

DOI: 10.5748/9788599693100-11CONTECSI/PS-1008
AN ANALYSIS OF THE CLOUD COMPUTING APPROACH TO SOA

Tâmara Batista Reis (Universidade Salvador, Bahia, Brasil) - tamarabr38@hotmail.com
Paulo Caetano da Silva (Universidade Salvador, Bahia, Brasil) - paulo.caetano@pro.unifacs.br

Technological development and popularization of the Internet has provided increased opportunities for Web use since data storage until providing services. In this scenario the Service Oriented Architecture emerged as an architectural approach to service-oriented enterprise systems, whose focus is on service definition, service interfaces and behaviors. In turn, with the aim of providing services that can be since the services offered to the end users that hosts your personal documents on the Internet until companies that outsource the entire IT infrastructure to other companies, the proposal of Cloud Computing (Cloud Computing) becomes a new paradigm of Engineering Service-Oriented Systems. This paper presents an analysis of the proposed Cloud Computing and SOA, identifying the main characteristics of each one, the relationship between them and the different points.

Keywords: Service Oriented Architecture; Cloud Computing; IT infrastructure; database; management systems.

UMA ANÁLISE DA ABORDAGEM CLOUD COMPUTING EM RELAÇÃO A ARQUITETURA ORIENTADA A SERVIÇOS (SOA)

O desenvolvimento tecnológico e a popularização da internet vêm proporcionando o aumento de oportunidades de uso da *Web*, que vai desde o armazenamento de dados até o oferecimento de serviços. Neste cenário surge a Arquitetura Orientada a Serviço (*Service Oriented Architecture – SOA*), uma abordagem arquitetônica para sistemas empresariais orientados à oferta de serviços, cujo foco se concentra na definição de serviços, interfaces e comportamentos de serviços. Por sua vez, com o objetivo de prover serviços que podem ser desde os ofertados ao usuário final que hospeda seus documentos pessoais na Internet, até empresas que terceirizam toda infraestrutura de TI para outras empresas, a proposta da *Cloud Computing* (Computação em Nuvem) torna-se um novo paradigma das Engenharias de Sistemas Orientados a Serviços. Este trabalho traz uma análise das propostas Computação em Nuvem e da SOA, identificando as principais características de cada uma, a relação entre elas e os pontos distintos.

Palavras-chave: Arquitetura Orientada a Serviço; Computação em Nuvem; infraestrutura de TI; banco de dados; gerenciamento de sistemas.

1 Introdução

A partir do advento da Revolução Industrial, o planeta vem experimentando cada vez mais as mudanças proporcionadas pelo desenvolvimento tecnológico nas mais diversas áreas, seja nas comunicações, no transporte, na saúde, etc. Produtos e serviços jamais imaginados pelo homem surgem a uma velocidade surpreendente (Milagre & Cusin, 2009).

A despeito deste avanço, as tecnologias de transmissão da informação, sendo a principal delas a *Internet*, permitiram uma facilidade de comunicação tal, que o planeta entrou num contexto de globalização. Usando a metáfora da planificação do planeta, Thomas Friedman (2009) chega a comentar que a Terra não é mais redonda, mas achatada, ou seja, a facilidade em que o mundo hoje se comunica, deixa a impressão de que o mundo foi planejado. A informação, portanto, tornou-se, nos dias atuais, a mercadoria de grande valor e o conhecimento, a “moeda” do nosso tempo.

Sendo o maior conglomerado de redes de comunicações em escala mundial, ou seja, vários computadores e dispositivos conectados, a *Internet* dispõe de milhões de dispositivos interligados por protocolos de comunicação que permitem o acesso a informações e vários tipos de transferência de dados. Além disso, carrega uma ampla variedade de recursos e serviços, incluindo os documentos interligados por meio de hiperligações da *World Wide Web* (Rede de Alcance Mundial), e a infraestrutura para suportar correio eletrônico e serviços como comunicação instantânea e compartilhamento de arquivos.

Toda essa tecnologia abriu um verdadeiro leque de oportunidades de uso da *Web*, desde o simples armazenamento de dados de um usuário comum utilizando uma rede social até a troca de informações sigilosas entre Organizações Governamentais ou mesmo transações comerciais.

Neste cenário, surge a proposta da *Cloud Computing* (Computação em Nuvem) com o objetivo de prover serviços para as massas que vão desde o usuário final que hospeda seus documentos pessoais na Internet até empresas que terceirizam toda infraestrutura de TI para outras empresas.

Segundo o estudo apresentado por Sousa, Moreira, Macêdo e Machado (2011), a computação em nuvem surge da necessidade de construir infraestruturas de TI complexas, nas quais os usuários têm que realizar instalação, configuração e atualização de sistemas de software. Além disso, os usuários precisam lidar com questões da infraestrutura de TI, propensa a ficar obsoleta rapidamente. A utilização de plataformas computacionais de terceiros é a solução apresentada pela tecnologia da computação em nuvem, na qual os recursos de TI são fornecidos como um serviço, permitindo que os usuários o acessem sem a necessidade de conhecimento sobre a tecnologia utilizada. Desse modo, os usuários e as empresas passam a acessar os serviços sob demanda e independente de localização, conhecidos como *Web Services*.

O conceito de *Web Service* pode ser entendido como sendo o nome dado à tecnologia que permite a comunicação entre aplicações de uma maneira independente de sistema operacional e de linguagem de programação. Um *Web Service* é uma classe escrita em uma linguagem suportada pela plataforma.NET que pode ser acessada via protocolo http (Hadade, 2014).

Por sua vez, a visão geral da SOA é fornecida e implementada por meio de *Web Services*, um conjunto de padrões da indústria, e tem se tornado o predominante conjunto de ferramentas práticas usado por engenheiros das empresas para os atuais projetos de SOA.

A caracterização do problema abordado neste trabalho se apresenta pela constatação de que na medida em que a tecnologia da informação avança, apresentam-se os reveses do seu desenvolvimento com o aumento da complexidade dos sistemas computacionais, e.g. a necessidade de escalabilidade dos bancos de dados, a necessidade de novos hardwares, que por sua vez requerem novos softwares, tornando obsoleto o que muito recentemente era considerado novo. Outra limitação a ser considerada diz respeito à administração dos grandes sistemas, pois além de possuírem custo elevado, requerem a manutenção da própria aplicação e uma base de usuários própria com informações e níveis de privilégio para permitir acesso aos serviços, conforme visto em Moreira *et al.* (2011).

O objetivo principal deste trabalho está centrado na investigação das propostas da Computação em Nuvem e da SOA como resposta aos problemas de montagem, manutenção e gerenciamento de infraestrutura de TI, enfrentados pelos usuários de serviços *Web*.

Será feito também um levantamento das características dessas duas tecnologias, a fim de identificar suas semelhanças e diferenças, para que possam servir como orientação às empresas que pretendem iniciar atividades voltadas prestação de serviços via *internet*.

Sendo este trabalho de natureza exploratória, a metodologia utilizada teve como base a pesquisa aplicada, tendo sido realizada a busca em fontes da literatura especializada. Como embasamento à proposta do trabalho são apresentadas as análises de dois artigos base, um sobre Computação em Nuvem e outro sobre SOA, e artigos correlatos.

Este estudo está estruturado da seguinte forma: esta seção introdutória apresenta uma breve descrição do trabalho trazendo conceitos sobre a Computação em Nuvem e sobre a SOA. Contém também a identificação do problema, traça os objetivos a serem alcançados e traz as descrições da metodologia e da estrutura do trabalho. Na Seção 2 é abordado o paradigma da Computação em Nuvem, suas características essenciais, os modelos de serviço, implantação e limitações. Na Seção 3 é discutida a *Service-Oriented Architecture* (SOA), enfatizando suas principais características e utilização. Na Seção 4 é realizada uma análise comparativa entre o paradigma SOA e a Computação em Nuvem. Na Seção 5 são apresentados os resultados deste trabalho, colocando em destaque os desafios encontrados. A Seção 6 traz a conclusão do trabalho e, por fim, na Seção 7 são apresentadas as referências bibliográficas.

2 A Cloud Computing

As definições encontradas na literatura acadêmica sobre a *Cloud Computing* são muitas, portanto, esta seção apresentará alguns dos principais conceitos.

Segundo Mell e Grance (2009), o *National Institute of Standards and Technology* (NIST) define a computação em nuvem como um paradigma em evolução que apresenta um modelo de acesso sob demanda, a um conjunto de recursos computacionais

configuráveis (e.g. redes, servidores, armazenamento, aplicações, serviços). Estes recursos são rapidamente adquiridos e disponibilizados com mínimo esforço gerencial ou interação com o provedor de serviços.

Em Brantner, Florescu, Kossmann e Kraska, (2008), temos que a computação em nuvem é uma evolução dos serviços e produtos de tecnologia da informação sob demanda, também chamada de *Utility Computing*. Estes estudiosos apontam diversas vantagens aos usuários deste paradigma, dentre as quais podemos destacar: a solução a respeito do problema da escalabilidade, pois a capacidade de armazenamento fornecida é praticamente infinita; a disponibilidade dos serviços é outro destaque, pois a *Utility Computing* propõe fornecer disponibilidade total, em que os usuários podem ler e gravar dados a qualquer momento, sem sofrerem qualquer bloqueio; os tempos de resposta são quase constantes e não dependem de premissas tais como o número de usuários simultâneos, o tamanho do banco de dados ou qualquer parâmetro do sistema. Outra preocupação retirada dos usuários é a necessidade da realização de backups, pois em caso de falha dos componentes, o provedor é responsável por substituí-los e tornar os dados disponíveis em tempo hábil por meio de réplicas.

Vecchiola, Chu e Buyya (2009) defendem que a *Cloud Computing* tem como principal objetivo o fornecimento de componentes básicos como armazenamento, processamento e largura de banda de uma rede como uma “mercadoria” através de provedores especializados, com um baixo custo por unidade utilizada. Isto é baseado no oferecimento desses serviços sob demanda e com o pagamento conforme o uso, da mesma forma que serviços considerados de utilidade pública a exemplo do fornecimento de água, energia elétrica, gás, etc.

