

# *Software as a Service: Uma Compreensão Pluralista*

**Katysuco FARIAS SANTOS (1); Edigley PEREIRA FRAGA (2)**

(1) Instituto Federal de Educação, Ciência e Tecnologia de Pernambuco, Av. Prof. Luiz Freire, 500, Cidade Universitária, Recife - PE, CEP: 50.740-540, Brasil, e-mail: [katysuco@recife.ifpe.edu.br](mailto:katysuco@recife.ifpe.edu.br)

(2) Universidade Federal de Campina Grande, Rua Aprígio Veloso, 882, Bairro Universitário, Campina Grande – PB, CEP: 58.429-140, Brasil, e-mail: [edigley@gmail.com](mailto:edigley@gmail.com)

## **RESUMO**

O acúmulo e a diversidade de Tecnologias da Informação criam, hoje, a possibilidade de computação nas nuvens (*cloud computing*), e faz ressurgir a possibilidade de oferecer software como serviço (*software as a service* – *SaaS*). É um termo da moda, que aponta como alta fonte de pesquisa para academia, e como um promissor mercado a ser explorado pela indústria de software. Logo, divergências sobre o que é essa tecnologia é uma realidade que a confunde com outras já existentes e que dificulta a sua disseminação. Para ser completamente compreendido, *SaaS* precisa ser definido e analisado por diferentes perspectivas: forma de entrega, comercialização e negócio. Este é um trabalho teórico, que metodologicamente parte de uma revisão bibliográfica, acadêmica e mercadológica, sobre as diversas perspectivas para *SaaS*, e suas tecnologias associadas. O objetivo é proporcionar um entendimento pluralista sobre *SaaS*, interligando essas suas diferentes perspectivas, e sobre elas encaixando as diversas tecnologias já existentes. Padronizando o entendimento sobre os conceitos, e tendo a consciência sobre a inviabilidade de *SaaS* ser visto apenas por uma única ótica, este trabalho contribui ainda com a apresentação dos principais benefícios e desafios quando da adoção de *SaaS* por fornecedores e clientes.

**Palavras-chave:** software as a service, *SaaS*, *cloud computing*

## **1 INTRODUÇÃO**

O ideal de computação utilitária vislumbrado por John McCarthy em 1961 foi perseguido em diversos momentos e de diferentes maneiras durante os quase 50 anos posteriores. As tentativas foram desde os birôs de serviços nos anos 60 e 70, passando pelos Provedores de Serviços de Aplicação (ASP) nas décadas de 80 e 90 até chegar na oferta do Software como Serviço (*Software as a Service* - *SaaS*) a partir do ano 2000. O principal ponto que diferencia as tentativas anteriores da atual, conhecida pelo hiperônimo "*cloud computing*", é o amadurecimento de um conjunto de tecnologias e a ocorrência de uma atmosfera favorável.

O termo *cloud computing* passou a ser bastante explorado tanto pelas empresas ligadas à Tecnologia da Informação (TI) quanto pelos canais de mídia tradicionais a exemplo de revistas e televisão. Os motivos para tanto são bastante diretos, pois os primeiros pretendem se manter competitivos no mercado e a mídia refletiu a ampla adoção de serviços on-line pelos usuários da internet. Ao mesmo tempo, o ambiente acadêmico tem realizado bastante pesquisa na área e levantado desafios, proposto padronizações, taxonomias e ferramentas, bem como avaliado soluções e modelos. Um dos termos que está englobado em *cloud computing* é o de Software como Serviço, que também tem gerado alvoroço no ambiente corporativo e acadêmico.

Embora a idéia de prover *software* como serviço, nos moldes do que se tem hoje, não ser algo novo, pois Bennet et. al. [7], já o discutia em 1999, e muito já se tenha discutido a respeito de *SaaS*, ainda há bastantes equívocos sobre o tema, principalmente no que se refere a definições e diferenciações entre *SaaS* e tecnologias, modelos e conceitos já existentes. Tem contribuído bastante para esses entendimentos equivocados a exploração comercial do termo, que por motivos de *marketing* é utilizado muitas vezes apenas como *buzzword*, e também o fato de ser algo ainda novo e que pretende revolucionar a maneira como se é atualmente encarado o produto software.

Levando em consideração esse panorama, é objetivo do presente trabalho apresentar uma visão pluralista do Software como Serviço, visto ser algo que possui implicações na forma de entrega do software e na exploração do mercado de TI. A seção 2 relata como o papel e os serviços de TI evoluíram desde os anos

1970 até os dias de hoje, e detalha a pilha arquitetural para aplicações em computação nas nuvens. Em seguida, na seção 3, é dada ênfase às visões de *SaaS* como modelo de entrega e de negócio e apresenta as principais características de *SaaS*. Vantagens para clientes e fornecedores ao adotarem *SaaS*, bem como os fatores que inibem a sua adoção, são elencados e discutidos na seção 4. Por fim, na seção 5, são apresentadas nossas conclusões.

