COMPUTAÇÃO EM NUVENS, VISÃO COMPARATIVA ENTRE AS PRINCIPAIS PLATAFORMAS DE MERCADO

Olavo Oliveira Neto*

Rejane Cunha Freitas**

RESUMO

Este artigo tem por objetivo explanar os aspectos técnicos e teóricos sobre a computação em nuvens como plataforma de disponibilização de aplicações. Através do referencial teórico, explicamos o conceito geral e a história da computação em nuvens, destacando principalmente as vantagens econômicas e usuais, mas demonstrando também os problemas que a computação em nuvens pode ocasionar. Também foi realizada uma pesquisa comparativa entre as principais plataformas de mercado e a implementação de uma aplicação em duas diferentes plataformas para a realização de testes e, com isto, a conclusão da pesquisa.

PALAVRAS CHAVE: Computação em nuvens – Provisionamento - Elástico

ABSTRACT

This article aims to explain the technical and theoretical knowledge about the cloud computing to provide applications. Through the theoretical referential, we explain the general concept of cloud computing history particularly focusing on the economic advantages and usual, but also demonstrating the problems that cloud computing can cause. Was also carried out a comparative research between the main market platforms and the implementation of an application in two different platforms for the accomplishment of tests and with this the conclusion of the research.

KEYWORDS: Cloud Computing – Provisioning - Elastic

*Aluno do curso de Sistemas de Informação da Faculdade Estácio do Ceará. Email: olavo.o.neto@gmail.com

* *Professora do Curso de Sistemas de Informação da Faculdade Estácio do Ceará. Mestre em Ciência da Computação – UFPE. Email: rejanecf@fic.br

1. Introdução

Vivemos hoje na era da informação, nosso ambiente é cercado de tecnologias que visam nos manter conectados com o mundo. Uma pessoa processa cerca de 34 Gigabytes por dia e nos últimos três anos criou-se mais informações do que nos últimos quarenta mil anos de humanidade (LIMA, 2010), toda essa evolução foi possível graças à inovação tecnológica, principalmente a criação da Internet.

A evolução da computação e da internet foi relativamente rápida. Em menos de 40 anos mudamos um cenário centralizado com *mainframes, em que* as aplicações e os dados eram locais e distribuídos através de redes internas passando para aplicações desktops que compartilham a mesma base de dados. Depois, as aplicações passaram a ser acessadas via browser, disponibilizadas localmente pelas empresas até chegarmos ao nosso cenário atual, em que as aplicações são armazenadas em servidores públicos, com alto poder de processamento e disponibilidade, visando mante-las sempre em funcionamento com o menor custo possível (SOUSA, MOREIRA, e MACHADO, 2009).

A computação em nuvens trouxe de volta uma idéia de forma remodelada, a centralização. Criam-se vários *datacenters* distribuídos, controlados por empresas gigantescas como Microsoft, Google e Amazon. Colocam-se o hardware feito sob demanda, centrais de energia e resfriamento de última geração e *softwares* de controle que provêm aos clientes uma forma rápida de somar, ou retirar, máquinas de seu pátio computacional. Isto é a computação em nuvens. Uma forma de prover serviços com pagamento sob demanda de uso. Você paga pelo o que usa e pelo que necessita. (COMPUTERWORLD US, 2010)

Várias empresas foram criadas a partir desta nova tendência de mercado e investiram pesado para poder garantir sua qualificação nesta nova modalidade da informática. A Amazon foi a primeira a lançar uma plataforma de computação em nuvens conhecida como EC2 (*Elastic Cloud Computing*). Seguida pouco tempo depois pela IBM, Intel, Google, com o *App Engine* e, por fim, a Microsoft, em 2009, disponibilizou o Windows Azure.

As plataformas de computação nas nuvens, baseadas em uma tecnologia de provisionamento elástico, são muito mais que ambientes para disponibilização de aplicações ou armazenamento de arquivos em nuvens. Temos aqui uma tecnologia de alto desempenho e disponibilidade, que visa publicar softwares como serviços na

Web, prover serviços de infraestrutura e promover o armazenamento de dados em nuvens. Além disso, os provedores estão disponibilizando ambientes de desenvolvimento integrados às ferramentas de programação já existentes para tentar viabilizar o desenvolvimento rápido e fácil.