2.1 A infraestrutura da *Cloud Computing*

Conforme aduzem Buyya, Yeo, Venugopal, Broberg, e Brandic (2009), a expressão "Computação em Nuvem" é uma metáfora para a *Internet* ou infraestrutura de comunicação entre os componentes arquiteturais, capaz de abstrair e ocultar a complexidade da infraestrutura. Cada uma de suas partes é provida como um serviço, usualmente alocados em centros de dados através da utilização de hardware compartilhado para computação e armazenamento. A Figura 1 apresenta o modelo da infraestrutura do ambiente de Computação em Nuvem.

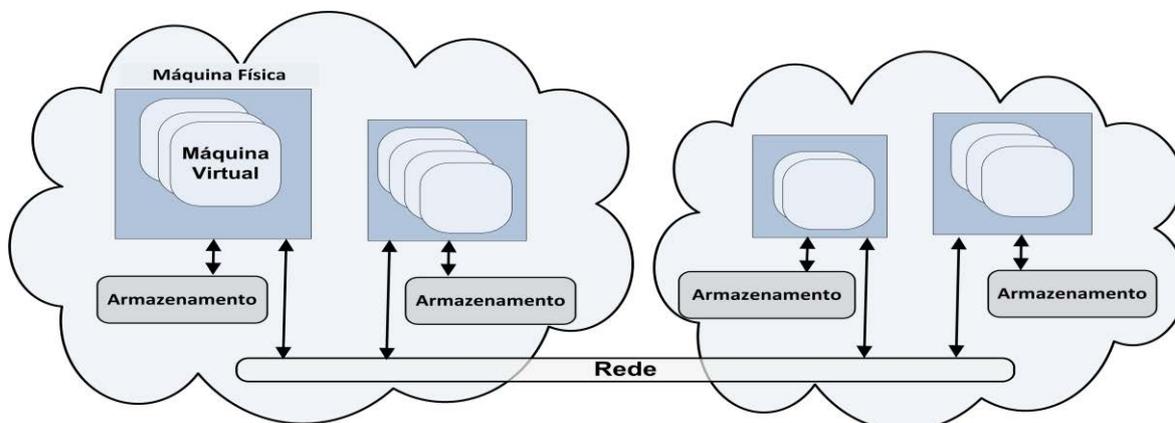


Figura 1. Ambiente de *Cloud Computing* (Sousa et al., 2011)

Conforme se pode observar na Figura 1, a arquitetura da *Cloud Computing*, em geral, é composta por centenas (ou até milhares) de máquinas físicas ou nós físicos de baixo custo, conectadas por meio de uma rede. Estas máquinas físicas podem variar na capacidade de *hardware* em termos de CPU, memória e armazenamento em disco, porém, devem possuir as mesmas configurações de *softwares*. Em cada máquina física existe um determinado número de máquinas virtuais (VM) ou nós virtuais em execução, a depender da capacidade do *hardware* disponível. Os dados são persistidos, geralmente, em sistemas de armazenamento distribuídos (Soror *et al.*, 2010).

2.2 A composição da *Cloud Computing*

Segundo o NIST (*National Institute of Standards and Technology*), a configuração da *Cloud Computing* é composta por cinco características essenciais, três modelos de serviço e quatro modelos de implantação, que serão abordados nesta seção (Mell & Grance, 2009).

2.2.1 Principais características

As características essenciais que devem compor a *Cloud Computing* estão diretamente relacionadas com as vantagens que este tipo de solução oferece, pois o conjunto delas define a computação em nuvem, fazendo a distinção com outros paradigmas, conforme se segue:

- ✓ **Amplio acesso:** Os recursos são disponibilizados por meio da rede e mecanismos padronizados que possibilitam o acesso a estes recursos através do uso por plataformas do tipo *thin*, tais como celulares, *laptops* e PDAs.
- ✓ **Serviços sob demanda:** O usuário pode adquirir unilateralmente o recurso computacional (e.g. armazenamento, tempo de processamento no servidor) de acordo com as suas necessidades, sem haver a necessidade de interação humana com os provedores dos serviços.
- ✓ **Elasticidade rápida:** Os recursos podem ser adquiridos ou liberados de forma rápida e elástica, a depender da necessidade de escalar com o aumento da demanda, ou liberar na retração da mesma.
- ✓ **Serviço medido:** O uso de recursos pode ser monitorado e controlado, favorecendo a transparência tanto para o provedor como para o usuário, pois os sistemas em nuvem permitem o controle e a otimização do uso de recursos por meio de uma capacidade de medição. A automação é realizada no nível de abstração apropriado para cada tipo de serviço: armazenamento, processamento, largura de banda e contas de usuário ativas.
- ✓ **Pooling de recursos:** Os provedores são capacitados a servir múltiplos usuários usando um modelo multi-inquilino (*multi-tenant*), pois os recursos computacionais são organizados em um *pool*, com diferentes recursos físicos e virtuais, dinamicamente atribuídos e ajustados de acordo com a demanda dos usuários.

2.2.2 Modelos de Serviços

No que diz respeito a modelos de serviços, o ambiente da *Cloud Computing* é composto de três modelos distintos, que definem um padrão arquitetural para soluções de Computação em Nuvem.

- ✓ **Software as a Service (SaaS):** Este modelo propõe *softwares* como serviço, apresentando sistemas com propósitos específicos, disponíveis por meio da

Internet e acessíveis a partir de vários dispositivos por meio de uma *interface thin client* como um navegador *Web*. A administração ou o controle da infraestrutura subjacente são abstraídos ao usuário, incluindo rede, servidores, sistema operacional, armazenamento, com exceção das configurações específicas. Exemplos: serviços de *Customer Relationship Management* (CRM) da *Salesforce* e o *Google Docs*.

- ✓ **Plataform as a Service (PaaS):** Neste modelo plataformas são oferecidas como serviço, o que pode compreender sistema operacional, linguagens de programação e ambientes de desenvolvimento para as aplicações. O usuário pode controlar as aplicações implantadas e as configurações de aplicações hospedadas, porém não administra ou controla a infraestrutura subjacente. Exemplos: *Google App Engine* e *Microsoft Azure* (Azure, 2011).
- ✓ **Infrastructure as a Service (IaaS):** Esta proposta torna mais acessível o fornecimento de recursos de computação fundamentais para construir um ambiente de aplicação sob demanda, tais como servidores, rede, armazenamento, sistemas operacionais e aplicativos. Usualmente, o usuário mantém controle sobre os sistemas operacionais, armazenamento, aplicativos implantados e pode selecionar componentes de rede, mas não administra ou controla a infraestrutura da nuvem. Exemplos: O *Eucalyptus* (Liu, Liang, & Brooks, 2007) e o *Amazon Elastic Cloud Computing* (EC2) (Robinson, 2008).

2.2.3 Modelos de Implantação

Os modelos de implantação para os ambientes de Computação em Nuvem se diferem principalmente quanto às restrições ou permissões de acesso e a disponibilidade. Tais premissas irão depender do processo de negócios e do tipo de informação desejado.

- ✓ **Nuvem Pública:** neste modelo a infraestrutura de nuvem é disponibilizada para o público, podendo ser acessada por qualquer usuário que conheça a localização do serviço.
- ✓ **Nuvem Privada:** a infraestrutura de nuvem é utilizada e administrada exclusivamente por uma organização (podendo ser a administrada por terceiros), sendo uma nuvem local ou remota.
- ✓ **Nuvem Comunidade:** fornece uma infraestrutura que pode ser compartilhada por uma comunidade de organizações com interesses em comum.
- ✓ **Nuvem Híbrida:** a infraestrutura traz uma composição de duas ou mais nuvens, que não obstante a este fato, continuam a serem entidades únicas, conectadas por meio de tecnologia proprietária ou padronizada, a fim de possibilitar a portabilidade de dados e aplicações.

2.3 Modelo Multi-Inquilino

O termo multi-inquilino se refere ao uso do mesmo software e instâncias por vários usuários e empresas de forma simultânea, pois o objetivo dessa abordagem é disponibilizar os mesmos recursos de software para um número maior de usuários.

Segundo a Microsoft (2010), esta abordagem é baseada no conceito da “Cauda Longa”.

Este modelo está ilustrado na Figura 2.

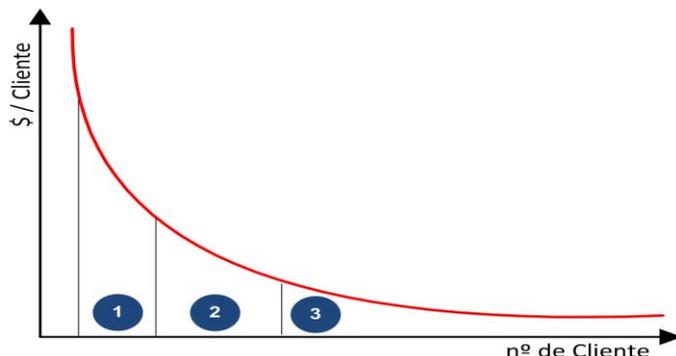


Figura 2. Gráfico da Calda Longa (Microsoft, 2010)

Esta abordagem é utilizada por diversas empresas de *Internet* que conseguem obter renda tanto com produtos de nicho como produtos tradicionais. A primeira parte do gráfico refere-se a grandes clientes, a segunda parte a clientes típicos e a terceira parte mostra uma grande quantidade de pequenos clientes, devido à diminuição dos custos. Observa-se que, conforme se diminui o custo de adoção, um número maior de clientes pode adotar uma solução, sendo que este número tende ao infinito, uma vez que a curva não toca o eixo “x”.

Outro conceito em destaque do modelo é o “micro-pagamento”. Na cauda longa, um número muito grande de usuários poderá adotar esta solução pagando pelo uso, por demanda, o que deve gerar um valor muito baixo de pagamento. A fim de permitir que um provedor ou desenvolvedor ofereça a mesma instância de um aplicativo SaaS para vários inquilinos, a aplicação deve ser multi-inquilino consistente, ou seja, cada inquilino pode interagir com o aplicativo como se ele fosse o único usuário da aplicação, mas não pode acessar ou visualizar os dados de um outro inquilino.

A maturidade do SaaS é definida de acordo com o nível de implementação do conceito de multi-inquilino, conforme pode ser observado na Figura 3.