## 2 CONTEXTO HISTÓRICO E ATUAL

### 2.1 Evolução da TI

Os conceitos e modelos associados ao Software como Serviço, tem influência em como a Tecnologia da Informação (TI) corporativa acumulou conhecimento e tecnologias ao longo de algumas décadas [4]. Durante os anos 1970, prevaleciam na TI os sistemas monolíticos operados por pessoal extremamente treinado e qualificado dentro de imensos centros de processamento com altos custos de manutenção e operação. Em 1980, dissemina-se o uso dos computadores pessoais de baixo custo, surgem as redes de computadores corporativas, interoperáveis e interligadas por equipamentos: de *bridges* a roteadores. Isso possibilitou que empresas integrassem ambientes de computação e sistemas, via rede. Nos anos 1990, consolida-se a arquitetura cliente/servidor e a *Web* surge como uma rede pública de baixo custo disponível para universidades, empresas e usuários domésticos. Também, surgem novos e mais modernos equipamentos de interconexão, ampliando as capacidades de comunicação e integração entre sistemas. Na virada deste século, padrões de *WebServices* (XML, SOAP, UDDI, WSDL) permitiram que sistemas fossem integrados mais rapidamente, suportando um número crescente de usuários, e aplicações passassem a ser definitivamente expostas através da internet. Ganham visibilidade, as redes sociais, (*blogs*, *wikis*). Nos dias atuais, temos as arquiteturas orientadas a serviço (SOA) [14] e tecnologias, como Ajax, que possibilitam a construção de interfaces ricas e funcionais para *thin-client*. Alias-se ainda, o incremento da capilaridade de pontos de acesso à Internet, inclusive com banda larga, e a necessidade de se ter acesso ubíquo aos dados. Surge assim, uma TI mais dinâmica, flexível e híbrida, que combina serviços locais com serviços remotos hospedados em *datacenters* diversos. São as "nuvens" de TI, "*cloud computing*".

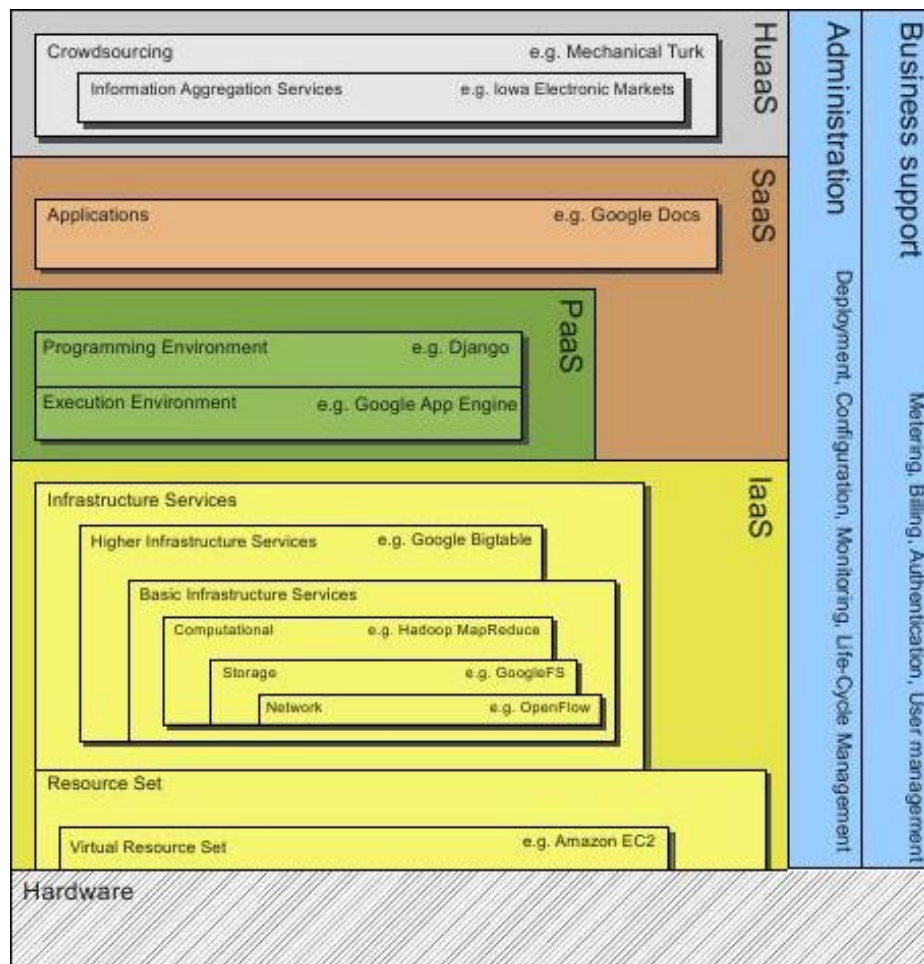
### 2.2 Cloud Computing

*Cloud computing* (computação nas nuvens) é para a indústria (grandes *players* como IBM, Google, Microsoft, Salesforce) um novo salto de tecnologia. Para a academia é um *hot-topic research*, prova disso é o surgimento de conferências e workshop sobre a temática, e, justifica a existência, na literatura de divergências e desentendimentos sobre o que realmente vem a ser essa tecnologia.

