

Modelo de responsabilidade

DigitalHouse>



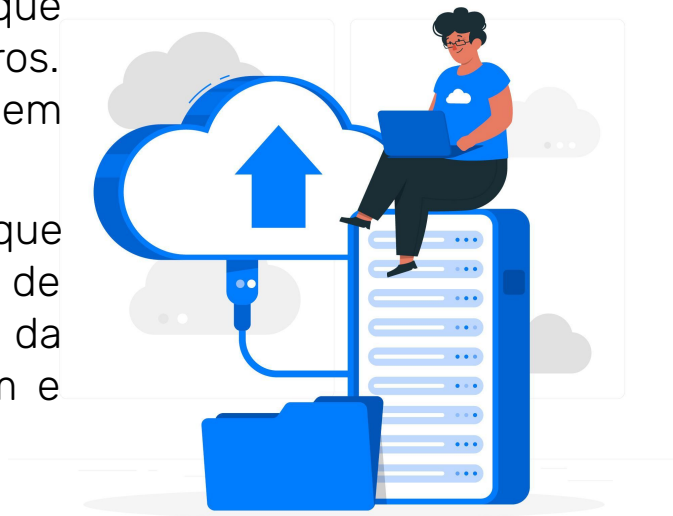
**Certified Tech
Developer**

The Ultimate Degree

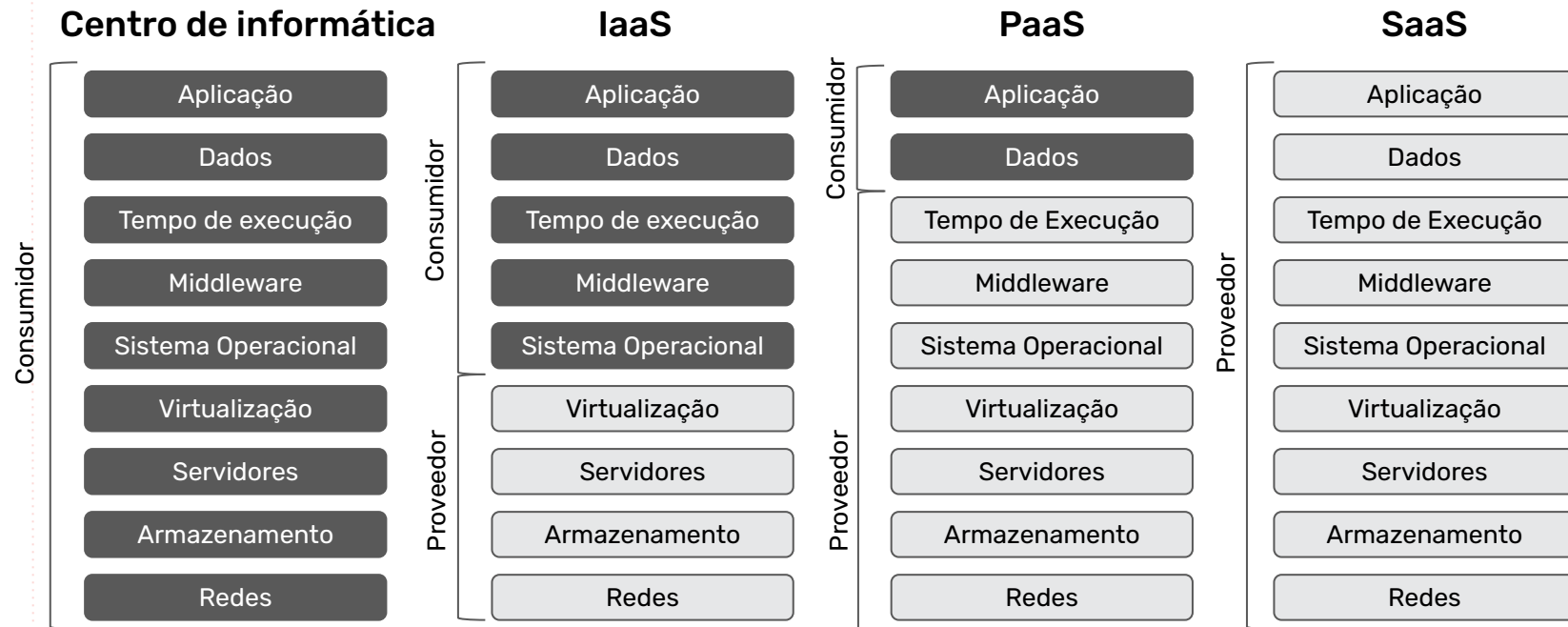
Porque é necessário?