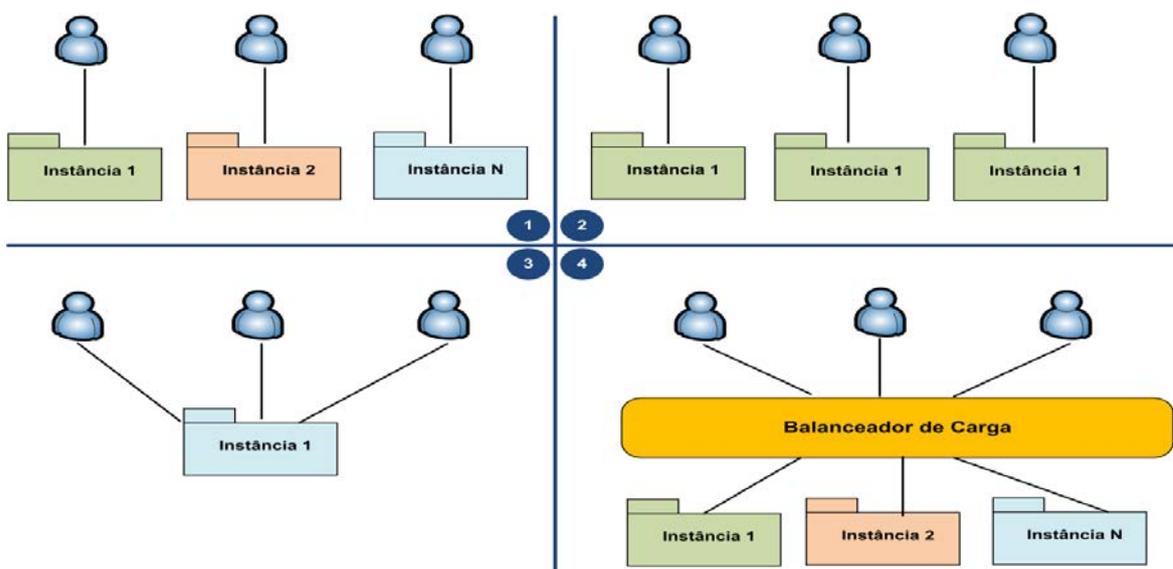


Figura 3. Modelo de maturidade SaaS, sobre a evolução do suporte multi-inquilino (Microsoft, 2010).

Conforme mostrado na Figura 3, no primeiro estágio há um software para cada inquilino (*ad-hoc*/personalizado), o que garante o atendimento das demandas do usuário e custo elevado devido a grande customização e disponibilidade de recursos individuais. O segundo estágio apresenta uma única solução configurável e eficiente destinada para todos os inquilinos separadamente, sem nenhuma customização, o que garante menor custo de manutenção. No terceiro estágio existe uma única solução multi-inquilino disponível para todos os inquilinos, onde ocorre total compartilhamento de recursos. No quarto estágio o atendimento é diferenciado para inquilinos que exigem elevada demanda de recursos, havendo uma carga balanceada na infraestrutura do provedor da solução SaaS.

2.5 Sistemas de Gerenciamento de Dados (SGBDs) em Nuvem

Em estudo apresentado por Curino, Jones, Zhang, Wu, & Madden (2010), podemos observar que a infraestrutura de SGBDs em nuvem possui diversas vantagens para os usuários, dentre as quais se destacam:

- ✓ Elasticidade;
- ✓ Escalabilidade;
- ✓ Uso de *interfaces* de acesso unificado, delegação de *tuning* e administração de SGBDs para reduzir a complexidade técnica;
- ✓ Previsibilidade e custos proporcionais (mais baixos) à qualidade do serviço (QoS) e cargas de trabalho reais;

Abouzeid, Bajda-Pawlikowski, Abadi, Rasin e Silberschatz (2009) advertem que apesar de diversos sistemas e arquiteturas estarem sendo desenvolvidos para suprir as novas demandas de aplicações com diferentes requisitos de processamento e armazenamento, existe o problema de provisionar recursos, que no ambiente em nuvem torna-se um problema de otimização, pois existe uma grande quantidade de usuários, múltiplos SGBDs e grandes centros de dados. Isso favorece a exploração da economia em escala, o balanceamento dinâmico de carga e o gerenciamento de energia. Como proposta de resolução do problema da escolha, implantação e soluções de administração para a gestão de dados, os SGBDs em nuvem são disponibilizados como serviços, que encapsulam a complexidade do gerenciamento por meio de formas de acesso simples e garantias de acordos de nível de serviço (SLAs).

2.5.1 Requisitos para o Gerenciamento de Dados em Nuvem

Curino *et al.* (2010) destacam a importância do gerenciamento de dados como um serviço e apresentam uma lista de requisitos de um SGBD como um serviço, sob a perspectiva do usuário, do provedor e apresenta requisitos adicionais relacionados à nuvem pública conforme se segue:

- ✓ **Requisitos do Usuário**
 - U1: API simples com pouca configuração e administração (ex. sem *tuning*)
 - U2: Alto desempenho (ex. vazão, escalabilidade)
 - U3: Alta disponibilidade e confiança (ex. *hot stand-by*, *backup*)
 - U4: Acesso fácil às características avançadas (e.g. *snapshot*, evolução de esquema, mineração de dados)
- ✓ **Requisitos do Provedor**
 - P1: Atender o SLA do usuário (e.g. potencialmente sob carga de trabalho dinâmica)
 - P2: Limitar *hardware* e custo de energia (e.g. multiplexação intensiva)

- P3: Limitar custo de administração (e.g. custo com pessoal)
- ✓ **Requisitos extra de Nuvem Pública**
 - P1: Esquema de preço: barato, previsível e proporcional ao uso (elasticidade)
 - P2: Garantias de segurança e privacidade
 - P3: Baixa latência (relevante para OLTP e aplicações *Web*)

Independente do tamanho da base de dados, o usuário deseja um desempenho satisfatório, expresso em termos de latência e vazão. Recursos como alta disponibilidade e as características avançadas de gerenciamento do banco de dados, tais como *snapshot*, evolução de esquema e mineração de dados devem estar prontamente disponíveis e simples de utilizar.

Já o provedor necessita atender aos acordos de nível de serviço, independentemente da quantidade de dados e alterações na carga de trabalho. O modelo de serviço proporciona a eficiência do sistema na utilização do *hardware*, por multiplexação de cargas de trabalho e ajuste dinâmico da alocação de recursos. Ferramentas sofisticadas de análise de carga de trabalho e centralização do gerenciamento de muitos bancos de dados são utilizadas para minimizar a quantidade de tarefas de administração.

Para provedores de serviços em nuvem pública, existem requisitos adicionais, tais como esquema de preço, segurança, privacidade e latência.

2.5.2 Banco de Dados como um Serviço (DaaS)

Em Agrawal, Abbadi, Emekci, e Metwally (2009), vemos que os DaaS estão surgindo como um novo paradigma para a gestão de dados em ambientes corporativos, em que um prestador hospeda um banco de dados e o fornece como um serviço, no qual o gerenciamento dos aplicativos de banco de dados e dos custos são transferidos do usuário para o provedor de serviços.

Por sua vez, o usuário utiliza o serviço de dados por meio de funcionalidades como a configuração das bases de dados, os esquemas, a carga de dados no serviço de dados e as interfaces padronizadas de interação com a base. A Figura 4 ilustra esta organização.

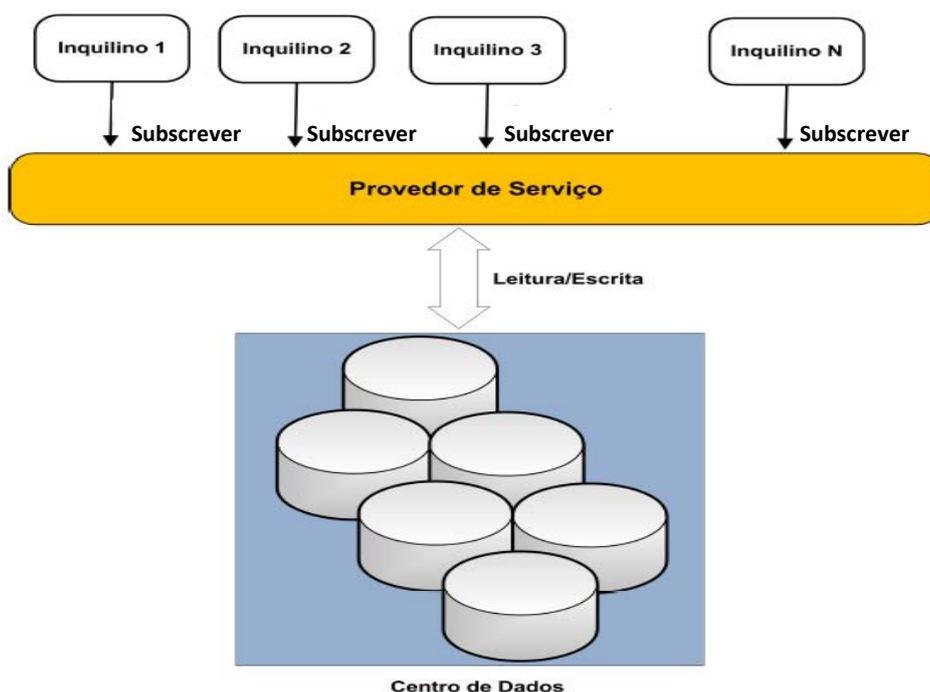


Figura 4. Banco de Dados como serviço (Sousa *et al.*, 2011)

Conforme ilustrado na Figura 4, cada inquilino contrata o serviço fornecido pelo provedor que por sua vez mantém um conjunto de banco de dados hospedados em centros de dados e deve garantir aspectos de disponibilidade, desempenho e qualidade do serviço definidos em SLA.

Dentre os benefícios que essa organização oferece aos usuários, destacam-se: um ambiente altamente escalável, disponível e rápido, a não preocupação de manter a infraestrutura de *hardware* e *software* para fornecer o serviço de dados e a garantia QoS requerida pelo usuário.