*Cloud* ("nuvem") é uma metáfora para a Internet (baseado em como é diagramada uma rede) e é uma abstração para a infra-estrutura complexa que esconde [8]. Está relacionada com as capacidades a que são fornecidos, tal como um serviço, permitindo que usuários acessem serviços de tecnologia a partir da Internet, sem o conhecimento onde estão localizados, e sem controle sobre a infra-estrutura tecnológica que os suporta. De forma didática computação nas nuvens pode ser definida como um modelo no qual a computação (processamento, armazenamento e softwares): está disponível em algum lugar da rede de forma escalável, é acessada remotamente, via internet, e seu pagamento se dá em função das demandas de computação.

Dada a diversidade de tecnologias emergentes para computação nas nuvens, alguns trabalhos [17] [12] apontam em direção a uma taxonomia e uma arquitetura de pilha de serviços padronizados (figura 1) que sirvam como referência para comparações, discussões e novas implementações. É através dessa pilha arquitetural que é possível prover e consumir recursos físicos, infraestrutura virtualizada, plataformas de *middleware* e aplicações, como serviços.

O nível *IaaS* - *Infrastructure as a Service* -, possui uma camada próxima ao *hardware*, a *Resource Set*, que engloba um conjunto de funcionalidades primárias (*e.g.* inicialização, configuração) e as disponibiliza através de uma API às camadas superiores da pilha. Divide-se em *Physical Resource Set (PRS)* cuja implementação é dependente do *hardware*, e *Virtual Resource Set (VRS)* implementado sob virtualização. Acima do *Resource Set*, são definidas a infraestrutura básica (*BIS* - *Basic Infrastructure Services*) de processamento, armazenamento e rede. Em *IaaS*, encaixam-se: *datacenters* e grades computacionais.



**Figura 1 – Pilha Arquitetural para Computação nas Nuvens [12].**

Os ambientes de programação (*Programming Enviroments*) e de execução (*Execution Enviroments*) para aplicações na nuvem estão no nível de *PaaS - Platform as a Service* -. Ambientes de execução tipicamente englobam os de programação, e podem utilizar diferentes ambientes de programação desde que haja compatibilidade entre os dois. *Google App Engine* suporta o *framerwork Django*, Java e Python [12].

As aplicações que executam sobre as nuvens, efetivamente, encontram-se na camada de *SaaS - Software as a Service* -. Observa-se que os desenvolvedores ou usam da camada de *PaaS* ou acessam diretamente a *IaaS* para construir e executar suas aplicações. É dividida em *Basic Application Services*, onde estão os serviços básicos (e.g. identificação), e em *Composite Application Services*, que são os serviços constituídos de outros, básicos inclusive (e.g. redes sociais). Uma aplicação na nuvem pode ser composta de um ou mais serviços.

No nível mais alto da pilha encontra-se o *HuaaS - Human as a Service* -. Nele estão os serviços agregados de informação (*IAS - Information Aggregation Services*) gerados a partir do uso coletivo ou em massa de uma aplicação, e que representam a opinião popular e promovem *marketing*. Tais aplicações se tornam ferramentas de trabalho para grupos de pessoas, e são denominadas de *CrowdSourcing* (e.g: *YouTube*).

Serviços de administração (*Administration*) e de suporte ao negócio (*Business support*), e.g. medição, monitoramento e tarifação, devem oferecer, aos fornecedores e clientes, visão coordenada e completa de suas atribuições quando atuam sobre as aplicações. Logo, estes precisam ter acesso a todas as camadas da pilha.

É nesse acúmulo de tecnologias que surge um complexo ecossistema composto de clientes e fornecedores de serviços na nuvem, onde aplicações através da *Web* tornam-se viavelmente escaláveis, fazendo uso de uma poderosa infraestrutura computacional de forma transparente, que é racionalmente utilizada promovendo redução de custos para quem as provê. Junta-se a necessidade de se ter acesso aos dados a partir de qualquer lugar através da Internet, e logo tem-se uma atmosfera propícia para disseminação do software como serviço.

