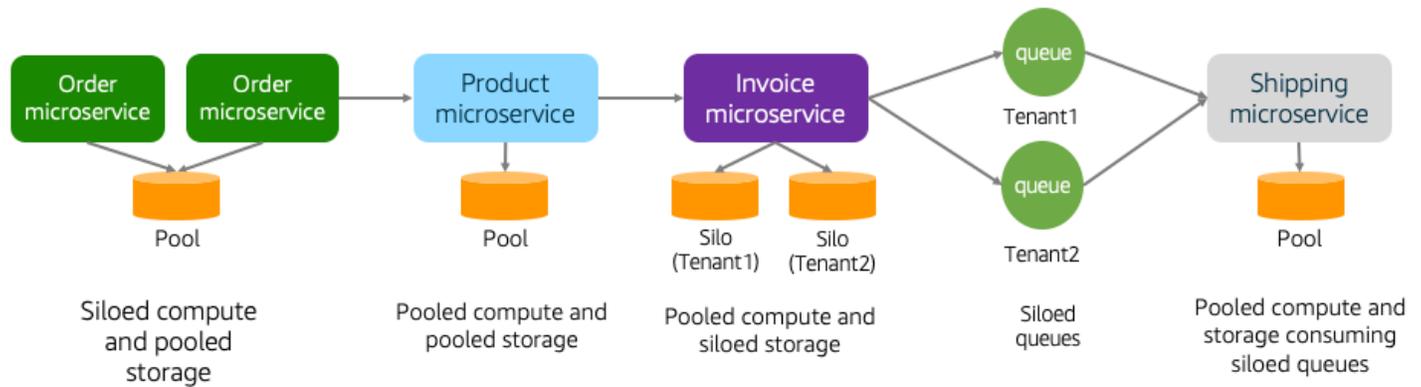
Com todas essas variações de modelos e considerando os desafios do termo multilocação, introduzimos algumas terminologias que nos permitem capturar e descrever com mais precisão os diferentes modelos usados na criação de um SaaS.

Dois termos que usamos para caracterizar o uso de recursos em um ambiente SaaS são silo e reserva. Esses termos nos permitem rotular a natureza dos ambientes SaaS, usando multi-inquilino como uma descrição abrangente que pode ser aplicada a qualquer número de modelos subjacentes.

No nível mais básico, o termo silo serve para descrever cenários em que um recurso é dedicado a um determinado inquilino. Por outro lado, o modelo de reserva é usado para descrever cenários em que um recurso é compartilhado por inquilinos.

Ao analisarmos como os termos silo e reserva são usados, é importante deixar claro que silo e reserva não são conceitos de tudo ou nada. O silo e a reserva podem ser aplicados a toda a pilha de recursos de um inquilino ou podem ser aplicados seletivamente a partes do seu ambiente SaaS geral. Portanto, se dissermos que algum recurso está usando um modelo de silo, isso não significa que todos os recursos desse ambiente estejam em silos. O mesmo vale para a forma como usaríamos o termo em reserva.

O diagrama a seguir fornece um exemplo de como os modelos em reserva e em silos são usados de forma mais granular em um ambiente SaaS:



## Modelos de silo e reserva

Esse diagrama inclui uma série de amostras destinadas a ilustrar a natureza mais específica dos modelos de silo e reserva. Se você seguir isso da esquerda para a direita, verá que começamos com um microserviço de pedidos. Esse microserviço tem computação em silos e armazenamento em reserva. Ele interage com um serviço de produto que tem computação e armazenamento em reserva.

O produto/serviço então interage com um microserviço de fatura que tem computação em reserva e armazenamento em silos. Esse serviço envia mensagens por meio de filas para o serviço de remessa. As filas são implantadas em um modelo em silos.