Aulbach, Grust, Jacobs, Kemper e Rittinger (2008) pontuam que um sistema de banco de dados multi-inquilino deve oferecer esquemas que sejam flexíveis em dois aspectos: a) deve ser possível estender o esquema base para suportar múltiplas versões do aplicativo; b) deve ser possível evoluir de forma dinâmica o esquema base e suas extensões, enquanto o banco de dados está em execução. A evolução de uma extensão deve ficar a cargo do usuário, o provedor de serviço não deve ser envolvido, para que os custos operacionais não sejam muito altos. Um sistema de banco de dados multi-inquilino pode compartilhar máquinas, processos e tabelas.

2.5.3 Tipos de Banco de Dados multi-inquilino

Existem vários modelos de compartilhamento do tipo multi-inquilino que se diferem no compartilhamento de diferentes recursos.

Para melhorar a compreensão destes modelos, Reinwald (2010) propõe a classificação apresentada na Tabela 1.

Tabela 1. Modelos de banco de dados multi-inquilino e correspondência com a Computação em Nuvem [Reinwald, 2010]

Modo de Compartilhamento	Isolamento	IaaS	PaaS	SaaS
1. Hardware	VM	X		
2. Máquina Virtual	Usuário SO		X	
3. Sistema Operacional	Instância do BD		X	
4. Instância	BD		X	
5. Banco de Dados	Esquema		X	
6. Tabela	Linha			X

Conforme explica Reinwald (2010), os modelos demonstrados na Tabela 1 e que correspondem às linhas 1, 2 e 3, compartilham recursos nos níveis das mesmas máquinas físicas com diferentes níveis de abstração (e.g. múltiplas VMs, contas de usuários diferentes, diferentes instâncias do SGBDs). Não existe, porém, compartilhamento de recursos de banco de dados, o que mantém independentes as instâncias do SGBDs. Já os

modelos correspondentes às linhas de 4 a 6 compartilham processos de banco de dados em vários níveis de isolamento (e.g. diferentes banco de dados, esquema e linha). O tipo de armazenamento de dados dos inquilinos vai depender do modelo. Se o modelo escolhido for o de *hardware* compartilhado, será utilizada a virtualização para chavear VMs na mesma máquina, onde cada VM possui apenas um processo de banco de dados e corresponde a um inquilino. Ao passo em que, se o modelo escolhido for o de tabela compartilhada, serão armazenados dados de vários inquilinos em uma mesma tabela, onde algumas linhas de uma tabela correspondem a um inquilino.

Ao analisar o tipo de recurso compartilhado, o estudo apresentado por Hui, Jiang, Li e Zhou (2009) descreve três abordagens para a construção de um banco de dados multi-inquilino:

a) Banco de Dados e Instância de Banco de Dados Independentes

Nesta abordagem cada inquilino cria seu próprio banco de dados (construídos com base nos atuais bancos de dados sem nenhuma extensão) e armazena seus dados através da interação com uma instância dedicada do banco de dados, compartilhando apenas o *hardware*. A Figura 5 mostra esta abordagem.

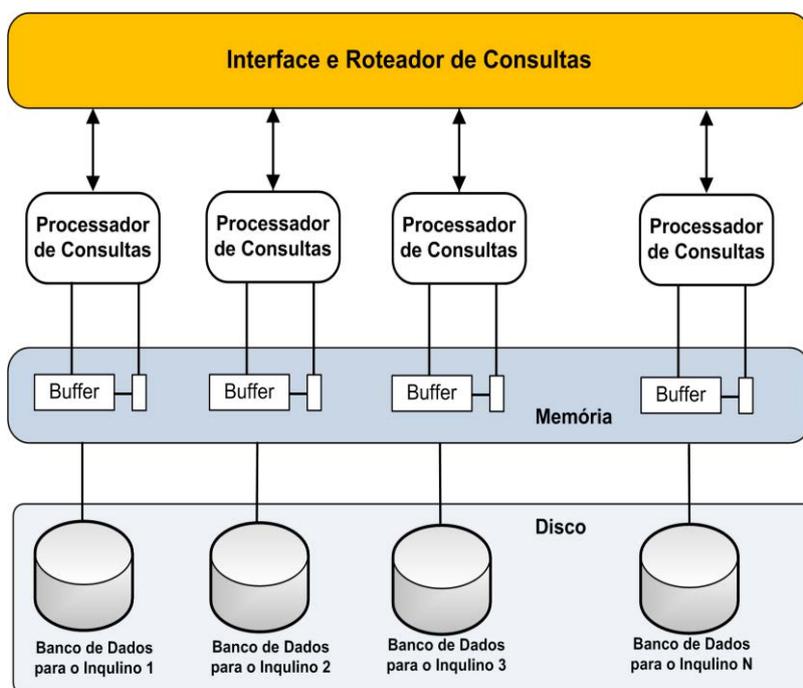


Figura 5. Banco de Dados Independentes e Instância de Banco de Dados independente (Sousa *et al.*, 2011)

Conforme pode ser observado na Figura 5, o provedor de serviços executa instâncias do banco de dados independentes para servir a inquilinos independentes.

b) Tabelas Independentes e Instância de Banco de Dados Compartilhados

Esta abordagem traz o compartilhamento do hardware e de instâncias de banco de dados pelos inquilinos, no qual cada inquilino utiliza esquemas privados do banco de dados compartilhado. Cada nome de estrutura privada possui um identificador para indicar quem é o proprietário da tabela. Ao se realizar uma consulta, esta é reescrita (por um roteador de

consultas) a fim de identificar o nome das tabelas modificadas. Esta abordagem é ilustrada na Figura 6.

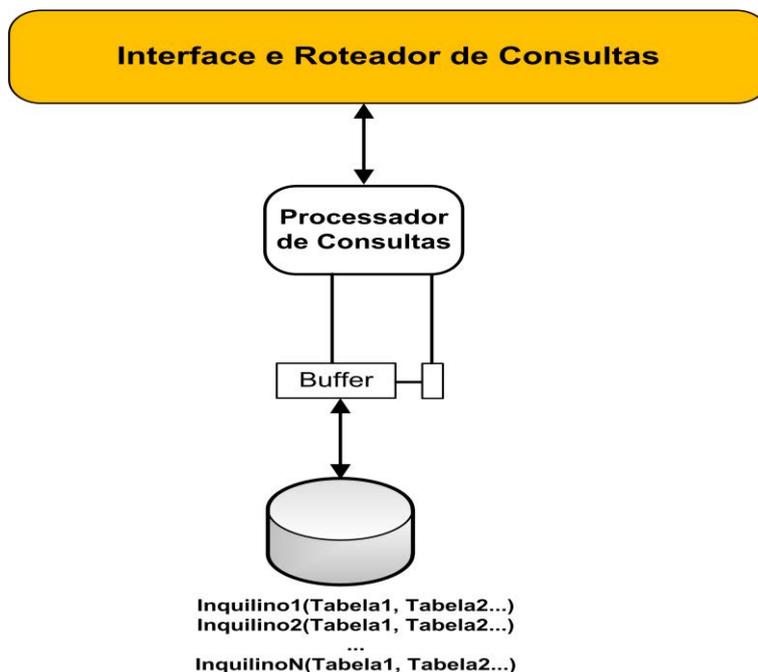


Fig. 6. Tabelas Independentes e Instância de Banco de Dados Compartilhados (Sousa *et al.*, 2011)

Conforme mostrado na Figura 6, o provedor de serviços mantém uma grande base de dados que serve a todos os inquilinos.

c) Tabelas e Instância de Banco de Dados Compartilhados

Nesta abordagem os inquilinos compartilham tabelas e instâncias do banco de dados. O provedor inicializa o banco de dados compartilhado criando tabelas vazias de acordo com o esquema base. Os inquilinos armazenam suas tuplas em tabelas compartilhadas anexando um identificador a cada tupla e define os atributos não utilizados com *null*. Para recuperar as tuplas na tabela compartilhada, um roteador de consultas é usado a fim de reformular as consultas de acordo com o identificador. A Figura 7 mostra esta abordagem.

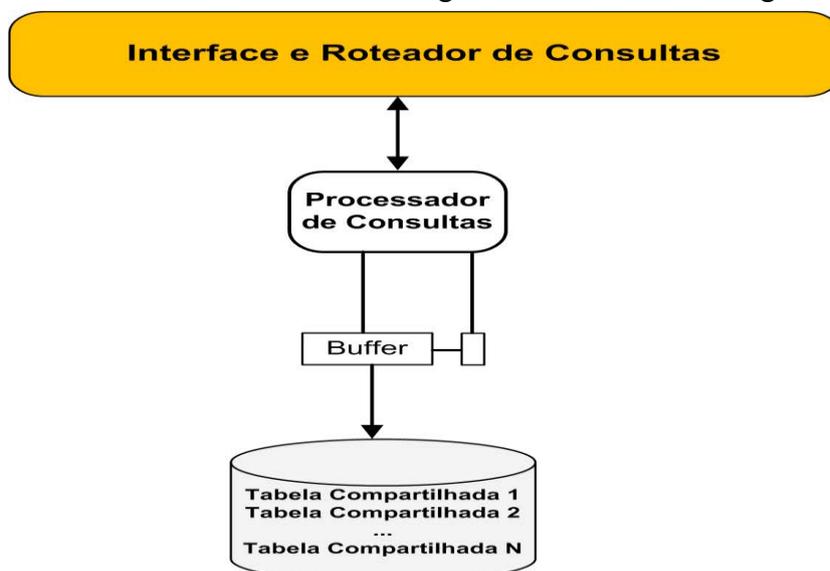


Figura 7. Tabelas e Instância de Banco de Dados Compartilhados (Sousa *et al.*, 2011)

O provedor de serviço mantém apenas uma instância de banco de dados simples, conforme se verifica na Figura 7.

Para melhor compreensão destas três abordagens, a Tabela 2 apresenta uma análise comparativa mostrando as principais vantagens, desvantagens e desafios referentes a cada uma delas:

Tabela 2. Tabela comparativa entre tipos de Bancos de Dados Multi-inquilinos
 Fonte: Resultado da pesquisa.

