

CENTRO PAULA SOUZA

GOVERNO DO ESTADO DE
SÃO PAULO¹

**Faculdade de Tecnologia de Americana
Curso Superior de Tecnologia em Análise e Desenvolvimento de
Sistemas**

COMPUTAÇÃO EM NUVEM: SISTEMAS ESCALÁVEIS

FREDERICO MOREIRA DA SILVA PACHECO

**Americana, SP
2013**

**Faculdade de Tecnologia de Americana
Curso Superior de Tecnologia em Análise e Desenvolvimento de
Sistemas**

COMPUTAÇÃO EM NUVEM: SISTEMAS ESCALÁVEIS

FREDERICO MOREIRA DA SILVA PACHECO
pachecofms@gmail.com

Trabalho Monográfico, desenvolvido em cumprimento à exigência curricular do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas da Faculdade de Tecnologia de Americana, sob orientação do Prof. . Anderson Luiz Coan

Área: Computação em Nuvem

**Americana, SP
2013**

BANCA EXAMINADORA

**Professor: Anderson Luiz Coan
(Orientador)**

**Professor: Eduardo Vicentini
(Convidado)**

**Professor: Diogo Robles
(Convidado)**

AGRADECIMENTOS

Agradeço primeiramente a Deus por todas as bênçãos que me concedeu que muitas pessoas chamam de sorte ou de coincidência.

A minha mãe, Cacilda Cecilia Moreira da Silva, que sempre me apoiou em cada etapa da minha vida, ajudando, incentivando em tudo.

Ao meu orientador Professor Anderson Luiz Coan que me ajudou a prosseguir os estudos nesta área.

A todos os professores da Faculdade de Tecnologia de Americana pela contribuição na minha formação das mais diferentes maneiras.

Aos colegas da graduação pela convivência e amizade durante todo o curso.

DEDICATÓRIA

5

Dedico este trabalho a Deus, a minha mãe, pelo apoio e paciência, aos meus amigos que aqui fiz e que estarão em minhas lembranças.

RESUMO

Computação em nuvem é uma tecnologia disruptiva, que inovou a arquitetura dos centros de processamento de dados, também inovou a forma dos provedores fornecerem recursos de infraestrutura computacional aos consumidores. O conceito de nuvem permite alto desempenho de processamento, escalonamento, virtualização, além de maior acessibilidade e portabilidade das aplicações, a forma de cobrança é baseada no uso, o consumidor paga só por aquilo que consome, os serviços em nuvem são disponibilizados do mesmo modo que os serviços de utilidades públicas, como água e telefone. Este trabalho teve como objetivo demonstrar os conceitos de Computação em Nuvem, e apresentar um experimento de como utilizar Plataforma como Serviço (PaaS) para desenvolver uma aplicação escalonável hospedada em uma infraestrutura em nuvem.

Palavras Chave: Computação em Nuvem; Escalonamento; PaaS.

ABSTRACT

Cloud computing is a disruptive technology, Innovated that the architecture of data processing centers, also innovated the way providers deliver computing infrastructure resources to the consumers. The cloud concept allows high-performance processing, elasticity, virtualization, and greater accessibility and portability of applications, the way recovery is based on using the consumer pays only for what they consume, cloud services are available in the same way that the services of public utilities such as water and telephone. This study aimed to demonstrate the concepts of Cloud Computing, and present an experiment using Platform as a Service (PaaS) to develop a scalable application hosted on a cloud infrastructure.

Keywords: Cloud Computing; Elasticity; PaaS.

SUMÁRIO

INTRODUÇÃO.....	16
1.COMPUTAÇÃO EM NUVEM.....	18
1.1.HISTÓRIA.....	18
1.2.CONCEITO.....	19
1.3.ANÁLISE SWOT.....	21
1.4.MODELOS DE SERVIÇO.....	21
1.5.MODELOS DE IMPLANTAÇÃO.....	22
2.VIRTUALIZAÇÃO.....	24
2.1.HISTÓRIA.....	24
2.2.CONCEITO.....	24
2.3.MÁQUINA VIRTUAL.....	24
2.4.PROPRIEDADES.....	26
3.COMPUTAÇÃO EM GRADE.....	27
3.1.PROPRIEDADES.....	27
3.2.COMPARAÇÃO ENTRE COMPUTAÇÃO EM GRADE E COMPUTAÇÃO EM NUVEM.....	27
4.DESENVOLVIMENTO.....	28
4.1.FERRAMENTAS UTILIZADAS.....	28
4.2.PLANEJAMENTO.....	28
5.CONSIDERAÇÕES FINAIS.....	29
6.REFERÊNCIAS BIBLIOGRÁFICAS.....	30
7.GLOSSÁRIO.....	32
8.APÊNDICE A – CÓDIGO FONTE.....	33

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

ABI	<i>Application Binary Interface</i>
API	Interfaces de Programação de Aplicativos.
ARPA	Agência de Projetos e Pesquisas Avançadas.
ARPANET	Rede da Agência de Projetos e Pesquisas Avançadas.
AWS	<i>Amazon Web Services</i> .
CRM	Gestão de Relacionamento com o Cliente.
CSC	<i>Cambridge Scientific Center</i>
EC2	<i>Elastic Compute Cloud</i>
EUA	Estado Unidos da América
GAE	<i>Google App Engine</i> .
IA	Inteligência Artificial.
IaaS	<i>Infrastructure as a Service</i> .
IDE	<i>Integrated Development Environment</i> .
ISA	<i>Instruction Set Architecture</i>
NIST	<i>National Institute of Standards and Technology</i> .
NSA	<i>National Security Agency</i>
MIT	Instituto de Tecnologia de Massachusetts.
Paas	<i>Platform as a Service</i> .
S3	<i>Simple Storage Service</i> .
SaaS	<i>Software as a Service</i> .
SDK	<i>Software Development Kit</i>
SLA	<i>Service Level Agreement</i>
SOA	Arquitetura Orientada a Serviços.
SWOT	Forças, Fraquezas, Oportunidades e Ameaças.
TI	Tecnologia da Informação.
UNESCO	Organização das Nações Unidas para a Educação, a Ciência e a Cultura.
URSS	União das Repúblicas Socialistas Soviéticas.
VM	Máquina Virtual

INTRODUÇÃO

Atividades comuns do nosso cotidiano passaram a ser realizadas através da *web*, compras, transações bancárias, reservas de hotéis, pesquisas de informações, comparação de preços, leitura de livros, notícias, tudo isto é feito por meio de serviços online.

Porém estes serviços exigem uma grande capacidade computacional para que eles funcionem, necessitam de uma infraestrutura tecnologia que possibilite comportar grandes números de acessos, processar varias operações simultâneas, sem prejudicar o desempenho e acessibilidade do serviço a ser utilizado. Tal estrutura requer muito investimento em tempo, equipamentos e mão de obra especializada.

Essa infraestrutura deve ser planejada conforme a demanda do serviço que será provido, considerando sempre os índices de maior consumo da aplicação e somando uma margem de potencial crescimento. Dessa forma garantimos a estabilidade e acessibilidade ao serviço prestado, porém em períodos de baixa demanda as requisições da aplicação diminuem, mas a infraestrutura continua com o mesmo potencial de operação, o que gera processamento ocioso, mesmo não utilizando parte dos equipamentos a empresa necessita arcar com a manutenção dessas máquinas paradas, o que acaba gerando um gasto que não terá retorno.

A solução para este tipo de problema partiu da computação utilitária (*Utility Computing*), também conhecida como computação sob demanda, nela a infraestrutura de *hardware* é terceirizada, e a cobrança é feita da mesma forma que serviços de utilidade pública como água, telefone e eletricidade, onde os pagamentos são realizados com base no uso. Um provedor pode disponibilizar recursos de infraestrutura conforme as necessidades do contratante, cobrando somente pelo o que foi consumido, dessa forma o custo é reduzido pagando-se especificamente o valor correspondente ao processamento utilizado (RAINER; CEGIELSKI, 2012).

Outro conceito que auxilia essa solução é a Computação em Grade (*Grid Computing*), prevenindo possíveis falhas de acessibilidade que podem ocorrer, pois através do uso de *data centers* com distribuição geográfica de nível global e canais de comunicações interligados entre si, dá-se a possibilidade dos provedores replicar e escalonar seus dados em toda essa rede, portanto quando um servidor estiver com problemas de conexão, outro pode ser instanciado automaticamente em qualquer lugar do mundo, contendo as mesmas informações garantindo assim que o serviço seja disponibilizado de forma transparente independente da sua origem.

Com a união da computação utilitária, computação em grade e virtualização surgiram um novo paradigma chamado Computação em Nuvem (*Cloud Computing*), possuindo como principais características: ampla acessibilidade, compartilhamento de recursos, automação e serviço sob demanda.

A computação em nuvem permite escalonar de forma automática e ágil as operações de processamento, demandando um baixo nível de custo, sem despesas com mão de obra, equipamentos, alocação de servidores ou paralelização de instancias do sistema em diferentes hardwares. O escalonamento é feito conforme a demanda do serviço, a infraestrutura terá sempre capacidade de processar todas as requisições do sistema, mesmo que haja um aumento exponencial na demanda o poder computacional terá um crescimento proporcional que irá suprimir todas as necessidades do sistema.

Este trabalho monográfico se justifica pelo fato da computação em nuvem ser uma tecnologia disruptiva e responsável por grande parte da inovação, como o surgimento de novos mercados, como foi o caso dos serviços provedores de *cloud*, *Google App Engine (GAE)*, *Amazon Web Services (AWS)*, *Windows Azure*, entre outros (LAURO, 2013). A relevância desta pesquisa é demonstrar os conceitos e características deste novo paradigma arquitetônico de *software*.

O problema gerado por servidores dedicados que impossibilitam rápido escalonamento de processamento e armazenamento das aplicações acarretam em alto custo de operação, e manutenção da infraestrutura computacional, podem

ocasionar sobrecargas de rede e consequentes limitações de acesso ao serviço. Por isso é importante ter conhecimentos sobre *cloud computing*, que é uma alternativa a todos estes problemas citados.

As perguntas que se buscou responder são: Quais as vantagens e desvantagens da utilização de uma aplicação escalonável desenvolvida sobre a arquitetura de computação em nuvem? Como desenvolver um sistema de fácil acesso onde o usuário, independente de sua localização, consiga ter a disponibilidade de seus dados e informações a qualquer instante, e ainda assim conseguir economizar na implantação do software?

As Hipóteses a serem exploradas e averiguadas neste trabalho são:

1. A adoção de infraestrutura em nuvem obtém ganhos na velocidade de comunicação e capacidade de processamento.
2. Aplicações arquitetadas para computação em nuvem possuem maior acessibilidade.
3. A utilização de computação em nuvem no desenvolvimento de aplicações faz com que o escalonamento de armazenamento de dados seja possível.
4. O custo de manutenção e operação de uma infraestrutura em nuvem é mais barata que uma infraestrutura proprietária e dedicada.

O objetivo geral consistiu em demonstrar os conceitos, vantagens e desvantagens da computação em nuvem durante o desenvolvimento de um projeto escalonável, abordando alguns pontos que podem ser críticos na execução de um serviço, tais como segurança, acessibilidade, confiabilidade, custos e tempo de projeto, procurei contribuir com informações sobre essa área que ainda é algo recente mas de grande potencial, merecedora de conscientização e difusão de sua importância dentro dos sistemas e tecnologias da informação.

Os objetivos específicos foram:

1. Estudar a história e conceitos da computação em nuvem, assim como suas características, realizando um aprendizado teórico e bibliográfico sobre o assunto.

- 1 Desenvolver e demonstrar um pequeno experimento que consiste em um sistema escalonado, que terá como servidor *web* o *Google App Engine*, visando demonstrar os processos de desenvolvimento e implementação.

2. Discutir a complexidade de desenvolvimento de um sistema em nuvem, com base no estudo bibliográfico demonstrado e no experimento desenvolvido, objetivando suas vantagens de implantação e otimização.

Como metodologia para o desenvolvimento deste trabalho, foi utilizada pesquisa qualitativa aplicada, pois não possui métodos ou técnicas estatísticas como instrumento a coleta de dados, as informações pesquisadas são obtidas de fontes diretas e “objetiva gerar conhecimentos para aplicação prática dirigida à solução de problemas específicos” (SILVA; MENEZES, 2001, p. 20).