Como já vimos, na computação em nuvem o que fazemos é usar recursos de terceiros. Basicamente nos tornamos inquilinos de quem possui os ativos de TI.

O modelo de responsabilidades é um padrão que nos permite definir diferentes tipos de contratação de serviços ao proprietário da nuvem, quais tarefas são delegadas à nuvem e qual será nossa responsabilidade.



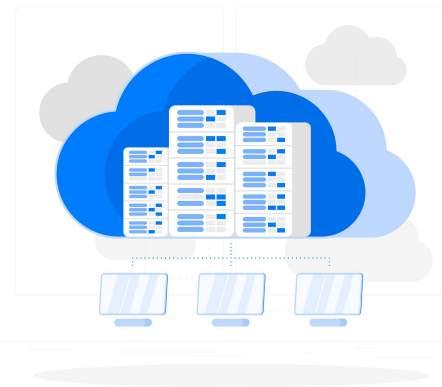
0 modelo



Vamos analisar o modelo

Se olharmos para o modelo podemos identificar:

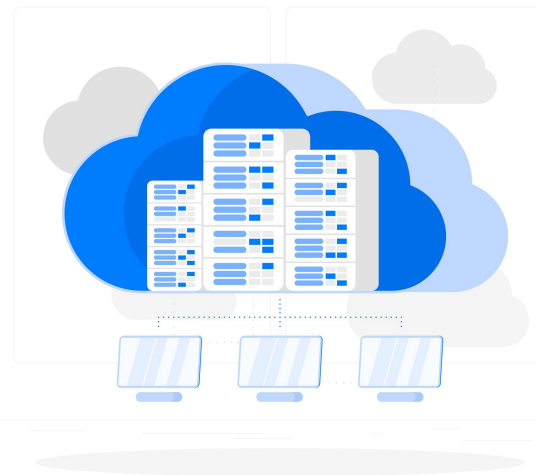
- Colunas: centro de computação, IaaS, PaaS e SaaS.
- Funções: consumidor e fornecedor.
- Boxes que definem um conjunto de tarefas, tecnologias e disciplinas com as quais já estamos familiarizados, como: redes, virtualização ou middleware.
- E, por fim, núcleos que tingem cada um deles, dependendo da coluna em que nos encontramos.
- Más o que eles significam?



Os papéis: consumidor e provedor

Usamos esses dois papéis para definir quem é o responsável. Já vimos que na computação em nuvem o que realmente fazemos é contratar recursos de forma flexível e com interfaces bem definidas –como APIs– para sua administração por terceiros.

Esse terceiro, proprietário da nuvem ou dos centros de computação em que decidimos executar nossos aplicativos, é o que chamaremos de **provedor**.

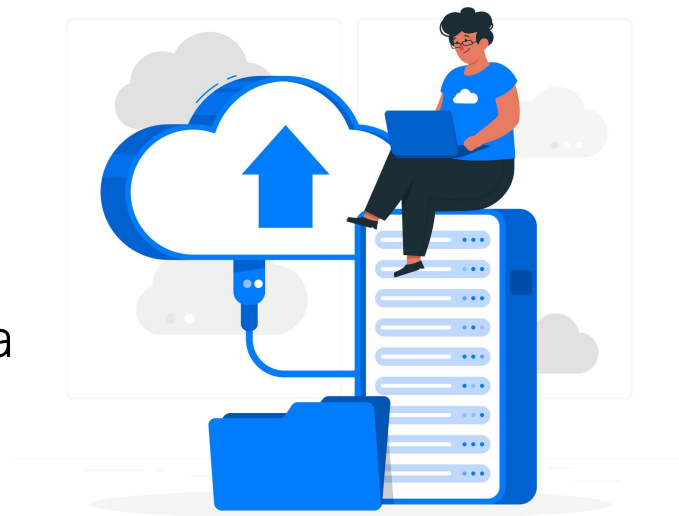


Os papéis: consumidor e provedor

E quem é o consumidor?