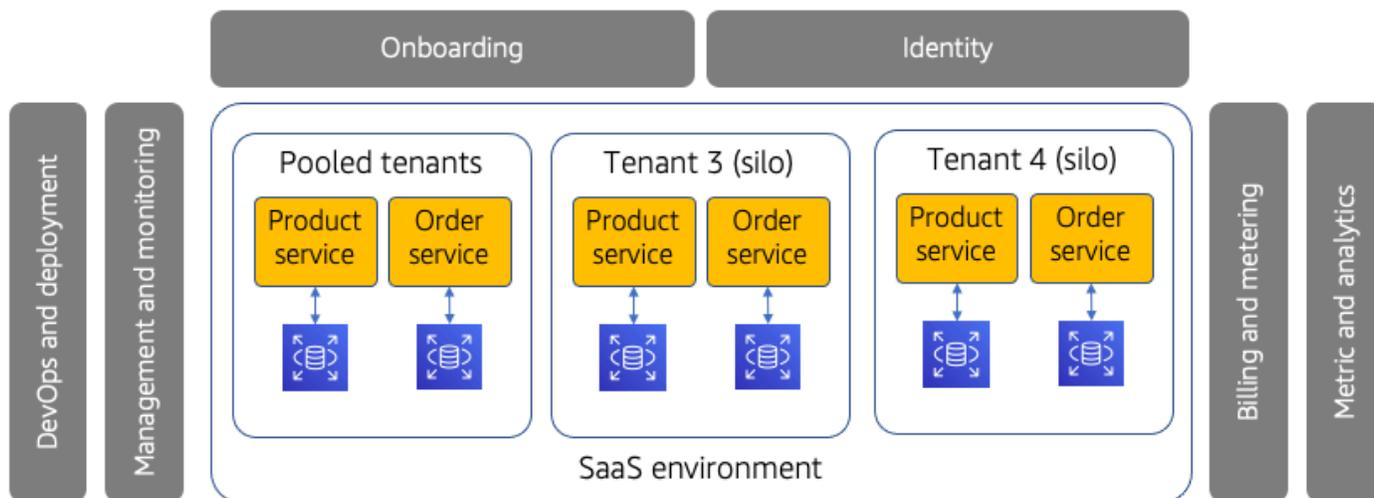
Por fim, o microserviço de remessa adquire mensagens das filas isoladas. Ele usa computação e armazenamento em reserva.

Embora isso possa parecer um pouco complicado, o objetivo é destacar a natureza granular dos conceitos de silo e reserva. Ao projetar e criar sua solução SaaS, espera-se que você tome essas decisões de silo e reserva com base nas necessidades de seu domínio e de seus clientes.

Vizinho barulhento, isolamento, hierarquização e uma série de outros motivos podem influenciar como e quando você escolhe aplicar o modelo de silo ou em reserva.

## Silo e reserva de pilha completa

O silo e a reserva também podem ser usados para descrever uma pilha SaaS inteira. Nessa abordagem, todos os recursos de um inquilino são implantados de forma dedicada ou compartilhada. O diagrama a seguir fornece um exemplo de como isso pode ser aplicado em um ambiente SaaS.



### Modelos completos de silo e reserva

Neste diagrama, você verá que existem três modelos diferentes para suas implantações de inquilinos de pilha completa. Primeiro, você verá que há um ambiente de reserva de pilha completo. Os inquilinos desse reserva compartilham todos os recursos (computação, armazenamento etc.).

As outras duas pilhas mostradas têm como objetivo representar ambientes de inquilinos em silos de pilha cheia. Nesse caso, o inquilino 3 e o inquilino 4 são mostrados como cada um com suas próprias pilhas dedicadas, onde nenhum dos recursos é compartilhado com outros inquilinos.

Essa mistura de modelos em silo e em reserva no mesmo ambiente SaaS não é tão atípica. Imagine, por exemplo, que você tenha um conjunto de inquilinos de nível básico que pagam um preço moderado pelo uso do sistema. Esses inquilinos são colocados no ambiente da reserva.

Enquanto isso, você também pode ter inquilinos de nível premium dispostos a pagar mais pelo privilégio de operar em um silo. Esses clientes são implantados com pilhas separadas (conforme mostrado no diagrama).