A pesquisa também terá caráter de pesquisa bibliográfica que é “[...] elaborada a partir de material já publicado, constituído principalmente de livros, artigos de periódicos e atualmente com material disponibilizado na Internet.” (SILVA; MENEZES, 2001, p.21).

No desenvolvimento da aplicação será utilizada a pesquisa experimental, visto que será analisado as vantagens e desvantagens do sistema implementado, através dos atributos que caracterizam a tecnologia em estudo, para averiguar as hipóteses experimentada. (SILVA; MENEZES, 2001, p.21).

O trabalho foi estruturado em 5 capítulos, sendo que o primeiro conceitua a computação em nuvem, apresenta-se uma introdução com levantamento histórico, uma análise de SWOT comparando arquiteturas *cloud* com arquiteturas *on-premise*, e classifica-se os modelos de serviço e de implantação desse paradigma, o segundo traz uma breve abordagem sobre virtualização uma das tecnologias que servem como base para a computação, neste capítulo apresenta-se uma introdução, a

história e evolução da tecnologia apresentada, define-se o conceito de virtualização e máquina virtual e, por fim, identifica-se suas propriedades, o terceiro descreve a computação em grade, outra tecnologia que compõe a computação em nuvem, este capítulo também define o conceito da tecnologia, e por fim faz um comparativo entre a computação em grade e a computação em nuvem, o quarto demonstra o experimento desenvolvido com objetivo de evidenciar a característica de escalonamento da computação em nuvem.

Com base nas informações conseguidas a partir dos estudos realizados no capítulo anterior, o quinto capítulo se reserva às Considerações Finais.

1 COMPUTAÇÃO EM NUVEM

A Computação em Nuvem (*Cloud Computing*) nasceu do processo evolutivo de compartilhamento e maior otimização dos recursos tecnológicos na área da informação, essa inovação surgiu do acúmulo de ideias e tecnologias como Virtualização, Computação em Grade (*Grid Computing*) e Interfaces de Programação de Aplicativos (API) Assíncronas (TAURION, 2009).

Em 2011 cerca de 74% das empresas já utilizavam algum tipo de serviço em nuvem, como um simples e-mail corporativo ou um sistema hospedado em *cloud*, o mercado movimentou nesse ano uma quantia de aproximadamente 40 bilhões de dólares. Segundo a *Forrester Research*, uma empresa de pesquisa e consultoria global em Tecnologia da Informação, no ano de 2020 este será um mercado de 241 bilhões de dólares (LAURO, 2013).

Existe uma evolução na migração dos serviços para a arquitetura em nuvem demonstrada na figura 1.

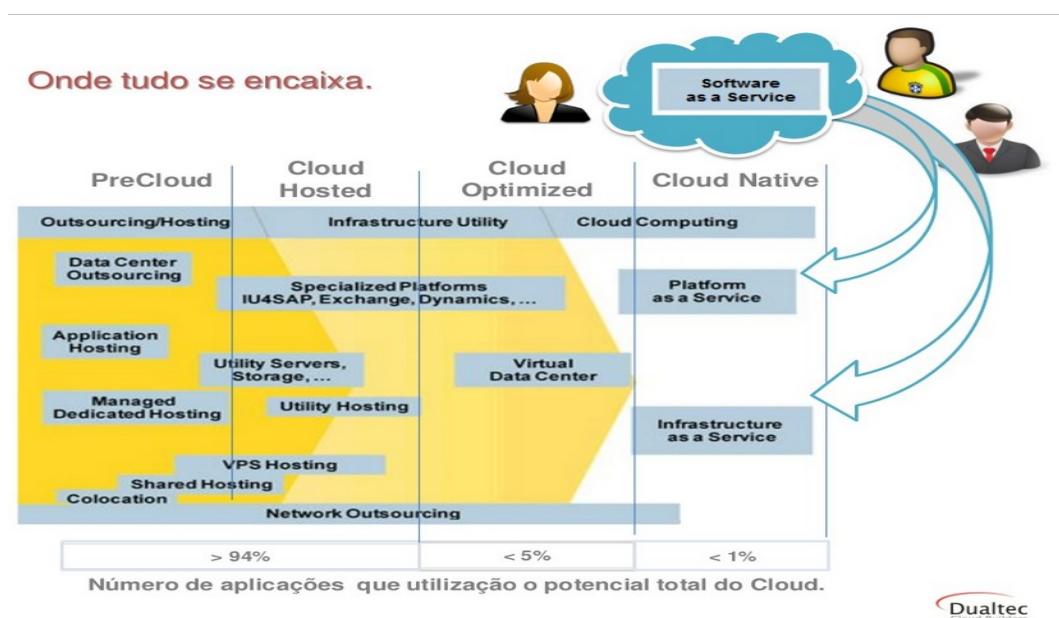


Figura 1 – História da Computação em Nuvem

Fonte: <http://pt.slideshare.net/Lauro2/cloud-computing-e-saa-s-a-importancia-dessas-ferramentas-no-seu-negocio-rev20>

Os *softwares* desenvolvidos a dez ou quinze anos atrás não possuem a capacidade de consumir os benefícios da infraestrutura *cloud*, estas aplicações monolíticas, desenvolvidas para atender uma funcionalidade em específico, limitando-se a uma infraestrutura pré-determinada, são categorizadas como “*pre cloud*”. Os *softwares* hospedados dentro da arquitetura *cloud*, mas que não possuem a capacidade de escalonamento são chamados de “*cloud hosted*”, normalmente são sistemas legados migrados para *cloud*. As aplicações que são escalonadas por meio de comandos, ou solicitações do cliente ao provedor são categorizadas como “*cloud optimized*”. Já os serviços auto-escalonáveis como Netflix, por exemplo, escalonam seus recursos automaticamente sem nenhuma interação humana, a própria aplicação gerencia sua capacidade de processamento e armazenamento, esse é tipo de aplicação “*cloud native*” (LAURO, 2013).

Para uma empresa adotar a computação em nuvem de modo corporativo é preciso antes levantar suas reais necessidades, ver como este modelo vai apoiar os processos da organização, se os custos de implementação e implantação serão viáveis e condizentes com o orçamento empresarial (LAURO, 2013).

A conectividade é essencial para garantir a funcionalidade do serviço, para prevenção de possíveis quedas da aplicação é preciso planejar contingências, pensar em opções de conexões: internet a cabo, via satélite, radio ou 3G. A conexão de contingência precisa ter a mesma qualidade e velocidade da conexão principal? Quanto tempo o sistema pode ficar *offline*? Em ultimo caso, tem como realizar os processos de forma manual ou *offline*? Estes são pontos importantes a serem levantados e estudados antes de adotar o modelo de computação em nuvem, outro fator importante é ter uma estratégia de saída, ou seja, caso a aplicação tenha que ser retirada da nuvem é preciso saber como proceder e qual alternativa recorrer, enfim nunca se deve ficar refém da nuvem (LAURO, 2013).

1.1. HISTÓRIA

Em 1961 John McCarthy (pioneiro em Inteligência Artificial), sugeriu durante o discurso do centenário do Instituto de Tecnologia de Massachusetts (MIT), que

algum dia a computação seria disponível como um serviço de utilidade pública. Conceituando desta forma o que se tornaria *cloud computing* (GARFINKEL, 1999).

Na época o mundo sofria de uma polarização ideológica, uma espécie de maniqueísmo entre os Estados Unidos da América (EUA) e a União das Repúblicas Socialistas Soviéticas (URSS), era os tempos áureos da Guerra Fria, em 1962 a Agência de Projetos e Pesquisas Avançadas (ARPA), do Departamento de Defesa dos EUA, contratou Joseph Carl Robnett Licklider para liderar as pesquisas sobre (SIMON, 1997):

“[...] uma rede de computadores que permitisse o trabalho cooperativo em grupos, mesmo que fossem integrados por pessoas geograficamente distantes, além de permitir o compartilhamento de recursos escassos, [...]” (SIMON, 1997)

O resultado das pesquisas de Licklider, futuramente iria realizar o que McCarthy tinha sugerido, o projeto foi amadurecendo e após oito anos de pesquisas, em janeiro de 1970, utilizando linhas telefônicas para interligar uma rede computacional, os pesquisadores conectaram quatro universidades norte-americanas, Universidade da Califórnia em *Los Angeles*, *Stanford Research Institute*, Universidade da Califórnia em Santa Bárbara e a Universidade de Utah, criando-se assim a “*Advanced Research Projects Agency Network*”, ARPANET, essa conexão de rede foi o que deu origem à internet, possibilitando o tráfego dos dados de maneira descentralizada (SIMON, 1997).

Em março de 1999, já com o advento da Virtualização e da Computação em grade, a empresa americana *Salesforce* lança a primeira aplicação com serviços *online* a ser fornecida por um site, com essa iniciativa o site *salesforce.com* torna-se pioneiro em sistemas de uso empresarial utilizando Arquitetura de *software* Orientada à Serviço (SOA), o *software* de Gestão de Relacionamento com o Cliente – *Customer Relationship Management* (CRM), da empresa passou a ser disponibilizada através da internet (MOHAMED, 2009).

O próximo marco da história da computação em nuvem ocorre em 2002, quando a *Amazon* lança seu provedor de arquitetura *cloud*, *Amazon Web Services*,

fornecendo serviços de processamento e armazenamento computacional, além do serviço de Inteligência Artificial (IA) “*Amazon Mechanical Turk*” que facilitou o acesso a IA, tornando-o simples, escalonável e econômico. (MOHAMED, 2009; AMAZON, 2013)

Após quatro anos, o *Amazon Web Services* (AWS) lança dois novos serviços, *Elastic Compute Cloud* (EC2) e *Simple Storage Service* (S3), o primeiro serviço oferece a capacidade de redimensionar os recursos computacionais na nuvem, em questão de minutos é possível escalar sua capacidade computacional conforme as necessidades vão se alterando, o segundo serviço facilita o armazenamento de dados, maximiza os benefícios de escalonamento e permite trafegar qualquer quantidade de dados a qualquer momento (MOHAMED, 2009; AMAZON, 2013).

Os serviços fornecidos pelo AWS, assim como os serviços fornecidos pelos demais provedores de arquitetura *cloud*, são tarifados conforme o modelo de cobrança *pay-as-you-go* (pago pelo uso), neste modelo o cliente tem o uso ilimitado de recursos e pagará somente pela quantidade de recursos utilizados, durante um período específico pré-determinado pelo contrato realizado com o provedor. Essa tarifação é bem semelhante a cobrança de serviços de utilidade pública como água, telefone e eletricidade, onde os pagamentos são realizados com base no uso mensal (TAURION, 2009).

O Google anunciou no seu blog oficial, em julho de 2009, que seu pacote de aplicações para produtividade executado na nuvem, o *Google Apps*, estava saindo da versão beta e passando para versões *enterprise* e *consumer*. O *Google Apps* surgiu em 2006 tendo inicialmente somente o *Gmail* como aplicação em nuvem, com o passar do tempo a gigante da internet foi criando e adicionando novas aplicações ao *Apps*, como o *Google Calendar*, *Google Talk* (atual *Hangout*), *Google Docs*, *Google Sites* e *Google Drive* (GOOGLE, 2009).