A resposta é simples, nós!

Que decidimos contratar os serviços provedor para executar nossos aplicativos em seus data centers.



As colunas: centro de computação

Quando falamos de um **centro de informática**, o que fazemos é definir quais tarefas seríamos responsáveis se somos nós que hospedamos todos os serviços de tecnologia. Em outras palavras, não contratamos nenhum fornecedor para hospedar ou gerenciar nossos sistemas.



Prós: Temos controle total sobre os ativos, podemos decidir como, para quê, onde e quando os usamos.

Contras: Temos total responsabilidade pela saúde, uso e manutenção do parque informático. Não podemos escalar com flexibilidade.



As colunas: IaaS

Quando falamos de **IaaS** nos referimos a uma sigla em inglês: Infrastructure as a Service (infraestrutura como serviço, em português). Nesse modelo de responsabilidade, o provedor nos dá a possibilidade de instanciar máquinas virtuais em seu data center sem que tenhamos que gerenciar a infraestrutura subjacente.



Prós: Ainda temos controle sobre o sistema operacional, ambiente ideal para instalação de aplicativos legados. Ele nos permite dimensionar a infraestrutura de forma mais dinâmica.

Contras: Ainda somos responsáveis por saúde e gerenciamento do sistema operacional e todos os serviços e aplicativos que são correr em cima dele.



As colunas: PaaS

Quando falamos de **PaaS** nos referimos a uma sigla em inglês: Platform as a Service (em português, plataforma como serviço). Nesse modelo de responsabilidade, o provedor nos oferece uma tecnologia específica, exposta por meio de interfaces definidas, sejam gráficas ou APIs, e nos abstrai de todo o gerenciamento dos recursos subjacentes.



Prós: Abstraimos completamente o gerenciamento dos recursos das camadas inferiores. Delegamos a gestão, monitoramento e escalabilidade da tecnologia ao fornecedor. Permitindo-nos focar totalmente na entrega de valor.

Contras: Tende a ser uma opção cara, pois nos abstrai de todo o músculo necessário para gerenciar e monitorar a infraestrutura subjacente. As configurações disponíveis são limitadas àquelas que o provedor optou por expor ou disponibilizar.



Exemplo de coluna PaaS

O modelo **PaaS** tende a ser difícil de visualizar quando somos inexperientes. Vamos dar um exemplo para nos ajudar: **Amazon DynamoDB**.

O DynamoDB é um banco de dados não estruturado, semelhante ao **MongoDB**. Mas, se quisermos usar o MongoDB, precisamos instanciar a infraestrutura necessária e instalá-la; Podemos consumir o Amazon DynamoDB, simplesmente, como um serviço.



Exemplo de coluna PaaS



O que significa isto? Que uma vez inscrito no serviço através no site da Amazon AWS, eles nos fornecerão:

- Uma interface web através da qual podemos fazer algumas configurações do serviço Dynamo.
- Uma API que nos permitirá configurar os mesmos parâmetros da interface web, mas programaticamente.
- E uma cadeia de conexão que usamos em nosso aplicativo para conectar ao banco de dados do Dynamo.

Dessa forma, não tivemos a necessidade de instalar nenhum recurso. Basta subscrever o serviço e a plataforma ou tecnologia fica disponível para nosso uso, através da Internet.

As colunas: SaaS

Quando falamos de **SaaS** nos referimos a uma sigla em inglês: Software as a Service (em português, software as a service). Nesse modelo de responsabilidade, o provedor nos oferece um aplicativo. A melhor maneira de entender esse modelo de responsabilidade é pensar em exemplos: **Trello**, **Salesforce** ou **Gmail** são exemplos de SaaS.



Prós: Com apenas um cartão de crédito podemos acessar um aplicativo pronto para usar e resolver uma necessidade. E sem ter que arcar com o custo de sua infraestrutura e operação.

Contras: Pouco ou nenhum controle sobre a configuração do sistema contratado. Dependendo da flexibilidade do software, podemos ter que adaptar nossos processos ao que o software nos permite fazer. Não há controle sobre os novos recursos do sistema e quando eles são lançados.



DigitalHouse>