Este trabalho teve por objetivo demonstrar as vantagens da utilização da computação em nuvens através de um estudo da evolução histórica da computação e da análise das principais características que permeiam esta nova tecnologia. Também foi realizada uma comparação entre as principais plataformas de mercado tanto em nível de funcionalidades como de usabilidade, com alguns exemplos de codificação.

Os resultados demonstraram que a computação em nuvens é uma plataforma com uma série de vantagens, principalmente para as pequenas e médias empresas que desejam disponibilizar serviços com o mínimo de gasto em hardware e economizando com mão de obra para manter os serviços funcionando.

Este artigo está dividido em quatro partes, sendo respectivamente: conceitos preliminares que contém o referencial teórico, um descritivo da arquitetura geral que permeiam as plataformas de computação em nuvens, seguido de uma análise individual de arquiteturas de mercado, também contém a metodologia empregada na pesquisa, um estudo comparativo de codificação e os resultados.

2. Conceitos preliminares de computação nas nuvens

Para Miller (2008, p. 7), “a computação em nuvens anuncia uma mudança importante na maneira como nós armazenamos informações e executamos aplicações. Em vez de executarmos os programas e as informações em computadores individuais, tudo será armazenado na ‘nuvem[1]’”. A IBM (IBM, 2009) conceitua computação em nuvens como uma forma de provisionamento sob demanda de recursos computacionais, tais como *hardware*, *software* e armazenamento. Baseado nestas concepções, conceituaremos a computação nas nuvens como um ambiente virtual alocado em “algum lugar” da Internet e, situado fisicamente em algum lugar do globo, em que o usuário, ao demandar determinado

[1] Internet

recurso computacional, tem controle sobre o quanto e quando irá precisar da demanda de *hardware* da máquina e irá pagar somente por aquilo que foi solicitado.

Entendemos ainda que computação nas nuvens seja a junção de *hardware* dedicado (servidores) dentro de complexos, chamados de *data centers*[2], que virtualizam outros servidores a fim de proporcionar o ambiente virtual que será alocado aos clientes (VECCHIOLA, CHU, e BUYY, 2009).

A figura 1 mostra a estrutura básica de um ambiente de computação nas nuvens, em que clientes acessão seus dados através de vários tipos de dispositivos que se conectam as aplicações em nuvens através da Internet.

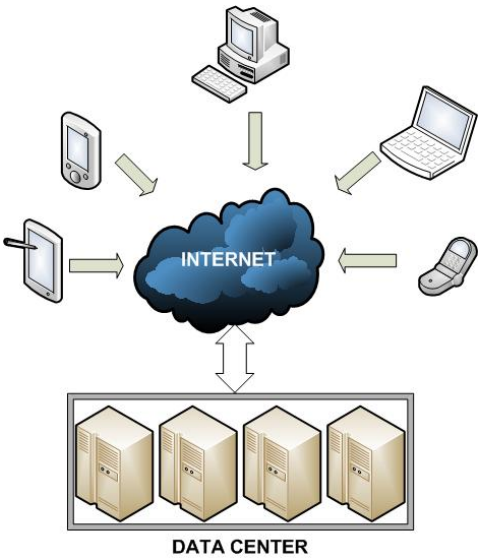


Figura 1. Arquitetura Básica de Plataforma de Nuvem

Independente do conceito, a computação em nuvens é tratada como uma evolução da computação em grade, iniciada na década de 1980, em que vários computadores pessoais independentes foram conectados em rede a fim de fornecer uma plataforma para execução de aplicações paralelas e distribuídas.

Temos na figura 2 uma série de pontos relevantes da evolução da computação em nuvens dos últimos 10 anos, por exemplo, o desenvolvimento dos protocolos que compõem os *webservice*, idealização da arquitetura orientada a serviços, seguido do lançamento das plataformas de computação nas nuvens pela pioneira Amazon em 2006, IBM e Google em 2007 e Microsoft no ano de 2009.