Segundo Dan Germain, diretor de tecnologia da *Cobweb Solutions* provedora de serviços de TI, a contribuição mais importante para a computação em nuvem tem sido o surgimento de “*killer apps*” das principais gigantes de tecnologia, como

Microsoft e Google. Quando essas empresas oferecem serviços de uma forma que seja confiável e fácil de consumir, o efeito em cadeia para a indústria como um todo é uma aceitação geral mais ampla dos serviços on-line fornecidos pela computação em nuvem. (MOHAMED, 2009)

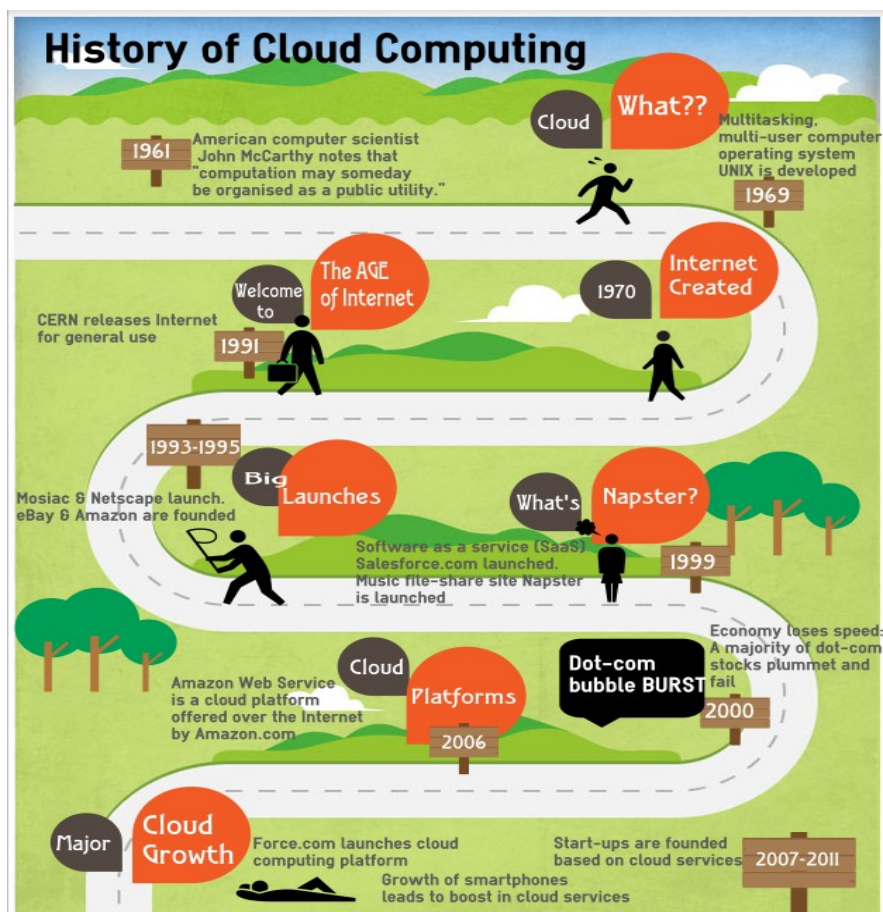


Figura 2 – História da Computação em Nuvem
Fonte: <http://zeraya.com/celebrating-founders-day/>

1.2. CONCEITO

Embora o conceito de computação em nuvem tenha começado a ser definido em 1960, este ainda é um conceito relativamente novo, suas definições continuam sofrendo inúmeras evoluções devido a grande onda de inovações que ocorrem em torno dele (ZUFFO et. al., 2013). O *National Institute of Standards and Technology* (NIST), norte-americano, atribui a seguinte definição para *cloud computing*: A computação em nuvem é um modelo para permitir de forma ubíqua e conveniente, o acesso à rede sob demanda, compartilhando um conjunto de recursos

computacionais configuráveis, por exemplo, redes, servidores, capacidade de armazenamento, aplicativos e serviços que podem ser rapidamente provisionados e liberados com mínimo esforço de gestão ou interação com o prestador de serviços (NIST, 2011).

A rede sob demanda e o compartilhamento de recursos, permite diferentemente dos modelos tradicionais onde a infraestrutura é proprietária e dedicada, a utilização do tempo ocioso de máquinas independentes, em diferentes lugares geográficos, sem a necessidade de investir em mais equipamentos ou espaço para alocar novos *hardwares*, gerando assim maior eficiência, maior adequação da estrutura computacional devido a flexibilidade das redes *on demand*, além dos baixos custos quando comparado com centros de processamentos de dados (CPD) particulares (TAURION, 2009).

O termo nuvem surgiu historicamente como uma metáfora a todas as tecnologias que representavam a internet, em diagramas de rede é comum a representação de uma nuvem para abstrair a complexidade que a infraestrutura *web* acaba englobando, as linhas do diagrama que chegam até a nuvem significam simplesmente que os dados trafegam pela internet (TAURION, 2009).

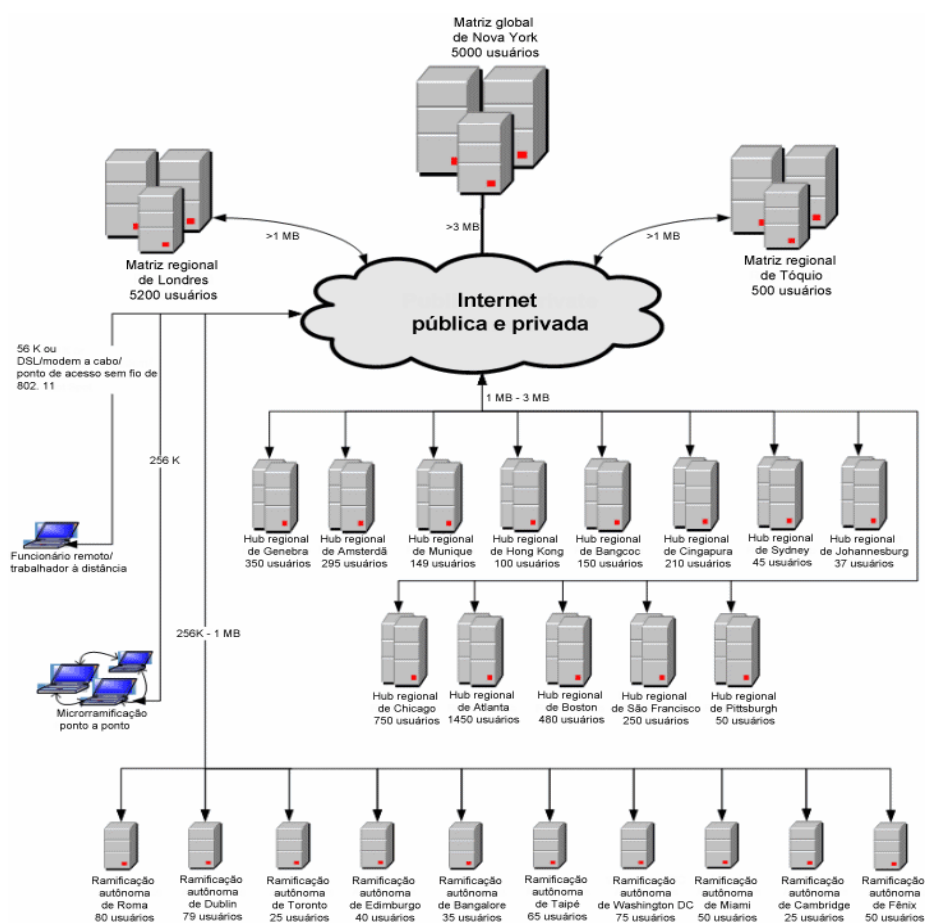


Figura 3 – Diagrama de redes

Fonte: <http://technet.microsoft.com/pt-br/library/>

A computação em nuvem tem como um dos seus objetivos ser ubíqua, exatamente como a representação da nuvem, onde não temos uma ideia de qual é a origem dos dados e não temos consciência do nível de complexidade e diversidade, da malha de redes em que as informações tiveram que trafegar até chegar na aplicação que usuário está utilizando. Essa abstração somada com a capacidade de virtualização, escalonamento, coesão e acoplamento, dá a ilusão que os recursos computacionais da nuvem são infinitos (TAURION, 2009).

A gestão de infraestrutura tradicional acabou se revelando desadequada e proibitiva, hoje em dia geramos petabytes de informações por dia, zettabytes passa a ser uma quantidade real e não mais futurística ou imaginária para um futuro bem próximo (IBM, 2012), “os custos de energia elétrica aumentaram pelo menos oito vezes de 1996 até hoje”(TAURION, 2009), dessa maneira fica claro que não é possível aumentar a capacidade de um CPD para comportar toda a quantidade de processamento, mantendo o modelo de métodos atuais de gestão.

Um novo modelo de gestão, *utility computing*, foi conceituado como um grande conjunto de recursos computacionais, que oculta todas as dificuldades de gestão, criação e manutenção de uma infraestrutura computacional, este serviço é fornecido de maneira customizável ao cliente, é possível escolher os recursos a serem utilizados conforme a necessidade do sistema, o provedor é auto-organizado capaz de aumentar ou diminuir o número de recursos automaticamente a fim de evitar a ociosidade ou sobrecarga da aplicação. (TAURION, 2009)

Seguindo a definição do NIST para *cloud computing*, temos cinco características essenciais da computação em nuvem elencadas pela mesma instituição:

Serviço sob demanda.

O consumidor pode provisionar unilateralmente recursos de capacidades computacionais, como tempo de processamento e armazenamento em rede conforme necessário, tudo isto é feito automaticamente sem que haja a necessidade de interação humana com cada prestador de serviço (NIST, 2011).

“Essa característica da computação em nuvem oferece grande conveniência aos usuários, que podem usar os serviços a partir de qualquer lugar. O agrupamento dos recursos computacionais traz enormes vantagens em gestão do patrimônio, otimização de gastos e diminuição de esforço humano em atividades de manutenção de equipamentos físicos.” (ZUFFO et. al., 2013)

Amplo acesso à rede.

Os recursos computacionais devem estar disponíveis através da rede, possibilitando acesso por meio de mecanismos padronizados que promovam o uso por meio de plataformas heterogêneas de consumo, os dispositivos de conexão a rede, como celulares, *tablets*, *notebooks* e *desktops* podem ter recursos robustos ou escassos, porém devem ter a mesma qualidade de acesso (NIST, 2011). A rede deve garantir interoperabilidade, consistência, disponibilidade, não repúdio e segurança dos serviços oferecidos (ZUFFO et. al., 2013).

“Cabe ressaltar que a existência de uma rede com essas características é condição essencial para a computação em nuvem. A interoperabilidade é a garantia de que qualquer dispositivo pode ser interligado à rede e de que vai funcionar. O não repúdio está associado à identificação do usuário, garantindo, de forma legal, o reconhecimento de sua existência íntegra e seu acesso à nuvem.” (ZUFFO et. al., 2013)

Agrupamento de recursos.

Os equipamentos de infraestrutura do provedor são agrupados para atender vários consumidores através do modelo de multi-clientes, possuindo diferentes recursos físicos e virtuais, dinamicamente atribuídos de acordo com a demanda do consumidor (NIST, 2011).

Há uma sensação de independência da localização, em que o cliente geralmente não tem controle ou conhecimento sobre a exata localização dos recursos computacionais fornecidos, porém existe a capacidade de se especificar a localização de um nível mais elevado de abstração, como o país, estado ou o *data center* onde está localizado o equipamento utilizado pelo cliente (NIST, 2011). O Instituto ainda apresenta exemplos de recursos que agrupam capacidades de armazenamento, processamento, memória e largura de banda de rede.

Elasticidade rápida.

Os recursos devem ser elasticamente provisionados e liberados, ou seja, precisa ter a capacidade de aumentar e reduzir seu volume conforme a latência de processamento e armazenamento, em alguns casos este fornecimento deve ser automático para que seja possível escalonar a aplicação rapidamente de forma extrínseca e intrínseca conforme a demanda (NIST, 2011).

Para o consumidor, a capacidade de provisionamento de recursos computacionais deve parecer como algo ilimitado, podendo ser apropriado ou ajustado a qualquer momento independente da quantidade solicitada (NIST, 2011).

Serviço medido.

Sistemas em nuvem controlam e otimizam o uso dos recursos, aproveitando uma capacidade de medição de algum nível de abstração apropriado para o tipo de serviço, por exemplo, armazenamento, processamento, largura de banda e contas de usuários ativos. O uso de recursos pode ser monitorado, controlado e reportado, oferecendo transparência tanto para o provedor ou consumidor do serviço utilizado (NIST, 2011).

“A característica de serviço medido é fundamental para a avaliação de uso da infraestrutura da computação em nuvem e para a gestão e a alocação apropriadas de seus recursos à comunidade de usuários.” (ZUFFO et. al., 2013)

1.3. ANÁLISE SWOT

A análise de Forças, Fraquezas, Oportunidades e Ameaças - *Strenghts, Weaknesses, Opportunities and Threats* (SWOT) é um método simples e transparente utilizado pela área de gestão para auxiliar no planejamento estratégico, através dessa ferramenta podemos destacar os principais pontos funcionais e competitivos do modelo computacional em estudo (TONINI, 2007).

Nesta seção vou apresentar algumas análises de SWOT, comparando dois tipos de infraestrutura com diferentes arquiteturas computacionais, a análise consiste em comparar as mudanças entre as forças, fraquezas, oportunidades e ameaças dos modelos arquitetônicos conforme a evolução do tempo.