### 3 DIFERENTES VISÕES DE SAAS

Qualquer menção a *SaaS*, deve-se considerar qual ponto de vista está se observando. Para um investidor, provedor de serviços ou até mesmo uma empresa de desenvolvimento, *SaaS* pode ser um modelo de negócio. Para um cliente em potencial, *SaaS* pode ser um modelo de entrega. Para um desenvolvedor, arquiteto ou gerente de projetos, *SaaS* pode ser visto como uma Arquitetura Orientada a Serviços (SOA), um web services. Ainda, para gerentes financeiros *SaaS* pode ser encarado como uma *commodity*.

Embora essas diferentes visões aparentem não ser relacionadas, há influências mútuas entre elas. Para o modelo de entrega proposto por *SaaS*, diversos desafios surgem para o desenvolvimento. Ainda, para o modelo de negócio com pagamento sob-demanda, são vários os interesses sistêmicos (tarifação, medições) e arquiteturais que precisam ser implementadas. A visão financeira do software como *commodity* implica em esforços de padronizações e arquiteturas de referência, além de consequências diante de indisponibilidade e falhas em serviços. Em síntese, para se obter um entendimento global do Software como Serviço, é importante considerar as diversas perspectivas em torno do tema e suas consequências para as demais.

#### 3.1 *SaaS* – Em busca de uma definição

Como uma funcionalidade a ser entregue ao cliente, Software como Serviço é um modelo de entrega de software. Considerando as implicações na forma como se comercializa e como se dá a cobrança em um *SaaS*, esse pode ser visto como um modelo de negócio. De uma maneira unificada, deve ser visto como um modelo de entrega de software que permite a exploração de um novo modelo de negócio. Bennet et. al. (2000), sintetiza bem essa visão em sua definição de Software como Serviço: "Software as a Service (*SaaS*) é um modelo de distribuição de software em que o fornecedor armazena a aplicação na Internet, sob sua própria infra-estrutura, e a disponibiliza via browser aos usuários que executam e armazenam seus trabalhos de maneira online, e cujos serviços oferecidos são tarifados sob demanda." [7]

#### 3.2 *SaaS* - Como modelo de entrega

A maneira tradicional de se comercializar software é tratando-o como um produto, *Software as a Product - SaaS*, da mesma forma como se faz com qualquer outro tipo de mercadoria. Quando o software oferecido como produto não satisfaz os requisitos de customização de um cliente, seja por regras de negócio, integração, ou requisitos não funcionais, uma opção é o software como projeto, em que uma equipe de desenvolvimento ou uma *software-house* é contratada e desenvolve todo o projeto do software. Estas duas visões do software, a primeira como uma mercadoria bastante padronizada, e a segunda como um produto altamente específico, guiaram todo o desenvolvimento de software nos últimos 50 anos.

Embora sejam abordagens diferentes para se obter a funcionalidade do software, ambas se assemelham ao considerarmos o cliente é quem fica com a posse do produto software. Na proposta de oferta de software como um serviço, a posse do produto software, passa para o controle operacional do provedor do software. Essa mudança de posse, mesmo parecendo insignificante, desencadeia novas formas de explorar a natureza intangível do software, a exemplo de diminuição da pirataria e *feedback* imediato entre cliente e provedor.

#### 3.3 *SaaS* - Como modelo de negócio

Na comercialização tradicional, o software é vendido como um produto, empacotado com manual, e o cliente paga antecipadamente por todas as funcionalidades disponíveis, mesmo por aquelas que nunca serão utilizadas. O cliente paga por ter a aplicação disponível 24 por 7, mesmo que a aplicação seja utilizada poucas horas por dia. Isto é compreensível, visto que o cliente adquiriu a licença (posse) do software, e o provedor não tem como acompanhar quanto o software está sendo utilizado e, conseqüentemente, quanto valor está agregando para o negócio do cliente.

Já o software oferecido como serviço viabiliza um novo mecanismo de cobrança (*pricing*), pagamento sob-demanda (pay-as-you-go). Nesta abordagem, clientes pagam apenas pelo o que utilizam. Aproxima-se o preço do software ao valor que suas funcionalidades fornecem ao cliente, logo, é uma maneira mais justa de cobrar do cliente. Com a posse do software e os custos de mantê-lo funcionando a cargo do provedor, fica evidenciado para o cliente o caráter intangível do software que, em essência, é um serviço e não um produto.

Com a mudança na posse do software o cliente fica a mercê do provedor para utilizar a funcionalidade. Para oferecer garantias ao cliente a respeito do fornecimento do serviço é feito um Acordo de Nível de Serviço

(SLA), um contrato em que ambas as partes formalizam a qualidade do serviço a ser prestado. SLAs refletem as preocupações dos clientes com segurança, privacidade, performance e disponibilidade, e representam um elemento fundamental na escolha de um provedor de *SaaS*. Para [8], SLAs são um ponto vital para a ampla adoção de *SaaS* em escala comercial.