[2] "Um *data center* é uma modalidade de serviço de valor agregado que oferece recursos de processamento e armazenamento de dados em larga escala para que organizações de qualquer porte e mesmo profissionais liberais possam ter ao seu alcance uma estrutura de grande capacidade e flexibilidade, alta segurança, e igualmente capacitada do ponto de vista de hardware e software para processar e armazenar informações" (Computação em nuvem)

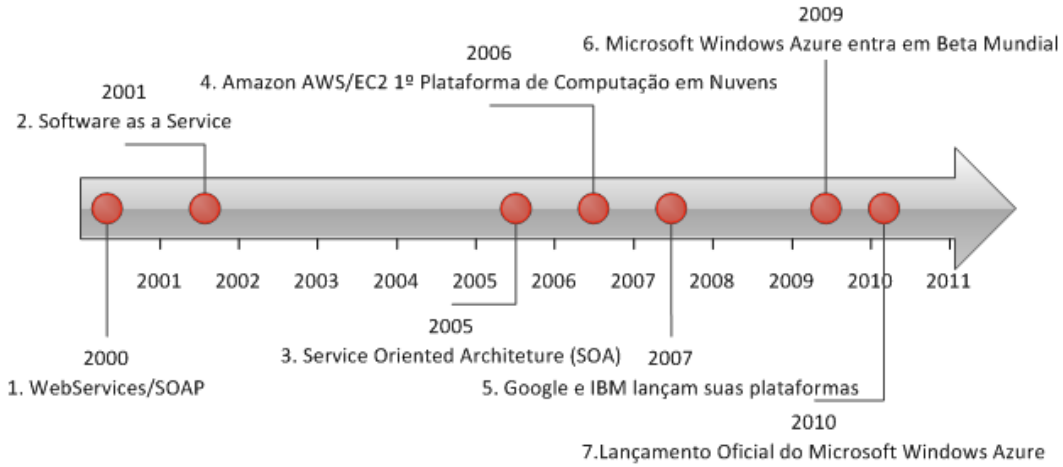


Figura 2. Evoluções relevantes para Computação em Nuvens. Fontes: (MOHAMED, 2009)

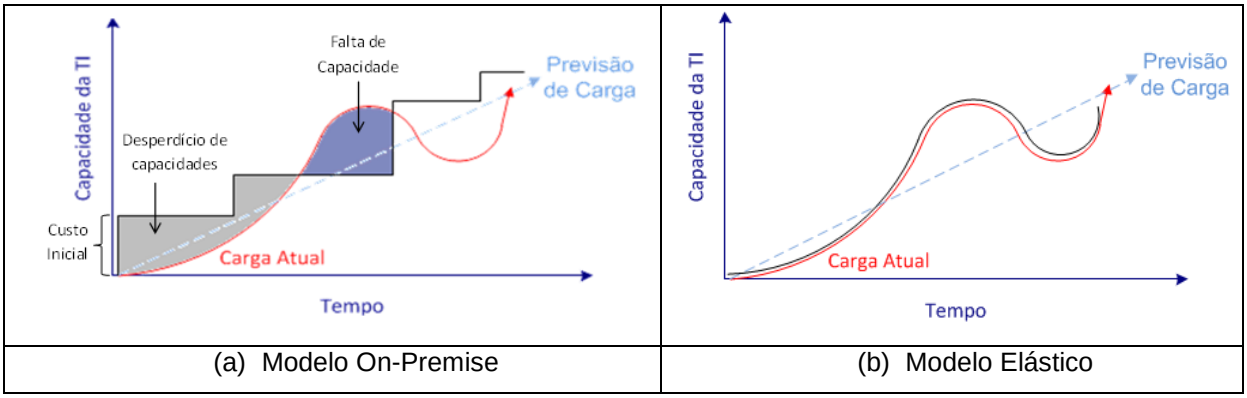
Como principais benefícios da utilização de uma plataforma com recursos computacionais disponíveis na Internet podemos citar:

- Custo:

Para as empresas, as vantagens econômicas do modelo de provisionamento elástico são várias, principalmente quando se trata do custo inicial para aquisição de maquinário de grande porte como servidores. Outra vantagem é a não necessidade de contratação de funcionários dedicados a manter aqueles serviços funcionando. (GARTNER, 2008).