Também terei a intenção de demonstrar que a computação em nuvem surgiu como uma tecnologia disruptiva, visto que esse modelo de arquitetura supera o desempenho de infraestruturas *on-premise*, rompendo os limites de capacidade dos centros de processamentos de dados em absorver a demanda de *performance* de processamento solicitada pelo mercado (KRIKOS, 2010).

Uma das diferenças entre *cloud computing* e *on-premisse*, é que em computação em nuvem não precisa comprar licença de *softwares* proprietários, a licença dessas aplicações são de responsabilidade do provedor de hospedagem

cloud, como os servidores são compartilhados, replicados e possuem instâncias dinâmicas a licença desses softwares são fornecidas ao provedor que compartilha essa licença com os clientes (KRIKOS, 2010).

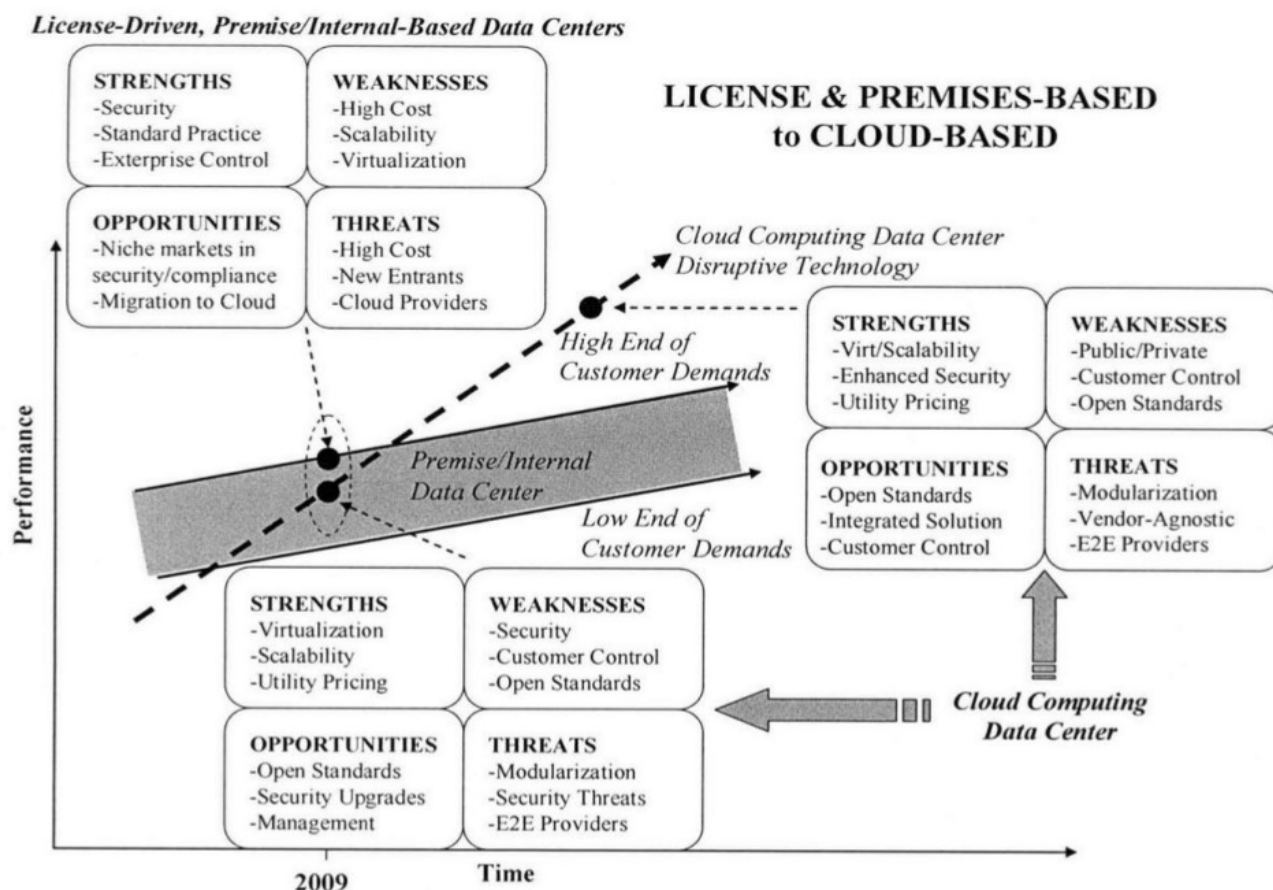


Figura 4 – Análise de SWOT – Cloud Computing
Fonte: <http://dspace.mit.edu/handle/1721.1/59255>

Na figura 4 é apresentada uma análise de SWOT conforme o surgimento da computação em nuvem em relação aos *data centers on-premise*, centros de processamentos de dados locais e privados que não suportam escalonamento. Outro aspecto considerado no diagrama é a capacidade de desempenho de cada ambiente conforme sua arquitetura (KRIKOS, 2010).

A faixa sombreada da figura demonstra os limites da capacidade de *data centers on-premise* utilizados pelo mercado, em assimilar uma dada tecnologia

conforme o aumento ou redução da demanda de consumo dos serviços prestados (KRIKOS, 2010).

Essa análise de SWOT aponta como pontos fortes da tecnologia *on-premise* a segurança e práticas normalizadas, enquanto os custos, capacidade de virtualização e escalonamento seriam suas fraquezas. É importante notar que inicialmente para a computação em nuvem, a segurança era uma fraqueza que deixou de existir com o passar do tempo, porém a privacidade dos dados surgiu como um ponto fraco (KRIKOS, 2010).

O provedor que hospeda os dados em nuvem deve garantir medidas adequadas de segurança, as normalizações e procedimentos de segurança dos dados devem estar especificados no contrato realizado com o provedor, bem como medidas de contingência, capacidade de recuperação dos dados através de *backup*, localização dos *data centers* e Acordos de Nível de Serviço - *Service Level Agreement* (SLA) que asseguram a confiabilidade, qualidade e segurança do serviço prestado (WINKLER, 2012).

Embora os problemas de segurança estejam resolvidos, a privacidade é um ponto que traz incerteza para que alguns adotem a tecnologia em nuvem, os casos recentes relacionados a espionagem virtual da Agência Nacional de Segurança - *National Security Agency* (NSA) do Estados Unidos, levantam questionamento sobre abusos e violações sobre as informações hospedadas por empresas terceirizadas e de diferente jurisdição do cliente, por isso é aconselhável utilizar provedores locais sob legislação local, desta forma é mais fácil resolver possíveis problemas jurídicos do que quando hospedados em servidores internacionais (CASCÃO, 2013).

A análise representa os servidores *on-premise* no topo da faixa de assimilação do mercado, onde rupturas tecnológicas tornam-se uma necessidade. Toda a infraestrutura *on-premise* é competitivamente vulnerável a computação em nuvem, pois a *cloud computing* surgiu como uma tecnologia disruptiva, capaz de suprimir as necessidades que estavam fora dos limites do modelo *on-premise*, oferecendo melhor custo, conveniência e escalabilidade (KRIKOS, 2010).

1.4. MODELOS DE SERVIÇO

A computação em nuvem tem suas origens como um modelo de produção, e não como um modelo de consumo, desde o seu início, a computação em nuvem foi concebida com um meio para fornecer aplicações de software à usuários finais. Como parte da implantação do software e mecanismo de entrega, a computação em nuvem foi projetada para trazer eficiência operacional para prestadores de serviços, não aos consumidores, seguindo esta linha, arquitetos de computação em nuvem criaram a “cloud” como um modelo de produção para gerenciar centros de dados (IaaS), criaram o modelo de plataformas de desenvolvimento de aplicações e serviço (PaaS), e para fornecer um veículo de implantação de aplicativos e serviços (SaaS) (KRIKOS, 2010).

A imagem abaixo demonstra a hierarquia dos modelos IaaS, Paas e SaaS, e a visibilidade do usuário conforme cada camada de serviço.

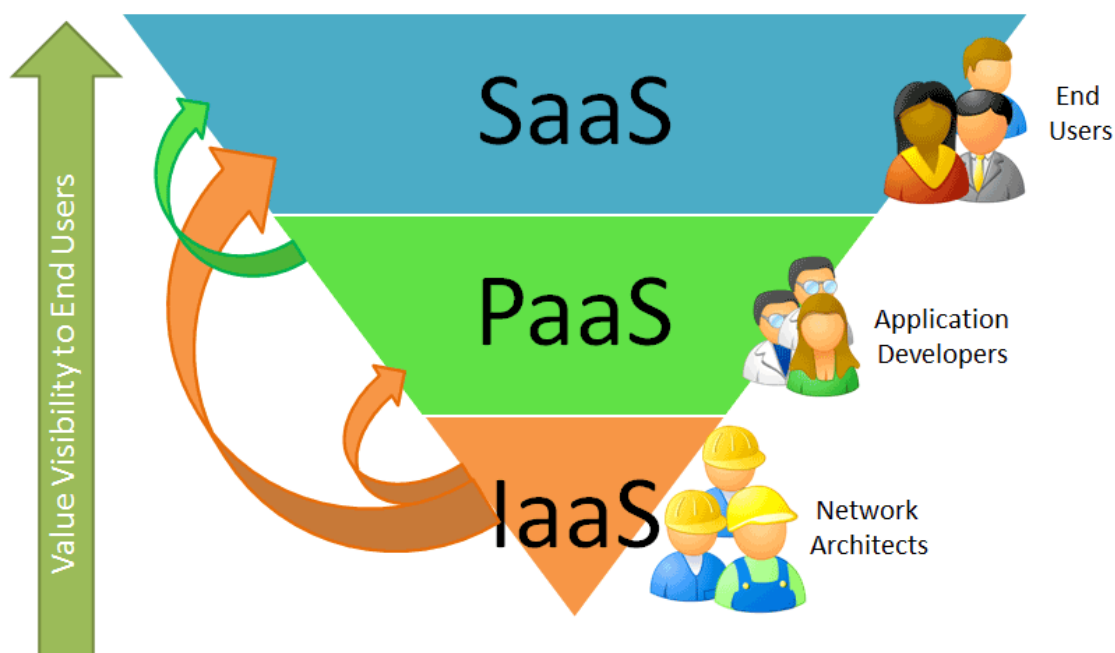


Figura 5 – Modelos de Serviço – Cloud Computing
Fonte: <http://wporfirio.files.wordpress.com/2013/05/>

Infraestrutura como serviço – Infrastructure as a Service (IaaS)

A base da pirâmide é composta pela camada de infraestrutura, IaaS, essa camada contém grande quantidade de redes interligando enormes *data centers*, destinados ao compartilhamento dos recursos computacionais. A IaaS é um elemento essencial na pirâmide da computação em nuvem, devido a sua capacidade de fornecer recursos tecnológicos de maneira flexível e de enorme capacidade computacional. (KRIKOS, 2010)

A capacidade fornecida ao consumidor é a de provisão de processamento, armazenamento, redes e outros recursos fundamentais de computação onde o consumidor é capaz de implementar e executar softwares arbitrários, que podem incluir sistemas operacionais e aplicativos. Neste modelo o consumidor não gerencia ou controla a infraestrutura de nuvem subjacente, porém tem controle sobre sistemas operacionais, armazenamento, aplicativos instalados, controle limitado a *firewalls* e outros componentes de rede. (NIST, 2011)

Com isto a infraestrutura como serviço se tornou fundamental para grandes, médias e pequenas empresas, visto que seu custo é menor que uma infraestrutura local. Normalmente estas redes são hospedadas por provedores terceirizados, que podem ser um fornecedor local, regional ou internacional, essa camada possui duas formas principais de implementação, a primeira é através de servidores e equipamentos dedicados, a segunda maneira é através de servidores multi-clientes compartilhados, e hospedados em provedores terceirizados (KRIKOS, 2010).

Plataforma como Serviço – Platform as a Service (PaaS)

No meio da pirâmide está a camada de plataforma para desenvolvimento, PaaS, Muitos analistas afirmam que a plataforma como serviço é camada por onde a computação em nuvem será mais disseminada. A PaaS permite que uma determinada aplicação funcione de maneira correta, quando inserida em um ambiente específico de hospedagem em nuvem, por meio de protocolos

proprietários, *middlewares*, sistemas operacionais e outras ferramentas que auxiliam no desenvolvimento de *softwares* específicos para essa plataforma (KRIKOS, 2010).