Diversas são as maneiras de cobrar um serviço sob-demanda. Pagamento de uma taxa periódica (e.g.: anual) fixa (assinatura), quantidade de transações ou de usuários que utilizam o serviço (*seats*) [15]. É possível oferecer diferentes opções para que o cliente escolha a que melhor se adéqua às suas necessidades. Ainda, a captação de receita pode se dar trocando a gratuidade do serviço pela veiculação de propagandas.

Os provedores de *SaaS* escolheram como público alvo os pequenos e médios negócios (SMB), por se tratar de um segmento do mercado excluídos dos benefícios da informatização, dado os altos preços da TI. E, com a popularização da web 2.0 e a proliferação do marketing on-line, o mercado de consumidores não corporativos, domésticos, também passou a ser visado. Os provedores então, definiram modelos de negócio diferentes para cada um desses dois segmentos. Ambos são mercados não atrativos aos grandes *players* da indústria de software que dominam o modelo tradicional, e representam a cauda longa no mercado.

A justificativa para a viabilidade de se voltar a tal público alvo está no fenômeno conhecido como *long tail* [3], que tem como exemplo de sucesso a livreria virtual da Amazon. A figura 2 ilustra o fenômeno no contexto do mercado de *SaaS*, onde existe uma quantidade considerável de clientes que não podem arcar com o custo para adquirir o software. Representam um mercado não rentável para o modelo tradicional de comercialização de software, apesar de estarem dispostos a pagar menos do que o preço definido. A figura 3 apresenta o mercado potencial de *SaaS*, que engloba também os aqueles clientes presentes na cauda longa, pois há uma diminuição no custo para fornecer o software e, conseqüentemente, diminuição no preço final ao cliente. Em se tratando do mercado de *SaaS*, esse preço final deve ser considerado como o custo total de propriedade (TCO) e não apenas o custo de aquisição do software em si [5]. A exploração da cauda longa no mercado de software está intimamente relacionada com a economia de escala onde para um único produto/serviço, existem centenas de milhares ou até mesmo milhões de potenciais clientes. [16]

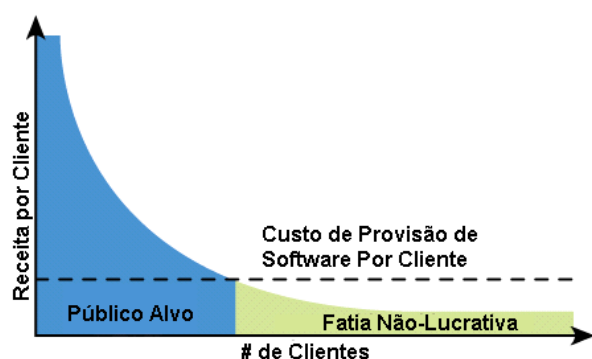


Figura 2. Fenômeno da *long tail* para o mercado de *SaaS*. Adaptada de [1]

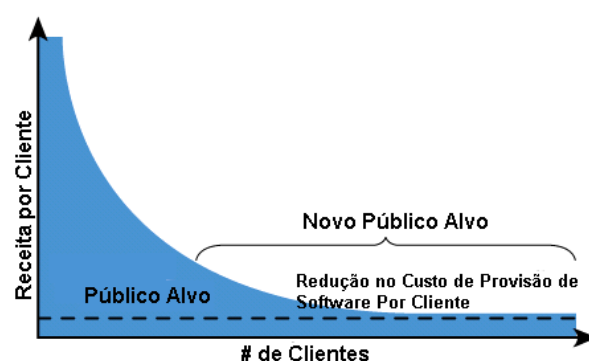


Figura 3. Exploração da *long tail* para o mercado de *SaaS*. Adaptada de [1]

É de se esperar que a exploração desse novo mercado acarrete em implicações na maneira como se entrega, se negocia e se desenvolve software. A Salesforce, uma das primeiras a explorar o *SaaS* para SMB, prega em suas campanhas de marketing o fim do software. Com a web 2.0 e a oferta gratuita de suítes de escritório e ferramentas de produtividade pessoal, o conceito de *SaaS* chegou aos consumidores domésticos. Logo, cabe verificar quais são as características dos *SaaS* que favorecem ou dificultam essa inevitável mudança.