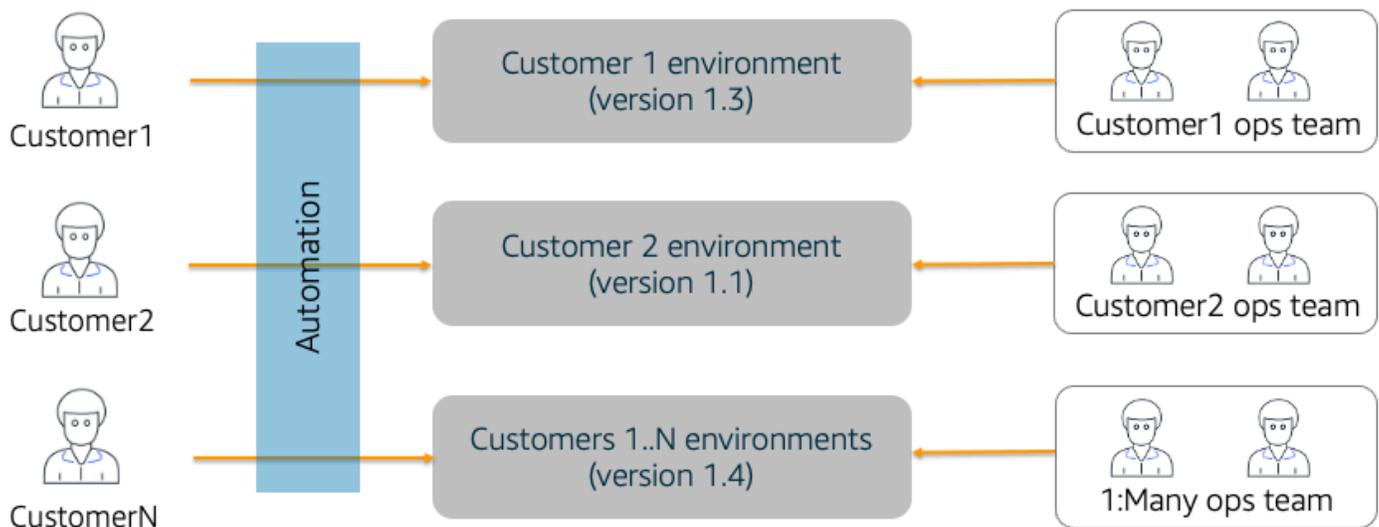
Mesmo nesse modelo, em que você pode ter permitido que os inquilinos trabalhassem em seu próprio silo completo, seria essencial que esses silos não permitissem nenhuma variação ou

personalização única para esses inquilinos. Em todos os aspectos, cada uma dessas pilhas deve estar executando a mesma configuração da pilha, com a mesma versão do software. Quando uma nova versão é lançada, ela é implantada no ambiente de inquilinos em reserva e em cada um dos ambientes isolados.

## SaaS versus Provedor de Serviços Gerenciado (MSP)

Também há alguma confusão que envolve as linhas entre os modelos SaaS e Provedor de Serviços Gerenciado (MSP). Se você observar um modelo MSP, ele pode parecer ter alguns objetivos semelhantes aos do modelo SaaS.

No entanto, se você se aprofundar um pouco mais no MSP, descobrirá que o MSP e o SaaS são, na verdade, diferentes. O diagrama a seguir fornece uma visão conceitual de um ambiente MSP.



### Modelo de Provedor de Serviços Gerenciado (MSP)

Esse diagrama representa uma abordagem do modelo MSP. À esquerda, você verá clientes que usam o modelo MSP. Geralmente, a abordagem aqui seria usar qualquer automação disponível para provisionar o ambiente de cada cliente e instalar o software para esse cliente.

À direita, há uma aproximação da área operacional que o MSP forneceria para dar suporte a esses ambientes de clientes.

É importante observar que o MSP geralmente instala e gerencia uma versão do produto que um determinado cliente deseja executar. Todos os clientes podem estar executando a mesma versão, mas isso normalmente não é necessário em um modelo MSP.

A estratégia geral é simplificar a vida de um fornecedor de software por meio da instalação e do gerenciamento desses ambientes. Embora isso simplifique a vida do provedor, ele não se relaciona diretamente com os valores e a mentalidade essenciais para uma oferta de SaaS.

O foco está em se livrar da responsabilidade gerencial. Fazer essa mudança não é o equivalente a fazer com que todos os clientes usem a mesma versão com uma experiência única e unificada de gerenciamento e operações. Em vez disso, o MSP geralmente permite versões separadas e geralmente trata cada um desses ambientes como operacionalmente separados.

Certamente, existem áreas em que o MSP pode começar a se sobrepor ao SaaS. Se o MSP basicamente exigisse que todos os clientes executassem a mesma versão e o MSP pudesse integrar, gerenciar, operar e faturar centralmente todos os inquilinos por meio de uma experiência, isso poderia começar a ser mais SaaS do que MSP.

O tema mais amplo é que automatizar a instalação de ambientes não significa ter um ambiente SaaS. Somente quando você adiciona todas as outras ressalvas discutidas anteriormente, isso representaria mais um verdadeiro modelo SaaS.

Se nos afastarmos dos aspectos tecnológicos e operacionais dessa história, a linha entre MSP e SaaS se tornará ainda mais distinta. Geralmente, como uma empresa de SaaS, o sucesso de sua oferta depende de sua capacidade de se envolver profundamente em todas as partes móveis da experiência.