Com o advento do modelo de Plataforma como Serviço, os consumidores puderam implantar suas aplicações dentro de infraestruturas em nuvem, configurar o ambiente de hospedagem *cloud*, gerenciar os aplicativos implementados, arrendar mais espaço para armazenamento ou capacidade para processamento, enfim a plataforma fornece serviços para criar, executar e operar aplicações, incluindo ferramentas de administração e gestão, mecanismo de gerenciamento de dados. (KRIKOS, 2010; NIST, 2011).

Geralmente os provedores de PaaS oferecem cotas gratuitas ou um período de testes, sem custos para o desenvolvedor construir, testar e implantar seus *softwares*. Uma vantagem competitiva dos provedores de plataforma de desenvolvimento é ter um rico conjunto de aplicações e serviços, e que acabam atraindo os programadores para esta nova tecnologia (KRIKOS, 2010; NIST, 2011).

Existem vantagens e desvantagens em relação à adoção de PaaS no ambiente de desenvolvimento de um cliente, ao mesmo tempo em que este modelo oferece maior desempenho, flexibilidade de recursos, segurança, custo e controle, os provedores possuem restrições (*vendor lock-in*), que as vezes acabam limitando o desenvolvimento, este é um dos pontos que podem ser melhorados para que a Plataforma com Serviço se torne mais atraente ao mercado, outro ponto de melhoria seria a interoperabilidade entre diferentes redes de PaaS e entre provedores PaaS e soluções *on-premise*. É provável que o modelo de Plataforma como Serviço passe por uma transformação, utilizando padrões abertos para remediar *vendor lock-in* e tornando as aplicações acessíveis, independentemente da plataforma de desenvolvimento do provedor (KRIKOS, 2010).

Software como Serviço – Software as a Service (SaaS)

No topo da pirâmide, temos a camada de aplicação, SaaS, onde são executadas aplicações especialmente desenvolvidas para a arquitetura em nuvem. *Software* como Serviço é caracterizado por sistemas desenvolvidos por PaaS, e

hospedados em provedores com IaaS. *Softwares* orientados a consumidores ou negócios são acessados nesta mesma camada a partir de diversos dispositivos computacionais através de uma conexão *web*, um exemplo de SaaS bastante utilizado hoje em dia são os serviços de e-mails (KRIKOS, 2010; NIST, 2011)).

Os *killers apps*, da computação em nuvem surgiram nesta camada, tais como o CRM da Salesforce, DropBox, serviços do Google como *Docs*, *Drive* e *Calendar*, por esse motivo o Software como Serviço tem sido o foco central da computação em nuvem (KRIKOS, 2010).

A figura 6 demonstra as camadas de gerenciamento que usuário tem conforme o modelo de serviço:

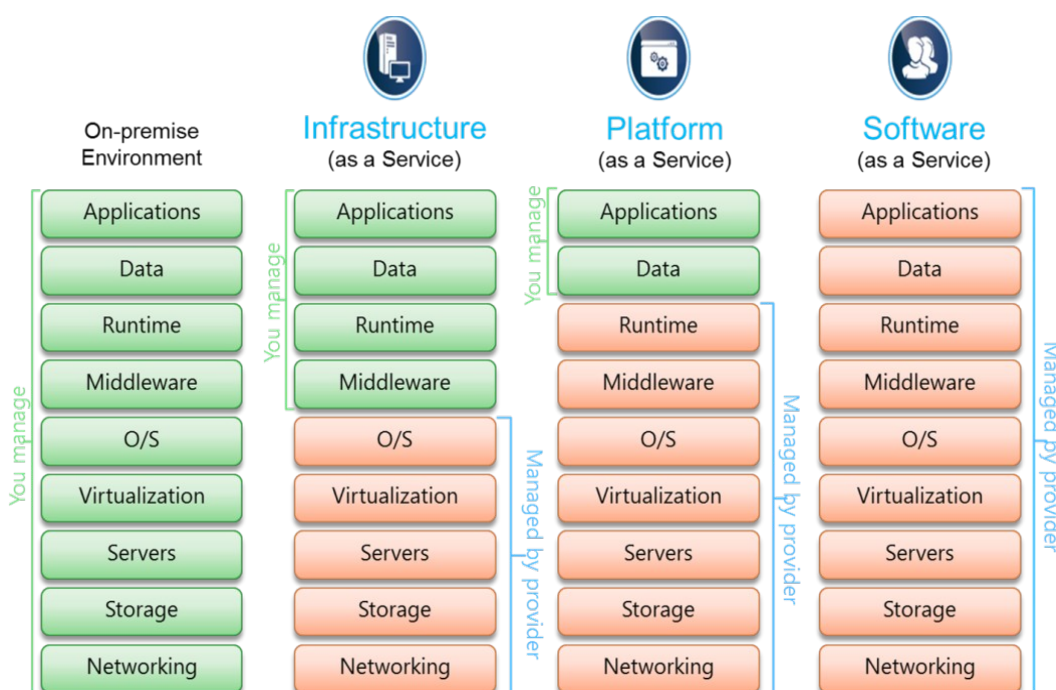


Figura 6 – Gerenciamento dos Modelos de Serviço
 Fonte: <http://wely-lau.net/2012/03/>

1.5. MODELOS DE IMPLANTAÇÃO

Nuvem Privada

A infraestrutura da nuvem privada é disponibilizada para uso exclusivo de uma única organização compreendendo vários consumidores, ela pode ser de propriedade, gerenciamento e operação, da própria organização, de terceiros, ou uma combinação mista entre a terceirização e propriedade dessa rede, as instalações da infraestrutura podem ser interna a empresa ou em ambientes externos (NIST, 2011).

A nuvem privada é um modelo de implantação utilizado quando os dados são muito sensíveis ou valiosos para serem disponibilizados publicamente na internet, nesse caso as empresas detêm a infraestrutura com o objetivo de controlar o nível de acesso e segurança da informação, neste modelo de nuvem o cliente tem total controle sobre o gerenciamento dos recursos, podendo configurar, alocar e distribuir os equipamentos como desejar, a implantação, manutenção e integração dos data centers ficam por conta da empresa proprietária (CASTRO, 2009).

Nuvem Comunitária

A infraestrutura da nuvem comunitária é compartilhada entre consumidores que partilham dos mesmos interesses, por exemplo, política, estudos, esporte e saúde. A administração e manutenção da nuvem também são compartilhadas entre empresas e organizações de comuns interesses, como instituições de ensino, empresas de mesmo setor e associações, o gerenciamento deste modelo se assemelha ao de nuvem privada, a política de acesso e configurações de rede são definidas pelas organizações que administram a infraestrutura (NIST, 2011).

Nuvem Pública

A infraestrutura da nuvem pública é fornecida para o uso aberto ao público em geral, podendo ser mantida por organizações governamentais, acadêmicas,

empresariais ou algum tipo de combinação entre esses tipos de instituições, (NIST, 2011).

Nuvens públicas são em sua maioria administradas por grandes provedores de serviços em nuvem, neste modelo os recursos de infraestrutura são compartilhados de maneira flexível, executando diferentes aplicações de diferentes clientes simultaneamente, porém virtualmente isolados, dando acesso ao cliente somente sobre as aplicações que ele possui permissão de utilização, a adoção e definição das políticas de acesso são estabelecidas pelas organizações que gerenciam a nuvem. Devido ao compartilhamento de *data centers* a nuvem pública possui uma estrutura muito superior comparada com a nuvem privada, como a nuvem pública é aberta a todo o público sua demanda é muito maior que nos outros modelos de implantação, o que torna sua maior capacidade computacional uma necessidade para este modelo (CASTRO, 2009).

Nuvem Híbrida

A infraestrutura da nuvem híbrida é uma composição de dois ou mais modelos distintos de implantação em nuvem (privada, comunitária ou pública), a infraestrutura deve pertencer a uma única entidade e possuir interoperabilidade entre os modelos adotados na composição da nuvem híbrida, este tipo de modelo pode ser utilizado para aumentar o poder de escalonamento de uma nuvem privada, frente a um pico ou sazonalidade de demanda não prevista de uma organização, possibilitando o balanceamento entre os diferentes modelos de implantação conforme a necessidade dos recursos. (CASTRO, 2009, NIST, 2011).

Outro exemplo de utilização da nuvem híbrida é utilizar nuvens públicas para realizar atividades periódicas da nuvem privada que representam um elevado consumo de processamento, como *backups*, por exemplo, que quando realizados por recursos externos liberam a capacidade de armazenamento e processamento da rede (CASTRO, 2009).

2. VIRTUALIZAÇÃO

Um computador é composto por várias peças eletrônicas que fornecem funcionalidades que são operadas, pelo sistema operacional e suas aplicações, basicamente uma máquina física é formada por dois componentes, *hardware* e *software*, no segundo componente podemos classificar o sistema operacional (SO) como o principal software de um computador, o SO controla o hardware através de chamadas do sistema que são realizadas por meio do processador que fornece instruções para alocação de memória, gerenciamento de entrada e saída dos dados e estruturação para armazenamento de dados, cabendo ao hardware executar tais operações (LAUREANO et. al., 2009).

O sistema operacional tem como um dos seus objetivos oferecer uma visão de alto nível, sobre o as solicitações e gerenciamento de *hardware* realizado pelas aplicações em execução, sendo assim caracterizado pelo nível elevado de abstração e interfaces de comunicação muito bem definidas (LAUREANO et. al., 2009).

“As abstrações oferecidas pelo sistema às aplicações são construídas de forma incremental, em níveis separados por interfaces bem definidas e relativamente padronizadas. Cada interface encapsula as abstrações dos níveis inferiores, permitindo assim o desenvolvimento independente dos vários níveis, o que simplifica a construção e evolução dos sistemas.” (LAUREANO et. al., 2009).

Para um software ser executado, é necessário que ele tenha sido desenvolvido seguindo o modelo do conjunto de instruções, utilizado pelo processador do ambiente em que será rodado, ou seja, é preciso respeitar a Arquitetura do Conjunto de Instruções - *Instruction Set Architecture*. (ISA). Além de da ISA o software também precisa respeitar o conjunto de chamadas utilizadas pelo sistema operacional, juntando estas duas primícias temos a Interface Binária de Aplicação - *Application Binary Interface* (ABI) (LAUREANO et. al., 2009).

Embora estas duas interfaces tenham sido criadas para facilitar o desenvolvimento de sistemas de forma independente, observou-se que eram pouco flexíveis, para que um sistema operacional possa trocar informações com o

hardware, é necessário que ele siga os padrões ISA referente ao hardware em que será implantado, uma biblioteca só é executada sobre a ABI para qual foi arquitetada, e uma aplicação precisa respeitar a ABIs/APIs pré-definidas. Essa falta de flexibilidade de interação entre o relacionamento das interfaces, levou ao estudo de uma nova abordagem capaz de facilitar essa integração, uma das técnicas mais promissoras é a virtualização.

2.1. HISTÓRIA

Dois anos antes de John McCarthy sugerir o que se tornaria *cloud computing*, em junho 1959 na cidade de Nova York, na UNESCO, Christopher Strachey publicou durante a Conferência Internacional de Processamento da Informação, o artigo “*Time Sharing Processing in Large Fast Computers*”, que iniciaria o conceito de virtualização, o artigo tratava sobre o tempo de compartilhamento - *time sharing* no uso de processamento fragmentado (multiprogramação), o que possibilitava virtualizar múltiplas aplicações (VERAS, 2011).

Essa publicação deu origem ao processamento multitarefa, técnica que possibilita a execução de duas ou mais aplicações ao mesmo tempo, na época os equipamentos de *hardwares* existentes não tinham a capacidade de realizar tarefas simultâneas como agora nos tempos atuais, os servidores e *data centers* tinham grandes limitações de memória e processamento devido as tecnologias utilizadas naquele momento (POLLON, 2008).

Logo no início dos anos 60, o Departamento de Engenharia Elétrica da Universidade de Manchester, cria o computador mais rápido da época, o primeiro supercomputador a utilizar os conceitos de *time sharing*, multiprogramação e controle de periféricos compartilhados, o Atlas foi responsável pelo surgimento dos conceitos de memória virtual, paginação por demanda e armazenamento de nível único. Em sua arquitetura o Atlas separava os processos de gerenciamento dos recursos essenciais do sistema operacional, dos processos responsáveis pela execução das demais aplicações do computador, assim nascia monitor de máquinas virtuais (WILLIAMS, 2007).