A figura 3 mostra uma correlação entre o modelo (a) *on-premise*[3] e o (b) modelo elástico. No primeiro modelo, há um custo inicial alto que acarreta um desperdício de carga. Esse custo pode ser composto pela aquisição de máquinas e mão de obra. Sempre que a demanda cresce é feito um novo aumento na capacidade sobrepujando a necessidade. Este modelo também é passível de momentos de falta de capacidade devido a um aumento inesperado de carga. No modelo elástico, diferentemente, a carga acompanha diretamente a demanda, não tendo falta ou excessos (MICROSOFT MSDN, 2010).

[3] Segundo Microsoft MSDN - Windows Azure (2009), On-Premise é o ambiente em que as máquinas são de responsabilidade da empresa local e tem alto custo inicial com infraestrutura.



(a) Modelo On-Premise | (b) Modelo Elástico

Figura 3. Correlação entre os, modelos On-Premise e Elástico.

- Riscos, segurança e alta disponibilidade.

Citando Hurwitz Bloor, e Kaufman (2010, p. 29)**, "**o provedor é responsável por toda a segurança, exceto para segurança de acesso**."** Também, neste sentido, Rhoton (2009) concorda e explica que a computação em nuvens move alguns riscos do cliente para o provedor do serviço. Podemos contratualmente estipular tanto a segurança dos dados como o plano de *Disaster Recovery*[4], de forma que, se o provedor não conseguir cumpri-lo, terá que indenizar o cliente.

A tabela 1 ilustra as taxas de disponibilidade para a computação em nuvens.

Tabela 1. Taxas de Disponibilidade em % por provedores de Cloud Computing.

Taxas de Disponibilidade em % por provedores de *Cloud Computing*	
Nome	**Disponibilidade em %**
Amazon Elastic Compute Cloud (Amazon EC2)	99,95%
Google App Engine	99,9%
Locaweb Cloud Computing	99,9%

Fontes: (AMAZON, 2010), (GOOGLE, 2010), (LOCAWEB, 2010)

- Elástico

Segundo Rhoton (2009, p. 9), um dos principais benefícios da computação nas nuvens é a escalabilidade que o provedor disponibiliza para o usuário final. Esta capacidade de provisionamento automático de capacidade pode variar um pouco entre os provedores, por exemplo, para Amazon (2010) e Microsoft (2010) o

[4] Para Snedaker (2007, p. 4), *disaster recovery* é parte do plano de continuidade do negócio e envolve interromper os efeitos do desastre e aplicar a correção o mais rápido possível. Estes desastres podem variar desde uma queda no *link* da internet até mesmo a passagem de um furacão.

provisionamento é controlado através de API[5], que possibilita o aumento ou diminuição da capacidade de forma automática.

- TI Verde

Segundo Locaweb (2010), “a computação em nuvem reduz o consumo de energia do *data center*, contribuindo para a preservação do meio ambiente”. De total acordo, Rhoton explica:

> “Computação ecologicamente sustentável é uma prioridade importante que os gerentes de TI precisam considerar a desenvolver em sua estratégia de infraestrutura de longo prazo. A eficiência energética e eficaz de eliminação e reciclagem dos equipamentos deverão se tornar ainda mais importante no futuro” (2009, p. 43).

Essa economia se dá com a redução de energia por parte dos provedores de computação em nuvens, da reutilização de equipamentos e através de implementos em tecnologia de refrigeração avançada, sensores de temperatura e dutos de ar elaborados para esta finalidade (RHOTON, 2009). .

3. Arquitetura

Segundo Rhoton (2009), “um dos aspectos característicos da computação em nuvens é o foco para orientação a serviço.” Também, neste sentido, concorda a Sun Microsystems (2009), quando explica que a arquitetura de uma plataforma de computação em nuvens pode ser separada por camadas de serviços que podem ser comparadas com as camadas tradicionais de *hardware* e *software*.

3.1. Modelos de Serviço

A arquitetura das plataformas de computação nas nuvens pode ser dividida em três modelos (camadas) de serviço distintas, são elas:

- Software como Serviço (SAAS)

Software como serviço é uma modalidade de distribuição de softwares através da internet na forma de um serviço. Estes softwares são hospedados, mantidos e distribuídos por um provedor de serviço que poderá cobrar ou não por sua utilização.

[5] Interface de Programação de Aplicações. É uma interface disponível por um software para que outras aplicações consigam utilizar suas funcionalidades. (HOWE, 1995).