Isso geralmente significa acompanhar a experiência de integração, entender como os eventos operacionais afetam os inquilinos, acompanhar as principais métricas e análises e estar perto do cliente. Em um modelo MSP em que isso é entregue a outra pessoa, você pode acabar ficando um nível distante dos principais detalhes que são essenciais para operar um negócio de SaaS.

# Migração SaaS

Muitos dos provedores que adotam o SaaS migram para o SaaS a partir de um modelo tradicional de software instalado (descrito anteriormente). Para esses provedores, é especialmente importante ter um bom alinhamento com os princípios fundamentais do SaaS.

Aqui, novamente, é onde pode haver alguma confusão sobre o que significa migrar para um modelo SaaS. Alguns, por exemplo, veem a mudança para a nuvem como uma migração para SaaS.

Outros consideram a adição de automação ao processo de instalação e provisionamento como uma conquista da migração.

É justo dizer que cada organização pode começar em um ponto diferente, ter diferentes considerações sobre o legado e provavelmente enfrentar diferentes pressões competitivas e de mercado. Isso significa que cada migração terá uma aparência diferente.

Ainda assim, embora cada caminho seja diferente, há algumas áreas em que há desconexões em torno dos princípios fundamentais que moldam as estratégias de migração. Ter um bom alinhamento em torno dos conceitos e princípios pode ter um impacto significativo no sucesso geral de sua migração para SaaS.

Com base nos conceitos descritos anteriormente, deve ficar claro que a mudança para o SaaS começa com a estratégia e as metas de negócios. Esse ponto pode se perder em um ambiente de migração em que há pressão para acessar o SaaS o mais rápido possível.

Nesse modo, as organizações geralmente veem a migração principalmente como um exercício técnico. A realidade é que toda migração de SaaS deve começar com uma visão clara dos clientes-alvo, da experiência do serviço, das metas operacionais e assim por diante. Ter um foco mais claro em como sua empresa de SaaS precisa ter um impacto profundo na forma, nas prioridades e no caminho que você segue para migrar sua solução para SaaS.

Ter essa visão clara desde o início da migração estabelece a base de como você migra a tecnologia e os negócios como parte de sua mudança para o SaaS. Ao seguir esse caminho, concentre-se nas perguntas que podem lhe dizer mais sobre para onde você está indo.

A tabela a seguir fornece uma visão da natureza contrastante das mentalidades de migração técnica e comercial.

Tabela 1 — Primeira migração técnica versus migração empresarial

A mentalidade que prioriza a tecnologia	A mentalidade que prioriza os negócios
Como isolamos os dados do inquilino?	Como o SaaS pode nos ajudar a expandir nossos negócios?
Como conectamos os usuários aos inquilinos?	Quais segmentos estamos almejando?
Como evitamos condições barulhentas de vizinhos?	Qual é o tamanho e o perfil desses segmentos?
Como fazemos testes A/B?	Quais níveis precisaremos oferecer suporte?
Como escalamos com base na carga de inquilinos?	Qual experiência de serviço estamos almejando?
Qual provedor de cobrança devemos usar?	Qual é a nossa estratégia de preços e embalagens?