Ainda na década de 60 utilizando-se do conceito de *time sharing*, o centro de pesquisas da IBM utilizou seu projeto M44/44x como um caso teste para a viabilização desse novo conceito, o compartilhamento do tempo de processamento das máquinas computacionais. A arquitetura do projeto foi planejada para a criação de uma rede de máquinas virtuais (VMs), sendo a máquina principal do modelo IBM 7044 (M44), e as demais máquinas da rede eram uma imagem virtualizada da máquina principal, essas máquinas virtuais foram chamadas de 44X, por isso o nome de M44/44x do projeto. Utilizando técnicas de memória virtual e multiprogramação o espaço de endereçamento das VMs ficavam inclusos na hierarquia de memória da máquina principal (SEO, 2009).

Este projeto serviu de base para outras pesquisas e estudos, assim gerando evoluções arquitetônicas, que culminou no desenvolvimento de um dos primeiros sistemas operacionais capaz de paralelizar a execução de *softwares*, através do compartilhamento de tempo do processador, o centro de computação do Instituto de Tecnologia de Massachusetts desenvolveu o *Compatible Time Sharing System* (CTSS), o termo "*compatible*" se refere a compatibilidade com padrões de processamento em lote utilizado pelo sistema operacional, o sistema operacional criado pela MIT foi de extrema importância para demonstrar a viável possibilidade dos sistemas utilizar *time sharing*, além de colaborar para o surgimento de novos modelos de implementação desse conceito (WILLIAMS, 2007; POLLON, 2008).

No final da década 60 a equipe do Centro Científico Cambridge - *Cambridge Scientific Center* (CSC) desenvolveu o primeiro sistema operacional de máquina virtual de sucesso baseado em *hardware* totalmente virtualizado, o CP-40. Com a evolução da plataforma e dos sistemas operacionais de virtualização a IBM criou o sistema VM/370, capaz de executar várias máquinas virtuais, com maior memória virtual rodando em cópias virtuais de hardware (WILLIAMS, 2007).

2.2. CONCEITO

A virtualização pode ser definida como uma metodologia que concede a capacidade de compartilhamento dos recursos, de um computador entre diversos ambientes de execução independentes entre si, possibilitando a simulação total ou

parcial de diferentes plataformas através do particionamento de *hardware* e *software* (QUEVEDO JUNIOR, 2008; SEO, 2009).

Com a virtualização podemos executar dois ou mais sistemas operacionais em uma mesma máquina simultaneamente podem transferir o processamento de máquinas físicas para ambientes virtualizados, diminuindo assim a quantidade de equipamentos bem como os custos de criação e manutenção uma rede, também é possível criar ambientes isolados para executar aplicações inseguras, ou realizar testes sem instabilizar ou afetar a infraestrutura computacional. física existente. (RODRIGUES, 2009)

A possibilidade de miniaturização, permitir a virtualização de inúmeros sistemas operacionais de forma massificada, distribuída entre vários equipamentos, faz com que o grau de abstração da infraestrutura em nuvem seja dependente da capacidade de virtualização (MARTINS, 2011).

A arquitetura *cloud* demanda de independência de sistemas operacionais e aplicações, se adaptando a qualquer necessidade do cliente provisionando recursos de maneira automática sem a realização de operações burocráticas, e integrando de forma global os vários níveis da aplicação: *hardware*, *middleware* e *software*. (MARTINS, 2011)

2.3. MÁQUINA VIRTUAL

Conforme visto anteriormente, as interfaces ISA e ABI padronizam o desenvolvimento independente de sistemas computacionais, porém possuem pouca flexibilidade de integração entre elas. No entanto hoje em dia é possível adicionar uma camada de virtualização por *software*, capaz de oferecer uma interface que possibilite o acoplamento entre as demais interfaces de comunicação, contornando assim os problemas de relacionamento entre as camadas, a máquina com esta nova camada é chamada de máquina virtual, e a camada de virtualização em si recebe o nome de monitor de máquina virtual ou hipervisor (SILVA, 2007; LAUREANO et. al., 2009).

Uma máquina virtual compõe-se de três partes essenciais: (LAUREANO et. al., 2009).

Sistema real

O sistema real, ou sistema hospedeiro, é o sistema nativo da máquina física, que hospeda os recursos de hardware e software do sistema (LAUREANO et. al., 2009).

Sistema virtual

O sistema virtual, ou sistema convidado, é o sistema executado sobre a camada de virtualização, podendo existir mais de um sistema virtual sendo executado simultaneamente sobre o mesmo sistema hospedeiro (LAUREANO et. al., 2009).

Camada de virtualização

A camada de virtualização, ou monitor, disponibiliza a interface virtual que irá interligar o sistema real com o sistema virtual (LAUREANO et. al., 2009).

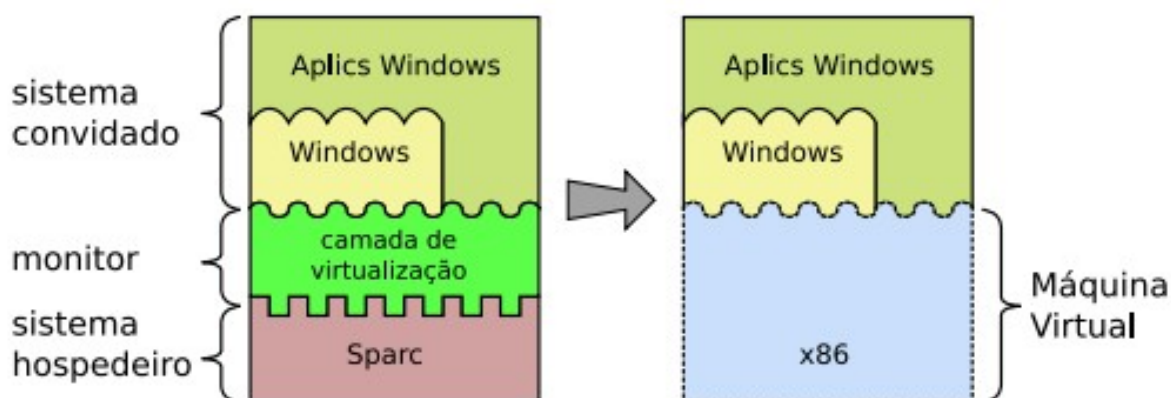


Figura 7 – Exemplo de Máquina Virtual

Fonte: <http://www.dainf.ct.utfpr.edu.br/~maziero/lib/exe/fetch.php/research:2008-sbseg-mc.pdf>

A virtualização pode ser implementada de diferentes formas, tais como:

Virtualização completa

Os sistemas operacional convidado planejado para uma determinada plataforma é executado em outra plataforma de arquitetura diferente, o SO desenvolvido para o hardware “A” é executado no hardware “B” (LAUREANO et. al., 2009).

Emulação do sistema operacional

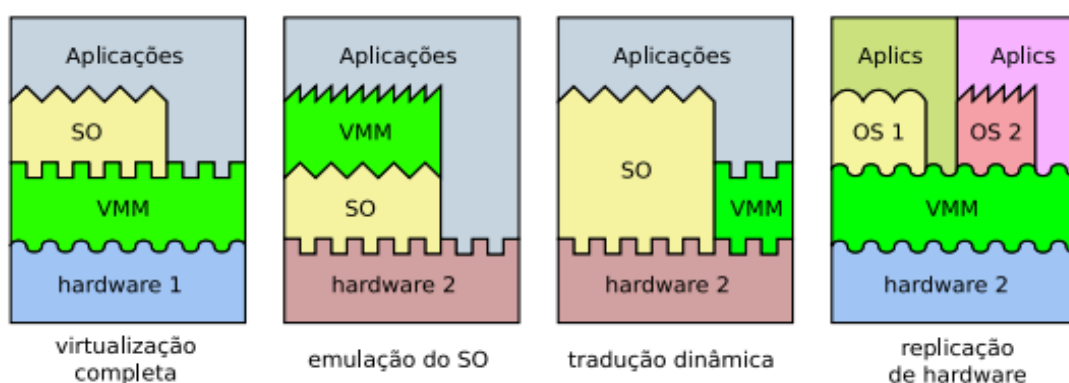
As aplicações desenvolvidas para um sistema operacional em específico, *Microsoft Office*, são executadas em sistemas operacionais de terceiros, como o Linux, por exemplo, porém os SOs são executados com os recursos das mesma plataforma (LAUREANO et. al., 2009).

Tradução dinâmica

"As instruções de máquina das aplicações são traduzidas durante a execução em outras instruções mais eficientes para a mesma plataforma." (LAUREANO et. al., 2009).

Replicação de hardware

Os recursos físicos reais são duplicados e instanciados varias vezes, cada instância possui seu próprio sistema operacional convidado com seus respectivos softwares (LAUREANO et. al., 2009).



2.4. PROPRIEDADES

Uma camada de virtualização para ser eficiente deve possuir algumas propriedades básicas: ela deve fornecer um ambiente de hardware idêntico ao da máquina hospedeira, deve ter um bom desempenho com mínima ou nenhuma queda de desempenho, além de ter total controle sobre o sistema nativo da máquina física (QUEVEDO JUNIOR, 2008; LAUREANO et. al., 2009).

Equivalência

A camada de virtualização fornece recursos de *hardware* quase idênticos ao da máquina hospedeira, podendo diferenciar no tamanho de disco para armazenamento e quantidade de memória alocada, todos os processos executados na máquina virtual devem ter o mesmo comportamento quando executados na máquina real (LAUREANO et. al., 2009)

Controle de recursos

“O hipervisor deve possuir o controle completo dos recursos da máquina real: nenhum programa executando na máquina virtual deve possuir acesso a recursos que não tenham sido explicitamente alocados a ele pelo hipervisor, que deve intermediar todos os acessos. Além disso, a qualquer instante o hipervisor pode resgatar recursos previamente alocados” (LAUREANO et. al., 2009).

Eficiência

“Grande parte das instruções do processador virtual (o processador provido pelo hipervisor) deve ser executada diretamente pelo processador da máquina real, sem intervenção do hipervisor. As instruções da máquina virtual que não puderem ser executadas pelo processador real devem ser interpretadas pelo hipervisor e traduzidas em ações equivalentes no processador real. Instruções simples, que não afetem outras máquinas virtuais ou aplicações, podem ser

executadas diretamente no hardware.” (LAUREANO et. al., 2009)

Isolamento

“Esta propriedade garante que um software em execução em uma máquina virtual não possa ver, influenciar ou modificar outro software em execução no hipervisor ou em outra máquina virtual. Esta propriedade pode ser utilizada para garantir que erros de software ou aplicações maliciosas possam ser contidos em um ambiente controlado (uma máquina virtual), sem afetar outras partes do sistema.” (LAUREANO et. al., 2009).

Inspeção

“O hipervisor tem acesso e controle sobre todas as informações do estado interno da máquina virtual, como registradores do processador, conteúdo de memória, eventos etc.” (LAUREANO et. al., 2009).

Gerenciabilidade

O gerenciamento das instâncias de máquinas virtuais é simplificado, devido a cada máquina virtual estar sobre a mesma camada de virtualização, e serem independentes das demais máquinas virtuais. O gerenciador de máquina virtual gerencia o uso e as solicitações realizadas pelas máquinas virtuais ao hardware (LAUREANO et. al., 2009)

Encapsulamento

“Como o hipervisor tem acesso e controle sobre o estado interno de cada máquina virtual em execução, ele pode salvar checkpoints de uma máquina virtual, periodicamente ou em situações especiais (por exemplo, antes de uma atualização de sistema operacional). Esses checkpoints são úteis para retornar a máquina virtual a estados anteriores (rollback), para análises post-mortem em caso de falhas, ou para permitir a migração da máquina virtual entre hipervisores executando em computadores distintos.” (LAUREANO et. al., 2009).

Recursividade

“Alguns sistemas de máquinas virtuais exibem também esta propriedade: deve ser possível executar um hipervisor dentro de uma máquina virtual, produzindo um novo nível de máquinas virtuais. Neste caso, a máquina real é normalmente denominada máquina de nível 0” (LAUREANO et. al., 2009).

3. COMPUTAÇÃO EM GRADE

A computação em *grid* ganhou em meados dos anos 90 a partir da pesquisa e evolução de duas áreas da tecnologia da informação, processamento paralelo e sistema distribuídos, a junção dessas áreas resultou em um novo conceito computacional, de grande desempenho, alta velocidade de compartilhamento de informações e heterogeneidade de recursos. O termo *grid* foi adotado como uma analogia, ao “power grid” (grade de energia) que disponibiliza energia elétrica de forma transparente, sem que o usuário saiba qual é sua origem ou tenha conhecimento da complexidade dos meios de distribuição e transmissão da energia elétrica (GÓES et. al., 2005).