Para o cliente, as vantagens de utilização de um software SAAS são muitas, tendo em vista que sua utilização se dá através de qualquer dispositivo que contenha um *browser*, não existindo a necessidade de aquisições de hardware, preocupações com compatibilidade de versões de sistema operacional e muito menos com atualizações de versões, já que toda a manutenção do software é responsabilidade do provedor. Por se tratar de uma versão visada para vários usuários e utilizando um modelo de pagamento de assinatura, os preços podem ser menores que de um software instalável (MICROSOFT MSDN, 2010).

- Plataforma como Serviço (PAAS)

Principal modelo para o ambiente de computação em nuvens, a plataforma como serviço é o ambiente fornecido pelo provedor, junto com um conjunto de ferramentas para o desenvolvimento, disponibilização e controle das aplicações.

Este ambiente se dá na forma de máquinas virtuais padronizadas pelo provedor, possibilitando ao cliente utilizar todos os recursos disponibilizados, mas não detendo acesso a modificações de configurações de baixo nível, *hardware* ou provisionamento de outras máquinas virtuais (RHOTON, 2009).

- Infraestrutura como Serviço (IAAS)

Em contrapartida ao PAAS, a infraestrutura como serviço é uma modalidade de distribuição que foca permitir ao cliente criar, customizar e remover máquinas virtuais no ambiente computacional em nuvens. Este controle é feito remotamente e sem a necessidade de contato com o suporte do provedor, a disponibilidade é imediata. Todos os recursos adicionais que permeiam um ambiente computacional, como os ativos de redes, são centralizados e disponibilizados como serviços adicionais que poderão ser utilizados quando necessário (BUYYA, SHIN YEO ET AL, 2009, p. 2).

Segue na tabela 2 exemplos de modelos de serviços correlacionados com as plataformas que implementam e disponibilizam estes modelos.

Tabela 2. Exemplos de modelos de serviços correlacionados com plataformas que as disponibilizam

Exemplos de modelos de computação nas nuvens	
Modelo	**Exemplos**
SAAS	CRM(Customer Relationship Management) da Sales Force
	Google Docs
	Mail Live Office
PAAS	Windows Azure
	Google App Engine
	Amazon AWS EC2
	Aneka
IAAS	Eucalyptus
	Amazon AWS EC2
	Windows Azure

Fonte: Tabela adaptada do artigo (SOUSA, MOREIRA e MACHADO, 2009)

3.2. Segurança

No quesito segurança, devemos entender que as camadas de modelos de serviços sofrem um decréscimo na quantidade de controle sobre o *hardware,* na medida em que vamos subindo da camada servidor até a cliente, ou seja, na camada de IAAS temos um controle total sobre o hardware que é disponibilizado como serviço, mas em nenhum momento é possível acessar diretamente o *hardware* da máquina. A camada PAAS também não nos oferece controles sobre o hardware da máquina, apenas questões relevantes a configurações de ambiente e instalações de softwares. Ao SAAS, por ser um modelo de disponibilização ao cliente, fica vetado qualquer controle sobre a máquina, o máximo permitido é uma customização em cima de sua aplicação. (HURWITZ, BLOOR e KAUFMAN, 2010)

3.3. Armazenamento em nuvens

Para a computação em nuvens e seus modelos de serviços, o armazenamento de dados é tratado como um serviço adicional que poderá ser disponibilizado e cobrado de forma diferenciada dos demais serviços. A utilização destes serviços geralmente é baseada em padrões da web, como REST

(*Representational State Transfer*)[6] e SOAP (*Simple Object Access Protocol*). A Microsoft e a Amazon fornecem bases de dados em nuvens utilizando estes padrões (BUYYA, SHIN YEO ET AL, 2009, p. 6).