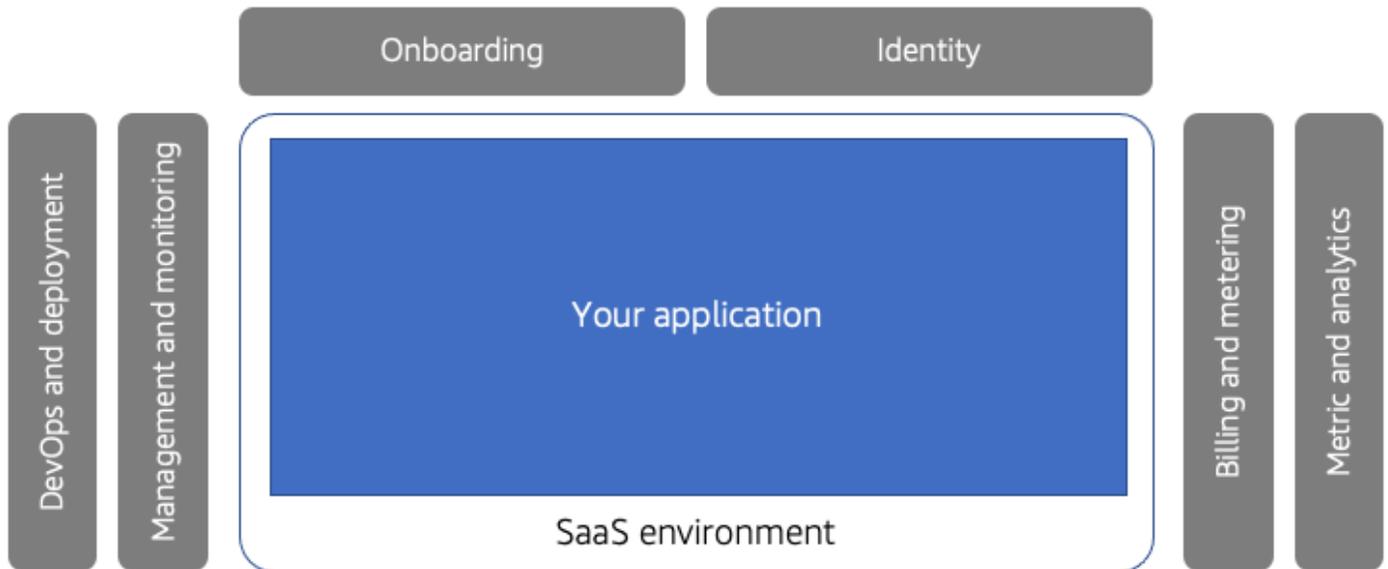
Veja à esquerda um exemplo da aparência de uma mentalidade de migração tecnológica pioneira. A equipe de engenharia está muito focada em abordar os tópicos clássicos de multi-inquilinos que certamente são importantes para qualquer arquitetura SaaS.

O problema é que as respostas para muitas das perguntas à esquerda geralmente são diretamente influenciadas pelas respostas à pergunta à direita. É improvável que esse ponto seja novo para quem está analisando a migração. No entanto, a realidade é que muitas organizações começam buscando a eficiência operacional e de custos como primeira etapa, presumindo que as partes do negócio se resolverão sozinhas.

Nessa estratégia de migração, também pode haver confusão sobre como seu ambiente legado pode evoluir para se adequar a um modelo SaaS. Essa também é uma área em que há uma infinidade de opções para migrar para o SaaS. No entanto, há um sistema de valores comum que geralmente defendemos para qualquer migração.

Em nossa discussão anterior sobre os princípios de SaaS, descrevemos diferentes padrões e terminologias que são usados para descrever ambientes SaaS. Um tema comum que abrange todas essas soluções é a ideia de ter serviços compartilhados que cercam seu aplicativo. Identidade, integração, métricas, faturamento — todos esses elementos são considerados elementos comuns que são essenciais para qualquer ambiente SaaS.

Agora, ao analisarmos a migração, você verá que esses mesmos serviços compartilhados desempenham um papel fundamental em qualquer história de migração. O diagrama a seguir fornece uma visão conceitual do cenário de migração.



## Migração para SaaS

Esse diagrama representa a experiência desejada para qualquer caminho de migração. Ele inclui todos os mesmos serviços compartilhados que foram descritos anteriormente. No meio, há um espaço reservado para sua inscrição.

A ideia principal é que você possa colocar qualquer número de modelos de aplicativos no meio desse ambiente. Sua primeira etapa na migração pode fazer com que cada inquilino funcione em seu próprio silo. Ou você pode ter uma arquitetura híbrida em que os elementos são isolados e outras partes da funcionalidade são tratadas por meio de uma coleção de microsserviços modernizados.

O quanto você moderniza inicialmente seu aplicativo variará com base na natureza do seu ambiente legado, nas necessidades do mercado, nas considerações de custo e assim por diante. O que não muda é a introdução desses serviços compartilhados.

Qualquer migração de SaaS precisa oferecer suporte a esses serviços compartilhados fundamentais para dar à sua empresa a capacidade de operar em um modelo SaaS. Todas as variações da arquitetura do aplicativo precisam de identidade SaaS, por exemplo. Você precisará de operações com reconhecimento de inquilinos para gerenciar e monitorar sua solução SaaS.

A implementação desses serviços compartilhados na frente da migração permite que você apresente uma experiência de SaaS aos seus clientes, mesmo que o aplicativo subjacente ainda esteja sendo executado em um silo completo para cada inquilino.

O objetivo geral é fazer com que seu aplicativo seja executado em um modelo SaaS. Em seguida, você pode voltar sua atenção para uma maior modernização e refinamento de seu aplicativo. Essa abordagem também permite que você mova as outras partes do seu negócio (marketing, vendas, suporte etc.) em um ritmo mais rápido. Mais importante ainda, isso permite que você comece a engajar e coletar feedback do cliente, que pode ser usado para moldar a modernização contínua de seu ambiente.