Computação em *grid* divide seu processamento entre diversos equipamentos heterogêneos, geograficamente distribuídos e francamente interconectados, a união de todos estes equipamentos forma a unidade de uma máquina virtual com grande capacidade computacional, capaz de alcançar altas taxas de processamento, podendo solucionar problemas de grande escala (GÓES et. al., 2005, POLLON, 2008)

3.1. PROPRIEDADES

Um ambiente em *grid* possui características inerentes a sua infraestrutura, as principais propriedades para suporte a uma plataforma em grade são:

Heterogeneidade

Uma grade agrega múltiplos recursos de natureza heterogênea e que envolvem uma ampla diversidade de tecnologias (GÓES et. al., 2005).

Escalabilidade

Uma grade pode aumentar sua capacidade de recursos, partindo de pouquíssimos servidores instanciados até alcançar milhões de instancias sem perder desempenho. Porém a dispersão geográfica da rede exige um planejamento

para prevenir imprevistos que possam acontecer, relacionados a largura de banda para o tráfego de informações ou à latência de transmissão. (GÓES et. al., 2005)

Compartilhamento de recursos

“Os recursos de uma grade computacional não podem ser dedicados para nenhuma aplicação específica.” (GÓES et. al., 2005).

Múltiplos domínios administrativos

Os recursos de uma grade podem distribuir-se geograficamente em diversos domínios administrativos, sendo compartilhados por inúmeras organizações onde cada uma define suas próprias políticas de restrições e regras de uso dos recursos, a serem seguidas (GÓES et. al., 2005).

Controle distribuído

“Em uma grade não existe um gerenciador centralizado que possui uma visão global do sistema. Então cada componente da grade deve ser autônomo”. (GÓES et. al., 2005)

Dinamismo e adaptabilidade

Em uma grade, a insuficiência de recursos não é exceção, é uma regra. Devido ao ambiente heterogêneo de uma grade, a probabilidade de ocorrer uma falha em algum recurso é alta. Portanto, os *softwares* e serviço devem se adaptar ao comportamento e disponibilidade da grade (GÓES et. al., 2005).

3.2. COMPARAÇÃO ENTRE COMPUTAÇÃO EM GRADE E COMPUTAÇÃO EM NUVEM

A computação em Grade e a Computação em Nuvem possuem aspectos similares, pois a *grid computing* é um dos modelos que deram origem a computação em nuvem, porém possuem diferenças que devem ser ressaltadas. Os pontos de divergência que merecem atenção são:

Caraterística	CG	CN
Foco	Orientação científica. Suportado, essencialmente, por comunidades de investigação.	Orientação maioritariamente comercial.
Tipo de gestão do utilizador	Sistema de gestão descentralizada e distribuída.	Sistema a operar centralizadamente em cada nuvem.
Modelo de pagamento	Tipicamente, o pagamento é feito por taxa fixa. Esta é cobrada por serviço ou pelas diferentes organizações, que partilham recursos excedentes.	Modelo de pagamento flexível, de acordo com as necessidades do negócio. É designado por PAYG.
Internet	Partilhada. Consequentemente, com alta latência e baixa largura de banda.	Dedicada. Portanto, com baixa latência e alta largura de banda.
Padrão de recursos	Modelo colaborativo e coordenativo, através de recursos heterogéneos e independentes, com interfaces normalizadas e abertas.	Modelo homogéneo e virtual, gerido por uma entidade central (e.g. <i>Google</i>), com capacidades de aprovisionamento dinâmico e escalabilidade.
Segurança/privacidade	Disponibiliza serviços de segurança. Por exemplo, através de credenciais, usando mecanismos de criptografia assimétrica e certificados digitais para acesso aos recursos.	Tipicamente, cada utilizador/aplicação utiliza uma máquina virtual, que assegura o isolamento da informação. Utilizam-se serviços de acordo com as permissões (e.g. <i>Access Control List (ACL)</i> e/ou <i>membership services</i>).
Negociação	Baseado em parâmetros <i>Service Level Agreement (SLA)</i> , definidos ao contratar o serviço.	
Uniformização	Interfaces abertas e normalizadas, inerentes à CG.	Utilização de tecnologias padrão, por exemplo, <i>web services (Simple Object Access Protocol (SOAP) e Representational State Transfer (REST))</i> .
Escalabilidade	Possibilita o aumento do número de nós.	Oferece reconfiguração dos recursos virtuais de um modo dinâmico e automático.
Qualidade do serviço	Devido à colaboração e partilha de recursos, não existem parâmetros <i>Quality of Service (QoS)</i> definidos.	QoS é uma das características inerentes aos fornecedores da nuvem. Constitui um dos fatores que, no mercado, distingue os fornecedores.
Alocação de recursos	A alocação é realizada de igual modo entre os utilizadores.	A alocação de recursos é diferenciada, consoante as necessidades de negócio.
Heterogeneidade	Suporte para recursos heterogéneos e sem obrigatoriedade quanto a tecnologias e padrões de desenho.	

Figura 9 – Comparação entre Computação em Grade e Computação em Nuvem
 Fonte: <http://repositorio.ipl.pt/bitstream/10400.21/2375/1/Disserta%C3%A7%C3%A3o.pdf>

4. DESENVOLVIMENTO

Neste capítulo será desenvolvido um projeto que visa demonstrar as etapas de implementação de um sistema escalonado. O sistema consiste em uma aplicação capaz de realizar *uploads* de arquivos, armazenando-os na nuvem, possibilitando fácil acessibilidade e capacidade de download das medias armazenadas pela aplicação.

O objetivo desse experimento é evidenciar a utilização do modelo de plataforma como serviço – PaaS, neste caso usaremos a plataforma *Google App Engine* - GAE, para o desenvolvimento de uma aplicação seguindo o modelo de software como serviço – SaaS, objetivando evidenciar a capacidade de escalonamento de armazenamento e processamento da computação em nuvem, conforme as necessidades do sistema.

4.1. FERRAMENTAS UTILIZADAS

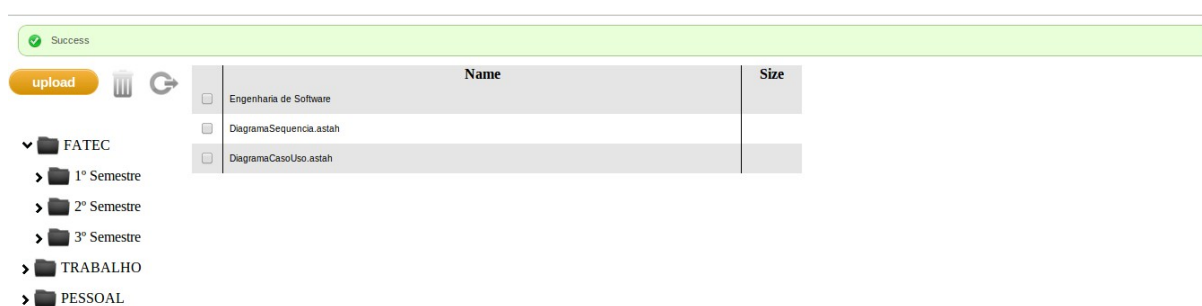
Para o desenvolvimento da aplicação foram utilizadas as ferramentas: Ambiente Integrado de Desenvolvimento - *Integrated Development Environment* (IDE) Eclipse, versão 3.8, a linguagem de programação adotada foi Java com a utilização do *framework* Spring, também foi utilizado o Kit de Desenvolvimento de Software - *Software Development Kit* (SDK) *Google App Engine*, além das APIs *Google Cloud Storage* e *Google OAuth2*.

4.2. PLANEJAMENTO

O *SyncBox*, será uma aplicação integrada com os serviços Google, devido ao fato da hospedagem e a plataforma de desenvolvimento serem providas pela própria empresa Google, o ambiente fornecido disponibiliza várias bibliotecas e APIs que possibilitam e facilitam a utilização dos diversos serviços ofertados pelo gigante da internet, para essa aplicação que será desenvolvida faremos o uso da biblioteca *UserService*, essa é uma biblioteca nativa do *Google App Engine SDK*, que permite utilizar contas Google, para que o usuário se autentique para a utilização da aplicação.

Outro serviço Google a ser utilizado será o *Google Cloud Storage*, serviço que habilita o armazenamento em disco dentro a infraestrutura em nuvem utilizada pelo GAE, a persistência dos arquivos será realizada por intermédio da *Json API*, versão 1.0 para a linguagem java, para utilizar está interface de desenvolvimento é preciso que o servidor da aplicação possua uma chave de identificação que permite a utilização desses serviços, para validar esta identificação iremos utilizar a API de autenticação *Google OAuth2*.

Ao se autenticar na aplicação o usuário terá as opções de subir um arquivo para nuvem, deletar arquivos já armazenados e também terá a opção de sair da aplicação. A baixo segue a imagem do sistema:



*Figura 10 – Aplicação SyncBox
Fonte: Do Autor*

A coluna do lado esquerdo mostra da aplicação mostra a estrutura de dados criada pelo usuário para organizar seus armazenamentos. O código-fonte da aplicação está em anexo no apêndice.

A próxima imagem mostra o gráfico que ilustra o consumo e alocação dos recursos conforme os usuários do sistema armazenam suas informações, e necessitam de mais espaço em disco, para comportar a quantidade dos arquivos enviados para a nuvem:

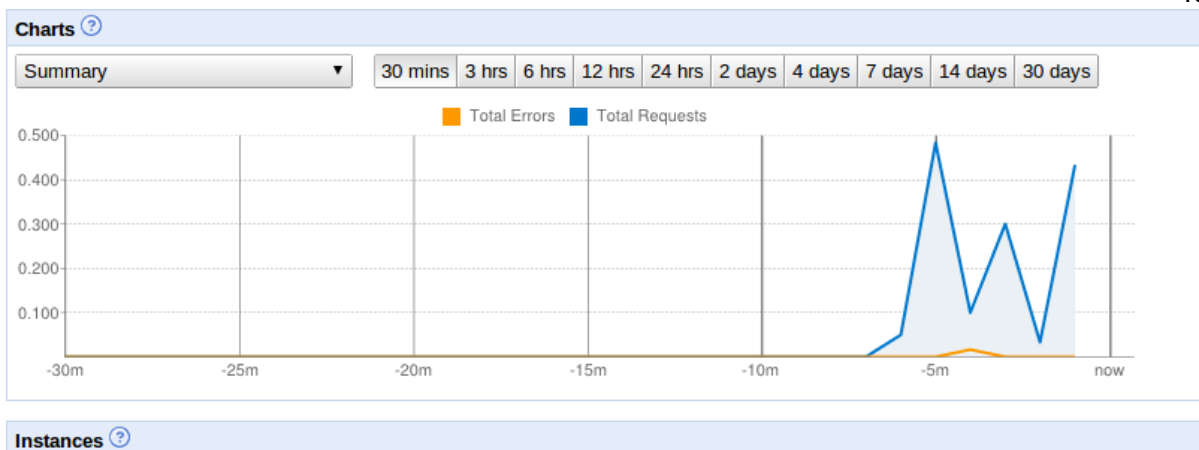


Figura 11 – Alocação de armazenamento
Fonte: Painel de administração do Google App Engine.

5. CONSIDERAÇÕES FINAIS

Os serviços *online* se tornaram parte do nosso cotidiano, compras *online* e *utilização de internet banking*, são exemplos de tarefas que passamos a realizar virtualmente, a quantidade e demanda de serviços como estes estão crescendo cada dia mais e servidores que possuem arquiteturas convencionais, não conseguem suprir a capacidade necessária de processamento que esses serviços demandam, *data centers on-premise* apresentam uma limitação e alto custo, em acrescentar recursos de armazenamento e processamento de dados conforme a necessidade da aplicação. Neste cenário a Computação em Nuvem é a melhor alternativa para contornar os problemas citados, mantendo a qualidade e desempenho do serviço hospedado.

Esse trabalho monográfico apresentou o conceito e a evolução histórica da Computação em Nuvem, no segundo capítulo também foram descritos os modelos de serviço e implementação que compõem o paradigma estudado, no terceiro capítulo foi abordado o conceito de virtualização que uma das concepções que possibilitam o escalonamento dos sistemas em nuvem, já no quarto capítulo foi levantado o conceito da computação em grade que também compõe os caracterização da computação em nuvem. Após toda a contextualização teórica foi apresentado o capítulo que descreve as etapas de desenvolvimento do experimento realizado para demonstrar a capacidade de escalonamento dos sistemas em nuvem.