### 3.4 Características de *SaaS*

Em função da maneira como são entregues pelos fornecedores aos clientes e de suas formas associadas de comercialização e negociação, as aplicações *SaaS* podem ser identificadas e diferenciadas de outros tipos de aplicações através das seguintes características preponderantes:

- Entrega e Acesso via Web – O mundo *SaaS* inexistiria sem a Web. As aplicações *SaaS* são projetadas para serem hospedadas em provedores, e globalmente acessíveis através dos *thin-clients* (*browsers*). Sua distribuição se constitui em possibilitar que o cliente faça uso de suas funcionalidades, sem deter a posse da aplicação como no modelo tradicional, que agora fica com o fornecedor;

- Pagamento sob-demanda – É a característica que atende ao novo modelo de negócio. O fornecedor cobra do cliente por utilização dos serviços, e não pela licença do produto (modelo tradicional);
- Gerência via Web – Possibilita que o cliente de um *SaaS* possa ter acesso as estatísticas de uso dos serviços, bem como o potencial para customizar a aplicação para atender as suas necessidades;
- Gerenciamento Centralizado - Permite que o provedor de *SaaS* tenha uma única aplicação para gerenciar para cada cliente. Para tanto, a customização deve ser feita pelos próprios inquilinos, seguindo o paradigma *self-service*, reduzindo os custos de gerenciamento por parte do provedor;
- Medição - Capacidade de rastrear e quantificar o uso de recursos e/ou de funcionalidades. Serve para tarifar clientes no modelo *pay-as-you-go*, para monitorar qualidade dos serviços, e para identificar as funcionalidades mais utilizadas;
- Composabilidade - É a possibilidade de criar novos serviços a partir de serviços já existentes, seja agregando ou integrando serviços;
- Desenvolvimento e Atualização Contínuos - Permite que continuamente os *SaaS* tenham disponível para uso, assim que concluídas, novas funcionalidades e correções. Para não prejudicar clientes de versões anteriores, as versões atualizadas dos *SaaS* devem garantir compatibilidade;
- Plataforma Multi-inquilino – É a característica mais relevante e inovadora, que diferencia *SaaS* de aplicações Web [11]. Uma plataforma multi-inquilino utiliza recursos comuns e uma única instância, tanto da aplicação como do banco de dados, para suportar diversos clientes simultaneamente, de forma semelhante a clientes que teriam recursos dedicados a apenas a sua aplicação individual [10]. Essa característica potencializa a customização do *SaaS* em função das necessidades de cada cliente;
- Escalabilidade – Remete a capacidade de *SaaS* de incrementar a quantidade de clientes atendidos, sem comprometer o nível da qualidade de serviços prestados (quantidade de funcionalidades, tempo de resposta) a todos os demais, inclusive aos já em atendimento;
- Online Marketing - Capacidade de divulgação automática do serviço através das plataformas online, a exemplo de redes sociais, blogs ou catálogos de serviços.

A maioria dessas características, dependendo do porte da aplicação *SaaS* e dos requisitos adicionais, pode implicar na necessidade de outras características: Multi-inquilino implica em confiabilidade e segurança, já que plataformas multi-inquilino requerem que os dados dos clientes sejam mantidos de forma confiável (consistentes), e seguros contra acessos indevidos; Acesso via *Web* implica em disponibilidade, pois mesmo com acesso irrestrito a *Web*, caso a aplicação *SaaS* falhe nenhum dado do cliente poderá ser acessado.

## 4 BENEFÍCIOS E DESAFIOS PARA ADOÇÃO DE SAAS

Integrando as visões de *SaaS* (modelo de entrega e de negócio) e compreendendo as suas características, é possível identificar vantagens na adoção de *SaaS* tanto para fornecedores quanto para clientes:

### 4.1 Para Fornecedores

Redução dos Custos - Com arquiteturas mais flexíveis, compatíveis e escaláveis, os fornecedores diluem seus custos de prover o serviço em função da quantidade de clientes atendidos;

Evita Pirataria – No desenvolvimento no modo *SaaS*, todos os serviços incluem autenticação de usuário de maneira centralizada, já que estão do lado do provedor do *SaaS*, diferente do modelo tradicional, onde a cópia dos softwares é difícil de ser evitada [6]. Agora, a posse do software é do provedor e não do cliente;

Compreensão Imediata de Requisitos - Como os clientes usam o serviço provido pelo *SaaS* em tempo real, é possível saber quais sub-partes são mais utilizadas. Isso ajuda no refinamento e na expansão das aplicações.

- Planejamento do Negócio de TI - Ao invés de receber um montante financeiro inicial como no modelo tradicional, provedores têm receitas parceladas, contínuas e previsíveis ao longo tempo [9];

Flexibilidade na Comercialização - *SaaS* é adequado tanto para negociação de software: como "*commodity*", individualizado ou mídia de veiculação de propaganda; por uso, por contratos ou por sequência de serviços.

- Potencializa a Expansão de Mercados e Oportunidades - A venda da funcionalidade do *SaaS* é feita pela Web para qualquer consumidor independente da plataforma ou linguagem, alcançando consumidores antes desprezados (*long tail*). E, permite que empresas antes produtoras de *SaaS* com alcance de mercado restrito, sejam capazes de entrar em mercados de alcance globalizado [8].