Abordagem	Vantagens	Desvantagens	Desafios
Banco de Dados Independentes e Instância de Banco de Dados independente	Fornecer um bom nível de isolamento dos dados e segurança.	Alto custo de manutenção para o provedor de serviço; Escalabilidade: o número de instâncias cresce proporcional ao número de inquilinos	Busca de Técnicas de virtualização capazes de minimizar os problemas.
Tabelas Independentes e Instância de Banco de Dados Compartilhados	Custo de manutenção reduzido para o gerenciamento de diferentes instâncias.	Escalabilidade: está limitada ao número de estruturas que o SGBD pode manipular e à memória disponível.	Busca de técnicas e ferramentas que resolvam o problema de escalabilidade
Tabelas e Instância de Banco de Dados Compartilhados	Escalabilidade: o número de tabelas do banco de dados é independente do número de inquilinos.	Necessidade de indexação e otimização: os inquilinos compartilham as mesmas tabelas, mas apresentam requisitos diferentes.	Necessidade de melhorias referentes aos aspectos de isolamento do banco de dados

Conforme ilustra a Tabela 2, cada uma das abordagens apresenta limitações que desafiam o estudo e a pesquisa a fim de minimizar problemas referentes à escalabilidade, alto custo de manutenção, isolamento do banco de dados, dentre outros.

2.5.4 Características do Gerenciamento de Dados em Nuvem

Agrawal, Das e Abbadi (2010) aduzem que o gerenciamento de dados em nuvem pode ser organizado em duas classes de sistemas: a primeira para apoiar aplicações com muitas atualizações e a segunda para análises dos dados e suporte à decisão.

A pesquisa realizada por Voicu e Schuldt (2009), apresenta uma tabela que põe em evidência as principais características do gerenciamento de dados em nuvem. Neste estudo, os dados são gerenciados em poucos centros de dados e acessados por meio de APIs simples, SQL ou variações. Os SGBDs em nuvem devem suportar atualizações concorrentes e transações ACID ou variações. ACID é o acrônimo para “Atomicidade, Consistência, Isolamento e Durabilidade”, Sousa *et al.* (2011) explicam que as propriedades das transações ACID apresentam como principais características a atomicidade e o isolamento, garantidos pela escrita de um único objeto. A durabilidade é obtida por meio

de escrita replicada, porém, a consistência apresenta debilidades. A replicação deve ser transparente para os usuários e fornecer garantias de QoS, além da granulosidade fina dos dados. A Tabela 3 apresenta estas características.

Tabela 3. Características do Gerenciamento de Dados em Nuvem (Voicu & Schuldt, 2009)

Distribuição	Poucos centros de dados
Ambiente	Recursos homogêneos em centros de dados
Operações Para acesso aos dados	API simples, SQL ou variações
Atualização	Suporte às atualizações concorrentes
Transações	ACID ou variações
Replicação	Garantia de QoS e transparência
Granulosidade da Replicação	Fina
Controle Global	Central ou Distribuído
Alterações Dinâmicas	Escalabilidade e suporte para cargas de trabalho inesperadas

Observando-se a Tabela 3, temos que os SGBDs podem apresentar um controle global central ou distribuído na distribuição de dados e devem fornecer escalabilidade e suportar cargas de trabalho inesperadas.

2.5.5 Armazenamento de dados em nuvem

Existem diversas propostas que tratam do armazenamento de dados e processamento de consultas em nuvem e cada sistema utiliza uma abordagem específica, dentre as quais neste trabalho se destacam algumas.

O projeto de código livre *Hadoop File System* (HDFS) (Apache, 2010) armazena grandes arquivos em vários servidores e obtém a confiabilidade por meio da replicação de dados. Inspirado no *Google File System* (GFS) sistema de arquivos distribuídos proprietário desenvolvido pelo Google e projetado especialmente para fornecer acesso eficiente e confiável aos dados usando grandes *clusters* de servidores, os dados são armazenados em nós geograficamente distribuídos.

DeCandia *et al.* (2007) explicam que algumas propostas para o armazenamento e processamento utilizam a estrutura chave-valor em uma *Distributed HashTable*, que armazena o valor e a chave associados de forma desnormalizada, o que facilita a distribuição dos dados entre os nós do sistema, sendo que cada um destes nós possui parte dos dados. As APIs básicas de acesso geralmente são simples, sendo que as mais sofisticadas permitem a execução de funções definidas pelo usuário no ambiente do servidor tais como *execute(key, operation, parameters)*. Para acessar ou modificar dados é necessário fornecer uma chave.

Já em Chang *et al.* (2006) é apresentada outra abordagem para armazenar e processar dados em nuvem que consiste em utilizar uma estrutura de colunas ou *arrays* multidimensionais. Os dados são armazenados em tabelas que possuem diversas colunas. Cada coluna armazena um valor, acessado por meio de uma chave.

Algumas abordagens são consideradas orientadas a documento, nas quais as técnicas são desenvolvidas para o armazenamento, modelagem, consulta e apresentação de dados de forma a refletir a estrutura de um documento. O formato *Java Script Object Notation* (JSON) tem sido bastante utilizado neste contexto.

Rodriguez e Neubauer (2010) trazem uma abordagem baseada em grafo para auxiliar na modelagem de estruturas complexas por meio da representação utilizando grafos. Este gerenciamento obedece aos seguintes padrões: o nó, ou vértice, possui o mesmo conceito de uma instância de um objeto com identificador único; os relacionamentos ou arestas fornecem uma ligação entre dois nós; propriedades ou atributos: são pares de *strings* chave-valor do objeto que podem existir tanto em um nó quanto em um relacionamento.

2.5.6 Transações

A utilização de transações distribuídas define o controle do processamento de dados no ambiente de computação em nuvem. Faz-se necessário utilizar protocolos de replicação de dados, terminação distribuída e sincronização de acesso para garantir as propriedades ACID ou variações destas (Voicu & Schuldt, 2009).

Gilbert e Lynch (2002) tratam de um ponto importante na construção de sistemas distribuídos no teorema *Consistency, Availability, Partition Tolerance* (CAP). Este teorema mostra que os sistemas distribuídos podem assegurar simultaneamente apenas duas das seguintes propriedades: a) consistência: todos os nós têm a mesma visão dos dados ao mesmo tempo; b) disponibilidade: falhas em nós não impedem os demais nós de continuar a operar; c) tolerância a partições: o sistema continua a operar mesmo com a perda arbitrária de mensagens. Quando ocorre uma falha de rede, é necessário escolher quais propriedades são mais importante para o sistema.

2.5.7 Escalabilidade e Desempenho

É possível identificar duas dimensões de escalabilidade: vertical e horizontal no enorme conglomerado de redes que compõe a nuvem. Para obtenção de escalabilidade vertical melhora-se a capacidade do *hardware*, incrementando individualmente os nós. Esse processo pode ser realizado pela disponibilização de um servidor com mais memória física ou da melhoria da largura de banda que conecta dois nós. Já a escalabilidade horizontal consiste em aumentar o número de máquinas a fim de distribuir as requisições entre estas máquinas.

A escalabilidade envolvendo dados é uma tarefa complexa, já que a maioria dos SGBDs em nuvem utiliza arquiteturas não compartilhadas, tornando a disposição dos dados um ponto chave. Portanto, a fragmentação dos dados deve ser feita de forma a minimizar o transporte de dados, o qual adiciona um custo relevante ao processamento. Em relação ao desempenho, as dificuldades existentes são em relação aos problemas de tempo de resposta, em virtude das diferentes tecnologias e heterogeneidade do hardware utilizado, o que pode afetar o desempenho.

2.5.8 Tolerância a Falhas e Distribuição de Dados

As soluções em nuvem utilizam técnicas que auxiliam a distribuição dos dados no tratamento das falhas, a exemplo do DHT, que por meio da fragmentação dos dados torna possível melhorar a disponibilidade e distribuir a carga, tanto para operações de escrita quanto de leitura. Na ocorrência de falha uma máquina, os dados pertencentes a essa máquina são afetados, mas não o armazenamento de dados. Conforme Soror *et al.* (2010), também é possível criar e iniciar novas réplicas rapidamente por meio de máquinas virtuais.

3 Service-Oriented Architecture (SOA)

Sob a ótica de Raines (2009) a SOA é baseada em engenharia de informática do passado, para oferecer uma abordagem arquitetônica para sistemas empresariais orientados à oferta de serviços de uma rede de consumidores. Sendo uma abordagem orientada a serviços, seu foco se concentra na definição de serviços, interfaces e previsíveis comportamentos de serviço.

Sordi, Marinho e Nagy (2006) afirmam que para conceituar a SOA, faz-se necessário desenvolver o termo “serviço” dentro do contexto da arquitetura de software. O “serviço básico” é o mais comumente encontrado em quantidade nas organizações, a exemplo dos algoritmos para validação de números ou de status, como os utilizados na validação do número do CPF e na averiguação da situação de crédito de uma pessoa. Outrossim, um serviço também pode ser um algoritmo mais abrangente, denominado de “serviço centrado em processo”. No ambiente computacional, há diversas outras variações de serviços; A Tabela 4 ilustra a tipificação de serviços desenvolvida por Krafzig, Banke e Slama (2004).

Tabela 4. Tipificação de serviços encontrados na SOA (Krafzig *et al.*, 2004)

ATRIBUTOS DOS SERVIÇOS	TIPOS DE SERVIÇOS			
	Serviço Básico	Serviço Intermediário	Serviço Centrado em Processo	Serviço Empresarial
Descrição	Serviço simples centrado em dados ou lógica	Conectores e adaptadores tecnológicos e serviços para adição de funcionalidades	Encapsula a lógica do processo	Serviço compartilhado com outras empresas ou parceiros
Complexidade da implementação	De baixa a moderada	De moderada a alta	Alta	Específica de cada serviço
Gerenciamento de Estado	Não	Não	Sim	Específica de cada serviço
“Reusabilidade”	Alta	Baixa	Baixa	Alta
Frequência da Mudança	Baixa	De moderada a alta	Alta	Baixa
Elemento Requerido ao SOA	Sim	Não	Não	Não

Conforme visto na Tabela 4, os serviços podem variar sob vários aspectos como a funcionalidade oferecida, complexidade de implementação e reusabilidade.