A parte prática demonstrou a agilidade de alocação de novos recursos computacionais em tempo real, focando em uma das principais características da computação em nuvem que a possibilidade de escalonar seu processamento e armazenamento de dados.

Foram utilizados para a implementação do experimento a linguagem Java, para a codificação da parte que será executada no lado do servidor da aplicação, o *Google App Engine SDK*, um kit de desenvolvimento fornecido pela empresa Google para que desenvolvedores possam codificar aplicações que serão executadas nos servidores *cloud*, que também recebem o nome de *Google App Engine*. O *SDK* oferece uma vasta gama de bibliotecas que facilitam a implementação de chamadas de serviços da nuvem, para auxiliar na parte Java foi utilizado o *framework* Spring

que também agiliza o desenvolvimento, tornando assim o trabalho de codar mais simples.

Após este estudo, posso afirmar que a computação em nuvem é um modelo inovador, que embora seja relativamente novo já é uma tendência, e por ser uma tecnologia disruptiva, será a fonte e plataforma de muitas outras inovações na área de tecnologia da informação. O conceito ainda novo e devido a esta grande quantidade inovações que ele gera e demanda o paradigma acaba sofrendo constantes evoluções, o que pode deixar alguns céticos, cautelosos em relação a adoção desta na tecnologia embora seu potencial seja inquestionável.

6. REFERÊNCIAS BIBLIOGRÁFICAS

AMAZON. **Amazon Elastic Compute Cloud (Amazon EC2)**. Disponível em: <http://aws.amazon.com/ec2>. Acesso em: 31/10/2013.

_____. **AWS Cloud Computing Use Cases**. Disponível em: <http://aws.amazon.com/what-is-cloud-computing>. Acesso em: 31/10/2013.

_____. **Amazon Simple Storage Service (Amazon S3)**. Disponível em: <http://aws.amazon.com/s3>. Acesso em: 31/10/2013.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **Citação: NBR-10520/ago - 2002**. Rio de Janeiro: ABNT, 2002.

_____. **Referências: NBR-6023/ago. 2002**. Rio de Janeiro: ABNT, 2002.

CASCÃO, M. **Segurança e privacidade na nuvem: ignorar cloud computing não é a melhor solução**. 2013. Disponível em: <http://corporate.canaltech.com.br/noticia/cloud-computing/Seguranca-e-privacidade-na-Nuvem-ignorar-Cloud-Computing-nao-e-a-melhor-solucao/>. Acesso em: 31/10/2013.

CASTRO, L. A. S. **Controlo de infra-estruturas de Cloud Computing**. Departamento de Electrónica, Telecomunicações e Informática. Universidade de Aveiro. 2009.

GARFINKEL, S.L. **Architects of the Information Society**. 1ª ed. Cambridge: MIT Press, 1999.

GÓES, L. F. W.; et. al. **Computação em Grade: Conceitos, Tecnologias, Aplicações e Tendências**. Escola Regional de Informática de Minas Gerai (ERI-MG), 2005.

GOMES, C. N. **Estudo do Paradigma Computação em Nuvem**. Instituto Superior de Engenharia de Lisboa. 2012.

GOOGLE. **Google Apps is out of beta (yes, really)**. 2009. Disponível em: <http://googleblog.blogspot.com.br/2009/07/google-apps-is-out-of-beta-yes-really.html>. Acesso em: 31/10/2013.

_____. **Google Apps grows up**. 2007. Disponível em: <http://googleblog.blogspot.com.br/2007/02/google-apps-grows-up.html>. Acesso em: 31/10/2013.

_____. **Salesforce for Google Apps.** 2008. Disponível em: <http://googleblog.blogspot.com.br/2008/04/posted-by-scott-mcmullan-google-apps.html>. Acesso em: 31/10/2013.

_____. **Google Apps for Business.** Disponível em: <http://www.google.com/enterprise/apps/business/>. Acesso em: 31/10/2013.

IBM. **Você realmente sabe o que é Big Data?**. 2012 Disponível em: https://www.ibm.com/developerworks/community/blogs/ctaurion/entry/voce_realmente_sabe_o_que_e_big_data. Acesso em: 31/10/2013.

KRIKOS, A.; **Disruptive Technology Business Models in Cloud Computing.** Massachusetts Institute of Technology. 2010.

LAUREANO, M. A. P. MAZIERO, C. A. **Virtualização: Conceitos e Aplicações em Segurança.** Pontifícia Universidade Católica do Paraná. 2009.

LAURO, L. **Cloud computing e Software as a Service** . PaLestra proferida na Endeavor Brasil, São Paulo, 05/02/2013.

MARTINS, A. **Fundamentos de Computação Nuvem para Governos.** Serviço Federal de Processamento de Dados. Brasília,2011.

MOHAMED, A. **A history of cloud computing.** ComputerWeekly. 2009. Disponível em: <http://www.computerweekly.com/feature/A-history-of-cloud-computing>. Acesso em: 31/10/2013.

NIST, NATIONAL INSTITUTE FOR STANDARDS AND TECHNOLOGY (EUA). Information Technology Laboratory. **The NIST Definition of Cloud Computing.** Gaithersburg, 2011. 7 p.

POLLON, V. **Virtualização de Servidores em Ambientes Heterogêneos e Distribuídos -Estudo de Caso.** Instituto de Informática. Universidade Federal do Rio Grande do Sul. 2008.

QUEVEDO JUNIOR, D. P. **Virtualização: Conceitos, Técnicas Aplicadas e Um Comparativo de Desempenho Entre as Principais Ferramentas Sem Custo de Licenciamento.** Instituto Superior Tupy. Sociedade Educacional de Santa Catarina. 2008.

RAINER, R.K.; CEGIELSKI, C.G. **Introdução a Sistemas de Informação: Apoiando e Transformando Negócios na Era da Mobilidade** 3ª ed. São Paulo: Elsevier, 2012.

RODRIGUES, E. M. **Realocação de Recursos em Ambientes Virtualizados**. Pontifícia Universidade Católica do Rio Grande do Sul. 2009.

SEO, C.E. **Virtualização – Problemas e Desafios**. IBM Linux Technology Center. 2009.

SILVA, E. L. da; MENEZES, E. M. **Metodologia da Pesquisa e Elaboração de Dissertação**. 3ª ed - Florianópolis: Laboratório de Ensino a Distância da UFSC,2001. p. 20-21.

SILVA, R. F. da. **Virtualização de Sistemas Operacionais**. Instituto Superior de Tecnologia em Ciências da Computação. 2007.

SIMON, I. **A ARPANET**. Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo – 1997. Disponível em: <http://www.ime.usp.br/~is/abc/abc/node20.html>. Acesso em: 31/10/2013.

TAURION, C. **Cloud Computing: Computação em Nuvem: Transformando o mundo da tecnologia da informação**. Rio de Janeiro: Brasport, 2009.

TONINI, C. A.; SPÍNOLA, M de M.; LAURINHO F. J. B. **A Análise SWOT: Uma Nova Perspectiva Para a Aplicação do Seis Sigma no Desenvolvimento de Software**. In: ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO, 28, Foz do Iguaçu, 2007. **Anais...** Foz do Iguaçu: Associação Brasileira de Engenharia de Produção, 2007 10 p. P 1-10.

VERAS, M. **Virtualização: componente central do Datacenter**. Rio de Janeiro: Brasport, 2011.

WILLIAMS, D. E.; JUAN G. **Virtualization with Xen(tm): Including XenEnterprise, XenServer, and XenExpress**. Syngress Publishing, Inc. Burlington, 2007.

WINKLER, V. J. R. **Cloud Computing: Data Privacy in the Cloud**. 2012. Disponível em: <http://technet.microsoft.com/pt-br/magazine/jj554305.aspx>. Acesso em: 31/10/2013.

ZUFFO, M. K.; et. al. A computação em nuvem na Universidade de São Paulo. **Revista USP**, São Paulo, n. 97, p 9-18, mar./abr./maio 2013.

7. GLOSSÁRIO

Cloud Computing. Computação em Nuvem.

Data Centers. Complexos que concentram vários equipamentos de processamento e armazenamento de dados.

Device. Dispositivo

Download. Baixar arquivos armazenados na nuvem

Firewall. É uma barreira de proteção que ajuda a bloquear o acesso de conteúdo malicioso, mas sem impedir que o tráfego dos dados que precisam ser transmitidos

Grid Computing. Computação em Grade.

Hardware. Parte física de equipamentos computacionais (teclado, monitor).

JSON. Formato de transmissão de dados, utilizado para exportar ou importar informações externas ao sistema.

Killer Apps. Aplicações que viabilizam uma nova tecnologia, aplicações que tornam uma nova tecnologia útil, atraente, altamente desejável aos olhos do consumidor.

Maniqueísmo. Dualismo entre o bem e o mal.

Middleware. Camada de software que permite a comunicação entre aplicações (distribuídas)

On Demand. Sob Demanda

On-premise. Recurso de execução local ou privado.

Software. Parte lógica que opera sobre o hardware (aplicações e sistemas computacionais).

Tecnologia Disruptiva. Rompe paradigmas, derrubando as regras e modelos de negócios já existentes.

Upload. Subir arquivos, que serão armazenados na nuvem

Utility Computing. Computação Utilitária.

Vendor Lock-in. É o fornecedor que cria uma dependência para o consumidor sobre seus produtos e serviços, que a mudança para outro fornecedor não pode ser feita sem que grandes custos de transferência estejam envolvidos.

8. APÊNDICE A – CÓDIGO FONTE

- Métodos Java de *upload* e exclusão dos arquivos salvos na nuvem:

```

/**
 * Upload File
 * @param file
 */
@RequestMapping(value = {"/uploadFile"}, method = RequestMethod.POST)
@ResponseStatus(HttpStatus.OK)
public @ResponseBody void upload(@RequestParam("mpFile") MultipartFile file) {

    if (file != null) {
        try {
            GcsOutputChannel outputChannel = gcsService.createOrReplace(
                getFileName(file.getOriginalFilename()), GcsFileOptions.getDefaultInstance());

            storageService.copy(file.getInputStream(), Channels.newOutputStream(outputChannel));
        } catch (IOException e) {
            log.severe(e.toString());
            e.printStackTrace();
        }
    }
}

@RequestMapping(value = {"/deleteFile"}, method = RequestMethod.POST)
public @ResponseBody void deleteFile(@RequestParam("path") String[] path) throws IOException{
    for(String file : path){
        storageService.delete(file);
    }
}

```

- Método Java de listagem dos arquivos armazenados na nuvem:

```

/**
 * List All Files
 * @param model
 * @return
 * @throws IOException
 */
@RequestMapping(value = {"/listFile"}, method = RequestMethod.GET)
public @ResponseBody String listFiles(Model model) throws IOException{
    Storage.Objects.List listObjects = storage.objects().list(BUCKET_NAME);
    List<FileDTO> listFiles = new ArrayList<FileDTO>();

    Objects objects;
    StringBuilder partId;
    String fileName;

    do {
        objects = listObjects.execute();

        /**
         *
        */
    } while (null != objects.getNextPageToken());

    BucketObjetsDTO objeDTO = new BucketObjetsDTO();
    objeDTO.setListFiles(listFiles);

    Gson gson = new Gson();
    String json = gson.toJson(objeDTO);

    return json;
}

```

- Clausula *do-while* do método de listagem dos arquivos:

```

do {
    objects = listObjects.execute();

    for (StorageObject object : objects.getItems()) {
        // Separa o folderId e o nome do arquivo da string
        String[] parts= object.getId().split("/");

        fileName = parts[parts.length-2];

        partId = new StringBuilder("");
        for(int i = 0; i < (parts.length-2); i++){
            partId.append(parts[i]).append("/");
        }

        // Cria objeto FileDTO
        FileDTO fileDTO = new FileDTO();
        fileDTO.setId(object.getId());
        fileDTO.setName(fileName);
        fileDTO.setType(object.getContentType());
        fileDTO.setBucket(object.getBucket());
        fileDTO.setSize(object.getSize());
        fileDTO.setComponentCount(object.getComponentCount());
        listFiles.add(fileDTO);
    }
    listObjects.setPageToken(objects.getNextPageToken());
} while (null != objects.getNextPageToken());

```