## 4.2 Para Clientes

Redução do Custo de Informatização – São os provedores de *SaaS*, agora, os responsáveis por toda a implementação inicial (infra-estrutura de rede, *software*, *hardware* e plataformas operacionais) e boa parte da manutenção. As empresas consumidoras de *SaaS* não têm que fazer um alto investimento inicial, nem se preocupar com custo de manutenção e treinamento da equipe, e processos de informatização [13]:

- Paga-se pelo que se Usa - No cenário de *SaaS* paga-se pelo software completo mesmo que se use apenas parte dele. Em *SaaS*, é pode-se pagar apenas por sub-serviços utilizados num período;
- Competitividade entre Provedores de *SaaS* – Para terem suas necessidades atendidas, os clientes de *SaaS* podem optar por fornecedores que ofereçam melhores: preço, tratamento, funcionalidade e escalabilidade do serviço. Se insatisfeito, o cliente pode mudar de fornecedor, dado o pagamento por demanda [8], e por não estar preso a investimentos iniciais de TI de um fornecedor específico;
- Melhoria da Qualidade dos Serviços: Com a competitividade, e para manter e expandir sua base de clientes, os fornecedores buscam melhoramentos: aumentam o ritmo de inserção de novas funcionalidades e de correções de seus serviços no mercado, aprimoram as relações com clientes [9];
- Possibilita Maior Foco no *Core* do Negócio – Como consumidoras de *SaaS*, as empresas mantêm o foco dos seus investimentos no seu negócio, já que contratam os serviços de TI de fornecedores.

*SaaS* representa um mercado crescente e promissor como revelam pesquisas sobre o nicho de TI: cerca de 80% das empresas hoje cogitam em fazer uso de algum serviço por demanda e até 2012 *SaaS* representará 25% do mercado de software corporativo com receita estimada de 14.8 bilhões de dólares [2]. No entanto, *SaaS* apresenta, para fornecedores e clientes, fatores relacionados à segurança, personalização e interoperabilidade que inibem ou desencorajam a sua adoção, constituindo desafios a serem suplantados para tornar realidade as suas estimativas.

Quanto à segurança, há a necessidade de prover um ambiente seguro e confiável, visto que os dados dos clientes ficam no provedor e cada cliente deve ter acesso apenas aos seus dados. Caso a aplicação falhe (defeito, *trojans*, desastres, sabotagem) é o provedor que deverá garantir que os dados não sejam danificados. No cenário atual, os clientes em potencial ainda não demonstram confiança suficiente de que seus dados não sofrerão danos, e, conforme citado em [2], trata-se de um dos obstáculos à adoção de *SaaS*.

Ao lado da segurança, a falta de interoperabilidade é uma das preocupação mais citadas nas pesquisas realizadas com futuros clientes. O temor de ficar preso (*lock-in*) a um único provedor afasta potenciais usuários e para mudar esse quadro os provedores devem possibilitar a migração de dados e aplicações entre si. Embora não se tenha algo parecido no fauna atual de provedores, um suporte dessa natureza é primordial para promover a competitividade entre provedores e a contínua melhoria dos serviços, de modo a quebrar essa resistência dos usuários. Ainda na linha da interoperabilidade, a Composição Dinâmica de Serviços é outro desafio, visto que um *SaaS* deve ter a capacidade de oferecer e consumir serviços de outros *SaaS*, mesmo estando em provedores diferentes Logo, padrões de composição e de integração dinâmica precisam ser definidos e adotados entre provedores.

No tocante à personalização se faz necessária a provisão de mecanismos de configuração para atender a característica multi-inquilino de forma que o *SaaS* deve ser implantado no quarto nível de maturidade, o mais alto [1]. Tais mecanismos devem atender às especificidades de clientes distintos, mas que possuem requisitos funcionais similares, de forma autônoma, isto é, sem intervenção humana por parte do provedor [6], para permitir a escalabilidade necessária para atingir um mercado de larga escala ("*long tail*").

O aumento do leque de funcionalidades de um *SaaS*, permite atrair novos clientes que antes não utilizavam o *SaaS* por não terem seus requisitos atendidos. Novas versões do *SaaS*, quando postos em operação, devem garantir compatibilidade com versões anteriores, e garantir que nenhum dos clientes seja prejudicado (sofra perda de dados ou redução de funcionalidade). Nesse cenário, evolução e implantação contínua são imprescindíveis, o que obriga o fornecedor a trabalhar em sintonia com as práticas mais avançadas de Engenharia de Software [7].