É importante observar que os serviços compartilhados que você implementa podem não incluir todos os recursos ou mecanismos necessários. O objetivo principal é criar os mecanismos compartilhados necessários no início da migração. Isso permite que você se concentre nos elementos do sistema que são essenciais para a evolução da arquitetura do aplicativo e da evolução operacional.

# Identidade SaaS

SaaS adiciona novas considerações ao modelo de identidade do seu aplicativo. À medida que cada usuário é autenticado, ele deve estar conectado a um contexto específico de inquilino. Esse contexto de inquilino fornece informações essenciais sobre seu inquilino que são usadas em todo o seu ambiente SaaS.

Essa vinculação de inquilinos a usuários geralmente é chamada de identidade SaaS do seu aplicativo. À medida que cada usuário se autentica, seu provedor de identidade normalmente gera um token que inclui tanto a identidade do usuário quanto a identidade do inquilino.

Conectar inquilinos a usuários representa um aspecto fundamental de sua arquitetura SaaS que tem muitas implicações posteriores. O token desse processo de identidade flui para os microsserviços do seu aplicativo e é usado para criar registros com reconhecimento de inquilinos, registrar métricas, medir o faturamento, impor o isolamento do inquilino e assim por diante.

É essencial que você evite cenários que dependam de mecanismos separados e autônomos que mapeiam usuários para inquilinos. Isso pode prejudicar a segurança do seu sistema e, muitas vezes, cria gargalos em sua arquitetura.

# Isolamento de inquilinos

Quanto mais você move os clientes para um modelo multi-inquilino, mais eles se preocupam com a possibilidade de um inquilino acessar os recursos de outro inquilino. Os sistemas SaaS incluem mecanismos explícitos que garantem que os recursos de cada inquilino, mesmo que sejam executados em uma infraestrutura compartilhada, sejam isolados.

Isso é o que chamamos de isolamento de inquilinos. A ideia por trás do isolamento de inquilinos é que sua arquitetura SaaS introduza estruturas que controlam rigorosamente o acesso aos recursos e bloqueiam qualquer tentativa de acessar recursos de outro inquilino.

Observe que o isolamento do inquilino é separado dos mecanismos gerais de segurança. Seu sistema suportará autenticação e autorização; no entanto, o fato de um usuário inquilino estar autenticado não significa que seu sistema tenha alcançado o isolamento. O isolamento é aplicado separadamente da autenticação e autorização básicas que podem fazer parte do seu aplicativo.

Para entender melhor isso, imagine que você tenha usado um provedor de identidade para autenticar o acesso ao seu sistema SaaS. O token dessa experiência de autenticação também pode incluir informações sobre a função de um usuário, que podem ser usadas para controlar o acesso desse usuário a uma funcionalidade específica do aplicativo. Essas estruturas fornecem segurança, mas não isolamento. Na verdade, um usuário pode ser autenticado e autorizado e ainda acessar os recursos de outro inquilino. Nada sobre autenticação e autorização necessariamente bloqueará esse acesso.

O isolamento do inquilino se concentra exclusivamente no uso do contexto do inquilino para limitar o acesso aos recursos. Ele avalia o contexto do inquilino atual e usa esse contexto para determinar quais recursos estão acessíveis para esse inquilino. Ele aplica esse isolamento a todos os usuários desse inquilino.

Isso fica mais desafiador à medida que analisamos como o isolamento de inquilinos é realizado em todos os diferentes padrões de arquitetura SaaS. Em alguns casos, o isolamento pode ser obtido com pilhas inteiras de recursos dedicadas a um inquilino, onde políticas de rede (ou mais grosseiras) impedem o acesso entre inquilinos. Em outros cenários, você pode ter recursos em reserva (itens em uma tabela do [Amazon DynamoDB](#)) que exigem políticas mais refinadas para controlar o acesso aos recursos.

Qualquer tentativa de acessar um recurso do inquilino deve ter como escopo apenas os recursos que pertencem a esse inquilino. O trabalho dos desenvolvedores e arquitetos de SaaS é determinar

qual combinação de ferramentas e tecnologias suportará os requisitos de isolamento de seu aplicativo específico.

# Particionamento de dados

O particionamento de dados é usado para descrever diferentes estratégias usadas para representar dados em um ambiente multi-inquilino. Esse termo é usado amplamente para abranger uma variedade de abordagens e modelos diferentes que podem ser usados para associar diferentes estruturas de dados a inquilinos individuais.