Outra entidade em destaque por se tratar de um elemento ativo da SOA é o *application frontend*. Ele é caracterizado pelas diversas aplicações que interagem com o usuário final (e.g. web applications, GUI applications, CICS applications), sendo o responsável por iniciar todos os processos de negócios e receber seus resultados.

Existem ainda outros dois componentes fundamentais para o entendimento da arquitetura SOA:

- ✓ O repositório de serviços: é a entidade que auxilia o descobrimento dos serviços disponíveis na arquitetura, sobretudo daqueles fora do processo de desenvolvimento do nosso sistema. Disponibiliza informações como: localização virtual, provedor, taxas, limitações técnicas, aspectos de segurança, entre outras.
- ✓ O barramento de serviços: é o meio utilizado para conexão entre todos os participantes da SOA (serviços e *application frontends*) e engloba uma grande diversidade de produtos e conceitos.

Segundo Diirr, Faria, Azevedo, Santoro e Baião (2010) os sistemas orientados a serviços são desenvolvidos através da composição de serviços compartilhados entre as organizações. Para tal, SOA busca maior flexibilidade, agilidade e reuso de aplicações. O desenvolvimento de serviços requer preocupações adicionais ao desenvolvimento de software tradicional, tais como: cooperação entre os papéis arquiteturais de SOA (provedores, consumidores e *brokers*); bom entendimento dos modelos de negócio e relacionamento entre os parceiros de negócio; saber lidar com requisitos conflitantes; alinhar os requisitos de negócio com soluções de TI; distribuir serviços através das fronteiras da organização; produzir serviços de forma a possibilitar o reuso dos mesmos etc.

Em Krafzig, *et al.* (2004) foram identificados três estágios de maturidade da SOA nas organizações conforme se segue:

- ✓ SOA fundamental: a maior parte da complexidade e da responsabilidade reside no *application frontend*, que aciona os diversos serviços básicos. Seus maiores benefícios são os ganhos de manutenção com as aplicações internas de departamentos e com as aplicações entre departamentos;
- ✓ SOA em rede: caracteriza-se pelo uso intensivo de serviços intermediários que são utilizados para atender deficiências técnicas e funcionais dos *softwares* disponíveis na arquitetura, pois enquanto desempenham o papel de conectores e adaptadores, oferecem flexibilidade para integração entre *softwares*. Apresentam vantagens como os ganhos de flexibilidade ao utilizar aplicações entre unidades de negócios e entre diferentes organizações;
- ✓ SOA habilitadora de processos: a complexidade dos processos de negócios é delegada a SOA, isto é, aos serviços centrados em processos, ficando a utilização dos *application frontend* apenas para interação com o usuário. Também há separação entre as regras para gestão do processo de negócio, e os códigos fontes de programas necessários para a sua execução.

Com a engenharia pensada sob o ponto de vista empresarial, a SOA oferece importantes benefícios, tais como:

- ✓ **Integração com neutralidade de linguagem:** os padrões fundacionais contemporâneos *Web Services* usam *eXtensible Markup Language*, cujo formato não depende de plataformas de hardwares ou de softwares, criando uma infraestrutura única para diferentes linguagens.
- ✓ **Interoperabilidade:** a independência de plataformas de *hardwares* ou de *softwares* possibilita que estes sistemas possam oferecer e invocar serviços através de um mecanismo comum.

- ✓ **Reutilização dos componentes:** vários componentes podem ser combinados para oferecer mais recursos, o que é frequentemente chamado de "Orquestração".
- ✓ **Produtividade:** com o reuso, a equipe de desenvolvimento pode reutilizar serviços em outros projetos, diminuindo assim o tempo de desenvolvimento.
- ✓ **Manutenibilidade:** uma importante característica SOA é o baixo acoplamento, que facilita a manutenção dos serviços.
- ✓ **Flexibilidade:** a capacidade de isolamento da estrutura de um serviço possibilita que as mudanças sejam feitas com maior facilidade.
- ✓ **Alinhamento com o negócio:** possibilita que a área de negócio visualize os processos alinhados com a tecnologia.
- ✓ **Agilidade organizacional:** a SOA define construções de blocos de capacidade de *software* em termos de serviços oferecidos que atendam a alguma parte dos requisitos da organização. Estes blocos podem ser recombinados e rapidamente integrados.
- ✓ **Aproveitamento dos sistemas existentes:** Uma prática comum de SOA é definir elementos ou funções existentes em sistemas aplicativos e disponibilizá-los para a empresa em um padrão acordado, permitindo integração entre sistemas novos e antigos componentes.
- ✓ **Governança:** proporciona o gerenciamento nos processamentos de negócio.
- ✓ **Padronização:** é baseada no uso de padrões.
- ✓ **Abstração:** o serviço é totalmente abstraído da sua implementação.

Um dos motivos que levam a adoção de *Web Services* é o aumento da amplitude de integração sendo tentada em organizações. A integração de sistemas está aumentando em complexidade tanto dentro das organizações como entre organizações externas. A tendência é que esta complexidade continue a crescer pela combinação de várias fontes de dados para fornecer o maior valor da informação. Raines (2009) *apud* Ronan Bradley (2008), ao explicar que CIOs sempre têm dificuldade em justificar os custos associados com a integração, mas, no entanto, a fim de oferecer soluções atraentes para os clientes ou melhorar eficiência operacional, uma organização precisa superar o desafio de integração.

A Figura 8 retrata alguns pontos de interesse na tendência da crescente complexidade de integração de sistemas.

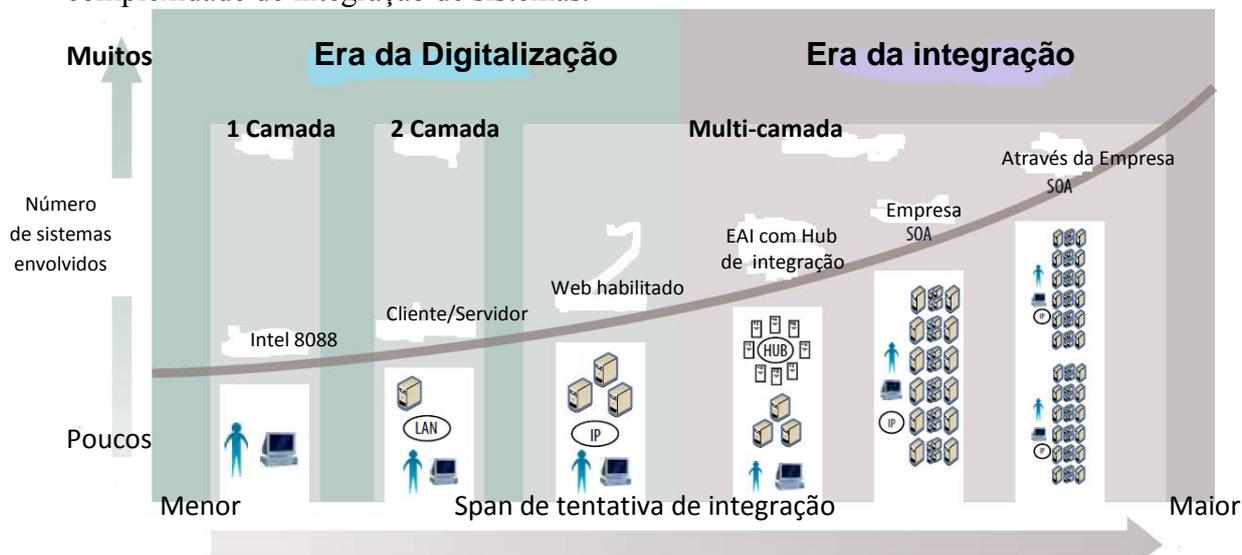


Figura 8. Crescimento da complexidade de integração dos sistemas (Raines, 2009)

A Figura 8 ilustra que conforme aumenta o número dos sistemas envolvidos no processo, maior é a complexidade exigida para integrá-los. A SOA tenta agilizar a integração entre sistemas fornecendo componentes que são arquitetados e descritos de uma forma consistente.

Da mesma forma como acontece com a computação em nuvem, SOA traz consigo uma série de benefícios chaves e riscos, incluindo:

- ✓ **Dependência na rede:** SOA é fundamentalmente dependente da rede para conectar o provedor de serviços com o consumidor. Por exemplo, os protocolos *Web Service* utilizam protocolos de *Internet* para invocar funções de *softwares* distribuídos através da rede. Consequentemente, o mau desempenho da rede pode prejudicar a disponibilidade de *Web Services* para o consumidor.
- ✓ **Custo Provedor:** Criar um componente de *software* genérico reutilizável predestinado para um público amplo requer mais recursos do que criar uma solução menos genérica. O custo da reutilização, portanto, muda para os prestadores de serviços, os quais por sua vez, beneficiam os consumidores (Poulin, 2006).
- ✓ **Padrões da empresa:** quando muitos componentes estão sendo desenvolvidos simultaneamente por equipes individuais, escolher e comunicar um conjunto abrangente de padrões da empresa é importante para ajudar a integração SOA da empresa.
- ✓ **Agilidade:** a "agilidade" para SOA é referente a questões organizacionais, ou seja, trata-se da capacidade de se adaptar mais rapidamente a ferramentas da organização para atender seus atuais requisitos. Os requisitos de TI podem mudar por uma série de razões: mudanças no negócio ou missão, mudanças na lei, nas novas tecnologias no mercado comercial, etc. A maior promessa de uma SOA empresarial é que uma vez que uma quantidade suficiente dos componentes do legado envolto existe e é acessível pelos protocolos da Internet (IP), eles podem ser mais rapidamente reagrupados para resolver problemas novos.

4 Cloud Computing e SOA

A *Cloud Computing* e a SOA compartilham importantes sobreposições e preocupações, conforme ilustra a Figura 9.