## 5 CONCLUSÕES

O acúmulo de tecnologias de TI ao longo das últimas décadas causa inconsistências no entendimento das definições de seus conceitos e modelos, tanto entre diferentes *players* da indústria quanto entre esses e a academia. A expansão da Internet e o surgimento da computação nas nuvens mudaram a forma de se encarar

a área de TI, trazendo consigo a possibilidade da oferta do software como serviço, *SaaS*. Agora, o cliente adquire o direito de fazer uso das funcionalidades do software e a sua posse continua com o fornecedor.

Trata-se de uma importante mudança de paradigma, que nos obriga a ter que entendê-lo por diferentes perspectivas: como modelo de entrega, de negócio, e com novas características e requisitos. Com o objetivo de compreender *SaaS* de uma maneira plural, o presente trabalho não foca em uma perspectiva em particular, mas apresenta suas inter-relações e evidencia como *SaaS* potencializa a expansão do mercado de software através de um novo modelo de entrega e de negócio, permitindo que aqueles que antes se encontravam à margem possam usufruir dos recursos de TI. Também se apresenta uma definição precisa do que é *SaaS*.

Muitos desafios ainda precisam ser superados para *SaaS* se tornar uma unanimidade, embora pesquisas indiquem sua expansão como algo certo e que representará um quarto do mercado de software nos próximos anos. No entanto, apenas com uma visão pluralista, é que se será capaz de fazer convergir os objetivos da indústria consumidora e fornecedora de software, com os da academia em prol da disseminação de *SaaS*.

## REFERÊNCIAS

- [1] \_\_\_\_\_. **Architecture Strategies for Catching the Long Tail.** . <http://msdn.microsoft.com/pt-br/library/aa479069.aspx>. Acesso em: 18/05/2010.
- [2] \_\_\_\_\_. **Gartner, Inc Research.** <http://www.gartner.com/it/page.jsp?id=783212>. Acesso em 02/08/2009.
- [3] \_\_\_\_\_. **The Long Tail.** <http://www.wired.com/wired/archive/12.10/tail.html>. Acesso em: 10/05/2010.
- [4] \_\_\_\_\_. **Uma introdução ao Software + Serviços, SaaS e SOA.** <http://msdn.microsoft.com/pt-br/library/dd875466.aspx> Acesso em: 17/05/2010.
- [5] \_\_\_\_\_. **Software as a Service: Strategic Backgrounder.** Software & Information Industry Association, White Paper, 2001.
- [6] SAEED, M.; JAFFAR-UR-REHMAN, M., "Service Based Software Engineering: The Personalization Challenges,". IEEE INMIC 2005.
- [7] BENNETT, K.; LAYZELL, P.; BUDGEN, D.; BRERETON, P.; MACAULAY, L.; MUNRO, M. **Service-Based Software The Future for Flexible Software.** APSEC 2000.
- [8] BUYYA, R., YEO C. S, VENUGOPAL S. "Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities". HICSS'07, 2007. Hawaii, USA.
- [9] CHOUDHARY, V.. "Software as a Service: Implications for Investment in Software Development," HICSS'07, 2007. Hawaii, USA.
- [10] KWOK, T., NGUYEN, T.; LAM, L. **A Software as a Service with Multi-tenancy Support for an Electronic Contract Management Application.** IEEE ICSEC 2008, Washington, USA.
- [11] LAPLANTE, PHILLIP A.; ZHANG, J.; VOAS, J.. "What's in a Name? Distinguishing between SaaS and SOA," IT Professional, May/June 2008, doi:10.1109/MITP.2008.60.
- [12] LENK, A.; KLEMS, M.; NIMIS, J.; TAI, S.; SANDHOLM, T.. **What's Inside the Cloud? An Architectural Map of the Cloud Landscape.** ICSE'09 CLOUD'09, May 2009, Vancouver, Canada.
- [13] LIAO, HANCHENG; TAO, CHANGQI. **An Anatomy to SaaS Business Model Based on Internet. Management of e-Commerce and e-Government,** 2008. doi:10.1109/ICMECG.2008.16.
- [14] LUBLINSKY, B.. **Defining SOA as an architectural style - Align your business model with technology.** <http://www.ibm.com/developerworks/library/ar-soastyle/>. Acesso em 03 de setembro de 2009.
- [15] MOORE, B.; MAHMOUD, QUSAY H. "A service broker and business model for saas applications," pp.322-329, 2009 IEEE/ACS Intern. Conf. on Computer Systems and Applications, 2009.
- [16] SUN, W.; ZHANG, X., GUO, C. JIE; SUN, P.; SU, H.. "Software as a Service: Configuration and Customization Perspectives," pp.18-25, 2008 IEEE Congress on Services, 2008.
- [17] YOUSEFF, L.; BUTRICO, M.; DA SILVA, D., "Toward a Unified Ontology of Cloud Computing," Grid Computing Environments Workshop, 2008. GCE '08 , vol., no., pp.1-10, 12-16 Nov. 2008.