Observe que muitas vezes há a tentação de ver o particionamento de dados e o isolamento de inquilinos como intercambiáveis. Esses dois conceitos não pretendem ser equivalentes. Quando falamos sobre particionamento de dados, estamos falando sobre como os dados do inquilino são armazenados para inquilinos individuais. O particionamento de dados não garante que os dados sejam isolados. O isolamento ainda deve ser aplicado separadamente para garantir que um inquilino não possa acessar os recursos de outro inquilino.

Cada tecnologia AWS de armazenamento traz seu próprio conjunto de considerações à estratégia de particionamento de dados. Por exemplo, isolar dados no Amazon DynamoDB terá uma aparência muito diferente de isolar dados com o Amazon [Relational Database Service \(Amazon RDS\)](#).

Geralmente, quando você pensa em particionamento de dados, começa pensando se os dados serão separados ou em reserva. Em um modelo em silos, você tem uma estrutura de armazenamento distinta para cada inquilino, sem dados misturados. Para particionamento em reserva, os dados são misturados e particionados com base em um identificador de inquilino que determina quais dados estão associados a cada inquilino.

Por exemplo, com o Amazon DynamoDB, um modelo em silos usa uma tabela separada para cada inquilino. O agrupamento de dados no Amazon DynamoDB é obtido armazenando o identificador do inquilino na chave de partição de cada tabela do Amazon DynamoDB que gerencia os dados de todos os inquilinos.

Você pode imaginar como isso pode variar em toda a variedade de AWS serviços, com cada um introduzindo suas próprias estruturas que podem exigir uma abordagem diferente para a realização de modelos de armazenamento em silo e em reserva com cada serviço.

Embora o particionamento de dados e o isolamento de inquilinos sejam tópicos separados, as estratégias de particionamento de dados que você escolher provavelmente serão influenciadas pelo modelo de isolamento de seus dados. Por exemplo, você pode isolar algum armazenamento porque essa abordagem se alinha melhor aos requisitos do seu domínio ou dos seus clientes. Ou você pode escolher o silo porque o modelo de reserva pode não permitir que você imponha o isolamento com o nível de granularidade que sua solução exige.

Um vizinho barulhento também pode afetar sua abordagem ao isolamento. Algumas cargas de trabalho ou casos de uso em seu aplicativo podem precisar ser mantidos separados para limitar os impactos de outros inquilinos ou para atender aos contratos de nível de serviço (SLAs).

# Medição, métricas e cobrança

As discussões sobre SaaS também tendem a incluir a noção de medição, métricas e cobrança. Esses conceitos geralmente são agrupados em um único conceito. No entanto, é importante distinguir as diferentes funções que a medição, as métricas e o faturamento desempenham em um ambiente SaaS.

O desafio desses conceitos é que eles geralmente têm usos sobrepostos da mesma palavra. Por exemplo, podemos falar sobre a medição usada para gerar sua fatura. Ao mesmo tempo, também podemos falar sobre a medição usada para rastrear o consumo interno de recursos que não estão conectados ao faturamento. Também falamos sobre métricas e SaaS em muitos contextos que podem se misturar nessa discussão.

Para ajudar a resolver isso, vamos associar alguns conceitos específicos a cada um desses termos (sabendo que não há absolutos aqui).

- **Medição** — Esse conceito, embora tenha muitas definições, se encaixa melhor no domínio de faturamento do SaaS. A ideia é que você meça a atividade do inquilino ou o consumo de recursos para coletar os dados necessários para gerar uma fatura.
- **Métricas** — As métricas representam todos os dados que você captura para analisar tendências em seus domínios de negócios, operações e tecnologia. Esses dados são usados em vários contextos e funções dentro da equipe de SaaS.

Essa distinção não é fundamental, mas ajuda a simplificar a forma como pensamos sobre o papel da medição e das métricas em um ambiente SaaS.

Agora, se conectarmos esses dois conceitos a exemplos, você pode pensar em instrumentar seu aplicativo com eventos de medição específicos usados para revelar os dados necessários para gerar uma fatura. Isso pode ser o número de solicitações, o número de usuários ativos ou pode ser mapeado para algum agregado de consumo (solicitações, CPU, memória) relacionado a alguma unidade que faça sentido para seus clientes.