Computação em Nuvem	Sobreposições	SOA via <i>Web Services</i>
<ul style="list-style-type: none"> - Software como serviço - Computação utilitária - <u>Terabytes</u> sob demanda - Dados distribuídos na nuvem - Plataforma como serviço - Padrões envolvendo diferentes camadas da pilha 	<ul style="list-style-type: none"> - Camada de aplicação componentes/serviços - Dependência da rede - <u>Cloud/WAN</u>- invocação de serviços suportados - Alavancagem dos ativos de software distribuído - Modelo produtor/consumidor 	<ul style="list-style-type: none"> - Foco na integração de sistemas - Consistência motriz da integração - Integração de aplicações empresariais - Normas de implementação maduras (REST, SOAP, HSDL, UDDI)

Figura 9. *Cloud Computing* e SOA- Sobreposições e interesses comuns (Raines, 2009)

A Figura 9 mostra a interdependência entre a *Cloud Computing* e SOA descrevendo algumas das principais preocupações e sobreposições compartilhadas entre estas duas abordagens. No entanto, a sobreposição mais importante ocorre na área dos serviços em nuvem, que são componentes de aplicativos e *softwares* de serviços acessíveis pela rede, tais como *Web Services*.

Lunn (2008) aduz que, tanto a *Cloud Computing* como a SOA compartilham dos conceitos de orientação serviços. A computação em nuvem se concentra em transformar os aspectos da computação de TI em *commodities* (mercadorias), sendo atualmente o maior desafio de SOA.

Por sua vez, a SOA, embora não se restrinja conceitualmente a *software* é implementada na prática como componentes (que podem ser combinados e executados em várias plataformas em toda a rede para fornecer uma função de negócios) ou serviços de *softwares*, como exemplificado pelos padrões de serviços da *Web*.

4.1 Dependência da rede

Tanto a rede de computação em nuvem quanto a SOA têm a mesma fraqueza estrutural fundamental, pois dependem de uma rede robusta para conectar os consumidores e os produtores (Naughton, 2008, conforme referido em Raines, 2009).

4.2 Formas de terceirização

Ambos os conceitos requerem formas de relações contratuais e de confiança entre prestadores de serviços e consumidores.

A reutilização de um serviço SOA por um grupo de outros sistemas é na verdade uma "terceirização" da capacidade para outra organização.

Em se tratando de computação em nuvem, a terceirização é mais evidente, sendo na maioria das vezes de caráter comercial. Armazenamento, plataformas e servidores são alugados de provedores comerciais que têm economias de escala no fornecimento desses produtos.

4.3 Adoção de Padrões

Tanto a computação em nuvem quanto a SOA proporcionam uma organização com oportunidade para selecionar padrões comuns para capacidades acessíveis de rede.

A SOA tem um conjunto amadurecido de normas com o qual serviços de software são implementados, como SOAP (W3C, 2003) *Representation a State Transfer* (REST) e *Web Services Description Language* (WSDL), dentre outros (Raines, 2009).

Na *Cloud Computing* (ainda em fase de amadurecimento) muitas das *interfaces* oferecidas são únicas para um determinado vendedor, aumentando assim o risco de vendedor *lock-in*. Do estudo de Raines (2009) transcrevemos a seguinte observação:

"A capacidade de alternar entre os prestadores supera a maior preocupação do uso de tais prestadores de serviços, a falta de segunda opção de *sourcing* (abastecimento) e o medo de fornecedor *lock-in* (as fraquezas subsequentes do controle

estratégico e falta de concorrência de preços). A pilha de computação a partir das aplicações que nós escrevemos está se movendo de um produto para uma economia baseada em serviços. A mudança para os serviços também vai levar à normalização de ordens inferiores da pilha de informática para *internet* fornecida de componentes ".

5 Resultados Observados

Embora existam importantes sobreposições e interesses comuns entre os dois conceitos abordados, existe uma ênfase diferente, resultante do foco original nos diferentes conjuntos de problemas.

Implementações de SOA são fundamentalmente tecnologias de integração empresarial de troca de informações entre os sistemas de sistemas. A implementação de tecnologias SOA como o grupo de padrões de serviços da *Web*, permite que uma aplicação de *software* consumidor possa invocar serviços através de uma rede comum. A integração ocorre através de uma variedade de linguagens de desenvolvimento e plataformas, fornecendo uma camada de *software* de linguagem neutra.

O principal benefício dos esforços da “empresa SOA” é a capacidade de fazer sistema-a-sistema, interfaces consistentes na arquitetura corporativa, economizando recursos relacionados à futura integração e proporcionando melhora da velocidade com que a integração pode ocorrer ou agilidade organizacional.

Por sua vez, a ênfase da *Cloud Computing* é aproveitar a rede para terceirizar funções de TI, o que vai muito além de *software* como serviço. Isto significa permitir ao mercado ofertar muitas funções de TI como *commodities*, reduzindo assim o custo para os consumidores, se comparados com a operarem internamente. Conforme observa Raines (2009), a computação em nuvem permite o acesso sob demanda e através da *Web*, aos recursos de TI virtualizados que estão alojados fora do seu próprio centro de dados, consistindo em um processo simples de usar e pago através de assinatura.

5.1 Desafios encontrados

Alguns desafios foram detectados durante o desenrolar deste trabalho, os quais são agora apresentados:

5.1.1 Desafios enfrentados pela Computação em Nuvem

- ✓ Verificou-se que no modelo SaaS de fornecimento de *software*, as soluções e infra-estruturas ainda são consideradas de custo muito elevado, o que restringe o alto aproveitamento de recursos a um número menor de usuários (Souza *et al.*, 2011).
- ✓ No que se refere ao armazenamento de dados em nuvem, os principais desafios detectados incluem segurança, gerenciamento de recursos compartilhados e extensibilidade (Souza *et al.*, 2011).
- ✓ Técnicas de virtualização são sugeridas a fim de minimizar problemas como alto custo de manutenção para gerenciar diversas instâncias de banco de dados nos modelos de Banco de Dados multi-inquilino do tipo Banco de Dados e Instância de Banco de Dados Independentes.
- ✓ O armazenamento de dados através da proposta de Banco de Dados Multi-inquilinos, na modalidade *Tabela Compartilhadas e Instância de Banco de Dados*

Compartilhados apresenta algumas desvantagens, tais como a necessidade de indexação e otimização, já que os inquilinos compartilham as mesmas tabelas, mas apresentam requisitos diferentes.

- ✓ Sousa *et al.* (2011) pontuam que a falta de estudos detalhados referentes aos aspectos de isolamento do banco de dados e segurança, no contexto da modalidade *Tabela Compartilhadas e Instância de Banco de Dados Compartilhados*, tem contribuído para aumentar as dificuldades de utilização desta proposta.
- ✓ Além dos altos preços dos *softwares* e da necessidade de constante atualização de *hardwares*, outro importante desafio no armazenamento de dados em nuvem e que merece atenção dos estudiosos em TI, diz respeito à necessidade do usuário em obter um desempenho satisfatório expresso em termos de latência e vazão.

5.1.2 Desafios enfrentados pela SOA

Estudiosos concordam que uma das principais preocupações em implementações dessa arquitetura diz respeito a segurança. Além disso, a arquitetura a SOA depende da implementação de normas, não é utilizada em aplicações com grande transferência de dados, alto acoplamento e aplicações que precisam manter estado. Alguns desafios detectados são listados a seguir (Oliveira, 2013).

- ✓ **Complexidade:** um grande volume de serviços que precisam ser gerenciados.
- ✓ **Performance:** a performance do serviço depende diretamente do servidor onde o serviço está publicado e da rede.
- ✓ **Robustez:** caso uma exceção aconteça, ações para reverter o processo são inviáveis.
- ✓ **Disponibilidade:** a dependência da rede e do servidor faz com que haja interrupção dos serviços, caso haja interrupção em algum deles.
- ✓ **Testabilidade:** o debug no serviço é um problema para os desenvolvedores.
- ✓ **Segurança:** como os serviços estão disponíveis na rede, os dados trafegados estão vulneráveis a serem interceptados.

5.1.2 Desafios enfrentados pela SOA e pela Computação em Nuvem

Este trabalho identificou que a Computação em Nuvem e a SOA enfrentam problemas semelhantes tidos como pontos chave, tais como: segurança, interoperabilidade, confiabilidade, disponibilidade, performance e escalabilidade (no caso da Computação em Nuvem).

A interoperabilidade, que é uma das principais propostas da SOA, se apresenta como característica amplamente desejável no ambiente de Computação em Nuvem. Existe, porém, a necessidade de implementação de padrões e interfaces para que a portabilidade seja possível (Dikaiakos, Pallis, Katsaros, Mehra, & Vakali, 2009).

A confiabilidade está relacionada à frequência com que o sistema falha e à capacidade de reação do sistema, a fim de diminuir os impactos negativos e evitar a perda de dados. As aplicações desenvolvidas devem possuir uma arquitetura que permita que os dados permaneçam intactos mesmo que haja falhas ou erros em um ou mais servidores ou máquinas virtuais sobre os quais essas aplicações estão executando (Dikaiakos *et al.* 2009).

A disponibilidade é outra preocupação das duas abordagens estudadas, pois ambas necessitam do funcionamento da internet que também é um serviço que não possui disponibilidade ao nível de uma rede local.

A performance dos serviços oferecidos tanto pela SOA quanto pela Computação em Nuvem dependem diretamente da disponibilidade e desempenho da rede (Oliveira, 2013).

A escalabilidade é uma característica fundamental na Computação em Nuvem, pois as aplicações para uma nuvem precisam ser escaláveis (ou “elásticas”). Esta tarefa pode não ser simples e normalmente depende da implementação (SUN, 2009).

A segurança é o maior desafio a ser enfrentado, pois a informação que antes era armazenada localmente irá trafegar pela *internet* através da utilização dos serviços SOA, podendo ser armazenada na nuvem, em local físico que não se tem precisão onde é e nem que tipos de dados estão sendo armazenados junto a ela, no caso da Computação em Nuvem. A privacidade e integridade das informações, portanto, tornam-se premissas de suma importância (Kaufman, 2009).

6 Conclusão

Tanto a Computação em Nuvem quanto a SOA apoiam boas práticas de engenharia, adotando conceitos fundamentais, tais como abstração, desacoplamento, encapsulamento e reutilização. Embora ambas compartilhem muitas preocupações em comum, sugerindo definição de *interfaces* claras e inequívocas, desempenho e comportamento previsíveis, elas podem ser perseguidas de forma independente ou em simultâneo, como atividades complementares. Como parte conclusiva, este trabalho observou algumas premissas que passam a serem relatadas a seguir.

A computação em nuvem não substitui a SOA, pois a necessidade de apoiar ampla e mais consistentemente a integração de sistemas continua a existir. A proposta SOA inspirada em componentização, na qual o *software* aproveita outro *software* baseado em rede, é uma resposta ao aumento da tendência que pressiona para baixo os orçamentos de TI.

Em se tratando de modelos de serviços propostos pela Computação em Nuvem, verificou-se que no modelo SaaS de fornecimento de *software*, ainda se faz necessário desenvolver soluções e infraestruturas de baixo custo, para assim atingir um público não suportado hoje em dia.

No que tange o armazenamento de dados em nuvem, diversos desafios de pesquisa ainda são impostos à comunidade especializada, incluindo segurança, gerenciamento de recursos compartilhados e a extensibilidade. Um exemplo está na modalidade *Tabela Compartilhadas e Instância de Banco de Dados Compartilhados* que apresenta a necessidade de indexação e otimização pelo fato dos inquilinos compartilharem as mesmas tabelas, mas apresentarem requisitos diferentes.

Os altos preços dos *softwares*, a necessidade de constante atualização de *hardwares* e a necessidade do usuário em obter um desempenho satisfatório expresso em termos de latência e vazão também desafiam a pesquisa em ambas as abordagens.

A dependência da rede apresenta problemas tanto para a SOA quanto para a Computação em Nuvem, cujos principais se traduzem em segurança, disponibilidade, performance, robustez e confiabilidade.

Este trabalho forneceu um levantamento das principais características da Computação em Nuvem e da SOA, identificando a proposta de cada conceito. Foi realizado um levantamento das semelhanças, sobreposições, interesses comuns e diferenças entre estas abordagens. Também foram detectados os desafios enfrentados por cada uma em separado e em conjunto. Ao final foram apresentadas algumas premissas conclusivas, as quais se espera que sirvam de incentivo à realização de pesquisa que contribuam para uma melhor utilização em conjunto dessas duas tecnologias.

7 Referências

- Abouzeid, A., Bajda-Pawlikowski, K., Abadi, D. J., Rasin, A., & Silberschatz, A. (2009). *Hadoopdb: An architectural hybrid of mapreduce and dbms technologies for analytical workloads*. PVLDB, 2(1):922–933.
- Agrawal, D., Abbadi, A. E., Emekci, F., & Metwally, A. (2009). *Database management as a service: Challenges and opportunities*. Data Engineering, International Conference on, 0:1709–1716.
- Agrawal, D., Das, S., & Abbadi, A. E. (2010). *Big data and cloud computing: New wine or just new bottles?* In PVLDB, 3(2):1647–1648.
- Apache.org (2010). *Apache Hadoop*. Disponível em: < <http://hadoop.apache.org>>.
- Aulbach, S., Grust, T., Jacobs, D., Kemper, A., & Rittinger, J. (2008). *Multi-tenant databases for software as a service: schema-mapping techniques*. In SIGMOD '08: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, pages 1195–1206, New York, NY, USA. ACM.
- Brantner, M., Florescu, D., Graf, D., Kossmann, D., & Kraska, T. (2008). *Building a database on s3*. In Proceedings of the 2008 ACM SIGMOD International Conference on Management of data - SIGMOD '08, page 251, New York. ACM Press.
- Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J. & Brandic, I. (2009). *Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility*. Future Gener. Comput. Syst., 25(6):599–616.
- Chang, F., Dean, J., Ghemawat, S., Hsieh, W. C., Wallach, D. A. Burrows, M., Chandra, T., Fikes, A., & Gruber, R. E. (2006). *Bigtable: a distributed storage system for structured data*. In OSDI '06: Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation, pages 15–15, Berkeley, CA, USA. USENIX Association.
- Curino, C., Jones, E., Zhang, Y., Wu, E., & Madden, S. (2010). *Relational cloud: The case for a database service*. Technical report, MIT-CSAIL-TR-2010-014. Computer Science and Artificial Intelligence Laboratory, MIT, USA.
- DeCandia, G., Hastorun, D., Jampani, M., Kakulapati, G., Lakshman, A., Pilchin, A., Sivasubramanian, S., Voshall, P., & Vogels, W. (2007). *Dynamo: amazon's highly available key-value store*. SIGOPS Oper. Syst. Rev., 41(6):205–220.

Diirr, T., Faria, F., Azevedo, L., Santoro F. & Baião F., (2010). *Projeto com UML e implementação de serviços em SOA*. Departamento de Informática Aplicada, Universidade Federal do Estado do Rio de Janeiro, Rio de Janeiro – Brasil.

Dikaiakos, M. D., Pallis, G., Katsaros, D., Mehra, P. & Vakali, A. (2009). *Cloud Computing – Distributed Internet Computing for IT and Scientific Research*. IEEE Internet Computing, 13(5): 10-13.

Friedman, T. L. (2009), *O Mundo é Plano*. Objetiva, Rio de Janeiro (RJ), Brasil.

Gilbert, S. & Lynch, N. (2002). *Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services*. SIGACT News, 33(2):51–59.

Hadade, R. (2014). *Web Services*. Developer Network, Microsoft. Disponível em: <<http://msdn.microsoft.com/pt-br/library/cc564893.aspx#mainSection>>.

Apache.org (2010). *Apache Hadoop*. Disponível em: <<http://hadoop.apache.org>>.

Hui, M., Jiang, D., Li, G., & Zhou, Y. (2009). *Supporting database applications as a service*. In ICDE '09: Proceedings of the 2009 IEEE International Conference on Data Engineering, pages 832–843, Washington, DC, USA. IEEE Computer Society.

Kaufman, L. M. (2009). *Data Security in the World of Cloud Computing*. IEEE Security and Privacy, 7(4): 61-64.

Krafzig, D., Banke, K. & Slama, D. (2004). *Enterprise SOA: Service-Oriented Architecture Best Practices*. Indianapolis: Prentice Hall.

Lunn, B. (2008), *The New Stack: SaaS, Cloud Computing, Core Technology*. Disponível em: <http://www.readwriteweb.com/archives/new_technology_stack.php>.

Liu, S., Liang, Y., & Brooks, M. (2007). *Eucalyptus: a web service enabled e-infrastructure*. In CASCON '07: Proceedings of the 2007 conference of the center for advanced studies on Collaborative research, pp 1–11, New York, NY, USA. ACM.

Mell, P. & Grance, T. (2009). *Draft NIST Working Definition of Cloud Computing*. National Institute of Standards and Technology. Disponível em: <<http://csrc.nist.gov/groups/SNS/cloud-computing>>.

Microsoft (2010). *Architecture Strategies for Catching the Long Tail*. Disponível em: <<http://msdn.microsoft.com/en-us/library/aa479069.aspx>>.

Microsoft Azure (2011).. Disponível em: <<http://www.microsoft.com/azure/>>.

Milagre, J. A. & Cusin, C. A. (2009). *Governança de TI Verde*. Revista de Instituto Arayara. Disponível em: <<http://www.sustentabilidade.org.br>>. Acesso: 02.09.2013.

Moreira, E. Q., Foscarini, É. D., Junior, G. C. S., Alixandrina, L. A. O., Neto, L. P. V. & Rossetto, S. (2011). *Federação CAFe: Implantação do Provedor de Identidade*. Rio de Janeiro: Escola Superior de Redes, RNP.

- Oliveira, E. M. (2013). *Vantagens e Desvantagens da SOA*. DEVMEDIA. Disponível em: <<http://www.devmedia.com.br/vantagens-e-desvantagens-de-soa/27437>>.
- Poulin, J. (2006). *The ROI of SOA Relative to Traditional Component Reuse*. Logic Library.
- Raines, G. 2009, *Cloud Computing and SOA*. Service-Oriented Architecture (SOA) SERIES, The MITRE Corporation - Approved for Public Release, Distribution Unlimited, Case Number: 09-0743, Document Number: MTR09002.6.
- Reinwald, B. (2010). *Database support for multi-tenant applications*. In IEEE Workshop on Information and Software as Services (WISS).Co-located with ICDE.
- Robinson, D. (2008). *Amazon Web Services Made Simple: Learn how Amazon EC2, S3, Simple DB and SQS Web Services enables you to reach business goals faster*. Emereo Pty Ltd, London, UK, UK.
- Rodriguez, M. A. & Neubauer, P. (2010). *Constructions from dots and lines*. Bulletin of the American Society for Information Science and Technology, 36(6):35–41.
- Sordi, J. O., Marinho, B. L. & Nagy, M. (2006). *Benefits of Software-Oriented Architecture to Corporate Services: An Analysis of ABN AMRO Bank Experience in Brazil*. Journal of Information Systems and Technology Management. Vol. 3, No. 1, 2006, p. 19-34. ISSN online: 1807-1775.
- Soror, A. A., Minhas, U. F., Aboulnaga, A., Salem, K., Kokosielis, P. & Kamath, S. (2010). *Automatic virtual machine configuration for database workloads*. ACM Trans. Database Syst., 35(1):1–47.
- Sousa, F. R. C., Moreira, L. O., Macêdo, A. J. & Machado J. (2011), *Gerenciamento de Dados em Nuvem: Conceitos, Sistemas e Desafios*, Universidade Federal do Ceará (UFC). Disponível em: <<http://www.es.ufc.br/~flavio/papers/sbbd2010.pdf>>.
- SUN Microsystems, INC. *Introduction to Cloud Computing Architecture*. White Paper, 1ª edição, junho 2009a.
- Vecchiola, C., Chu, X., & Buyya, R. (2009). *Aneka: A Software Platform for .NET-based Cloud Computing*. Pp 267–295. In: W. Gentsch, L. Grandinetti, G. Joubert (Eds.). High Speed and Large Scale Scientific Computing. IOS Press, Amsterdam, Netherlands.
- Voicu, L. C. & Schuldt, H. (2009). *How replicated data management in the cloud can benefit from a data grid protocol: there: gridit approach*. In Cloud DB '09: Proceedings of the First International Workshop on Cloud Data Management ,pages 45–48, New York, NY, USA. ACM.
- W3C, World Wide web Consortium (2003). *SOAP Version 1.2 Part 0: Primer, W3C Recommendation*. Disponível em: <<http://www.w3.org/TR/2003/REC-soap12-part0-20030624/#L1153>>.