Em seu ambiente SaaS, você publicará esses eventos de cobrança do seu aplicativo e eles serão ingeridos e aplicados pela estrutura de cobrança empregada pelo seu sistema SaaS. Isso pode ser um sistema de cobrança de terceiros ou algo personalizado.

Por outro lado, a mentalidade por trás das métricas é capturar as ações, atividades, padrões de consumo etc. que são essenciais para avaliar a saúde e a pegada operacional imposta ao seu

sistema por diferentes inquilinos. As métricas que você publica e agrega aqui são ditadas mais pela necessidade de diferentes personas (equipes operacionais, proprietários de produtos, arquitetos e assim por diante). Aqui, esses dados métricos são publicados e agregados em algumas ferramentas de análise que permitem que esses diferentes usuários criem visualizações da atividade do sistema que analisem os aspectos do sistema que melhor se alinham à sua personalidade. O proprietário do produto pode querer entender como diferentes inquilinos estão consumindo recursos. Um arquiteto pode precisar de visualizações que o ajudem a entender como os inquilinos estão consumindo recursos de infraestrutura e assim por diante.

## SaaS B2B e B2C

As ofertas de SaaS são criadas para os mercados B2B e B2C. Embora os mercados e os clientes certamente tenham dinâmicas diferentes, os princípios gerais do SaaS não mudam de alguma forma cada um desses mercados.

Se você observar a integração, por exemplo, clientes B2B e B2C podem ter experiências de integração diferentes. É verdade que os sistemas B2C podem se concentrar mais em um fluxo de integração de autoatendimento (embora o sistema B2B também possa suportar isso).

Embora possa haver diferenças na forma como a integração é oferecida aos clientes, os valores fundamentais subjacentes à integração são basicamente os mesmos. Mesmo que sua solução B2B dependa de um processo interno de integração, ainda esperamos que esse processo seja o mais simples e automatizado possível. Ser B2B não significa, de alguma forma, que estamos mudando nossas expectativas com o passar do tempo para gerar valor para nossos clientes.

## Conclusão

O objetivo deste documento é delinear os conceitos fundamentais da arquitetura SaaS, fornecendo alguns esclarecimentos sobre os modelos e a terminologia usados para caracterizar os padrões e estratégias de SaaS. A esperança é que isso forneça às organizações uma visão mais clara do cenário geral de SaaS.

Muito do que foi abordado aqui se concentrou no que significa ser SaaS, com ênfase significativa na criação de um ambiente que permite gerenciar e operar todos os seus inquilinos de SaaS por meio de uma experiência unificada. Isso se conecta à ideia central de que o SaaS é, antes de tudo, um modelo de negócios. A arquitetura SaaS que você cria tem como objetivo promover essas metas comerciais fundamentais.

## Outras fontes de leitura

Há vários recursos que detalham os padrões de arquitetura SaaS que estão em conformidade com os padrões descritos aqui.

Para obter informações adicionais, consulte:

- [Estratégias de isolamento de inquilinos de SaaS](#) (whitepaper do AWS)
- [Estratégias de armazenamento SaaS](#) (whitepaper do AWS)
- [Well-Architected SaaS Lens](#) (whitepaper do AWS)

# Colaboradores

Os seguintes indivíduos e organizações contribuíram para este documento:

- Tod Golding, principal arquiteto de soluções de parceiros do AWS SaaS Factory

# Revisões do documento

Para ser notificado sobre atualizações desse whitepaper, inscreva-se no feed RSS.

Alteração	Descrição	Data
<a href="#">Publicação inicial</a>	Publicação do whitepaper.	3 de agosto de 2022

# Avisos

Os clientes são responsáveis por fazer sua própria avaliação independente das informações contidas neste documento. Este documento: (a) é apenas para fins informativos, (b) representa as ofertas e práticas de produtos atuais da AWS, que estão sujeitas a alterações sem aviso prévio e (c) não criam nenhum compromisso ou garantia da AWS e de suas afiliadas, fornecedores ou licenciadores. Os produtos ou serviços da AWS são fornecidos “no estado em que se encontram”, sem garantias, representações ou condições de qualquer tipo, expressas ou implícitas. As responsabilidades e as obrigações da AWS com os seus clientes são controladas por contratos da AWS, e este documento não é parte, nem modifica, qualquer contrato entre a AWS e seus clientes.

© 2023 Amazon Web Services, Inc. ou suas afiliadas. Todos os direitos reservados.

# Glossário do AWS

Para obter a terminologia mais recente da AWS, consulte o [glossário da AWS](#) na Referência do Glossário da AWS.