4. Arquiteturas Individuais

Apesar de todas as arquiteturas em geral seguirem um modelo bastante comum de disponibilização de serviços através de camadas, que realizam desde o controle do hardware até a disponibilização do software na web, podemos dizer que cada provedor de serviços possui uma série de características próprias que variam desde a disponibilização ou não de todas as camadas, as linguagens suportadas e outras formas de serviços. A seguir, são comentadas as arquiteturas de três provedores de serviços:

4.1. Windows Azure

O Windows Azure é a plataforma de computação em nuvens da Microsoft que visa possibilitar um ambiente de desenvolvimento em múltiplas linguagens, com a possibilidade de integrações entre aplicativos locais e publicados na nuvem, armazenamento relacional, utilização dos serviços live e acesso aos sites pelos mais variados tipos de dispositivos, conforme a Figura 4. (MICROSOFT MSDN, 2010)

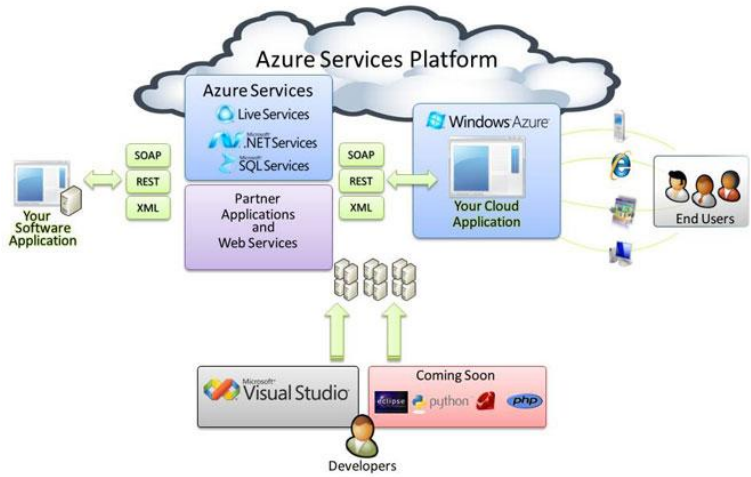


Figura 4. Plataforma de Serviços Azure

[6] Transferência de Estado Representacional: é uma técnica de engenharia de software para sistemas distribuídos, por exemplo, aplicações WEB com funcionalidades parecidas com a dos *webservices* com a diferença de tratar mais detalhes como a URL que está sendo passado e o que ela contém irá diferenciar o que será retornado. (IBM, 2008)

O Azure é composto de uma arquitetura desenhada ao provisionamento individual de instâncias diferenciais entre dois modelos: *Web,* que seria a interface da aplicação, e *Worker, responsável por armazenar a regra de negócio ou computacional.* Esta diferença visa proporcionar um ambiente para acesso simultâneo de vários usuários, além de escalabilidade, pois, na medida em que surgir a necessidade de uma nova instância, seja ela da parte Web ou da computacional, é possível realizar a cópia de instâncias em execução para a criação de um cluster de serviços, conforme mostra a figura 5.

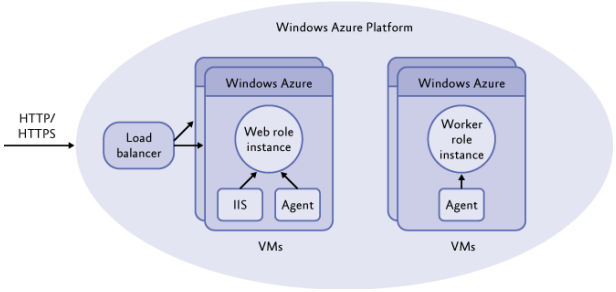


Figura 5. Arquitetura Flexível do Windows Azure

Além disso, a Microsoft disponibiliza uma série de aplicativos prontos para serem adquiridos e utilizados sem a necessidade de implementação, chamado de Azure Services. Ele é composto de:

- **Live Services**: também conhecido como Live Mesh, é um serviço de sincronização de dados dos usuários;
- **SharePoint Services**: portal de colaboração voltado para intranets;
- **CRM (*Customer Relationship Managment*)**: plataforma de relacionamento com o cliente;
- **.Net Services:** serviços para integração, controle de acesso e fluxo de trabalho para as aplicações desenvolvidas.

Outro aspecto da plataforma do Windows Azure é o armazenamento que disponibiliza três formas de persistência de dados em nuvens, sendo elas Tabelas, Filas e *Blobs*[7] e um ambiente relacional para o armazenamento de dados e processamento de consultas, baseado no SQL Server e acessível através de SOAP e REST.

[7] Devido à impossibilidade de um software acessar recursos locais da máquina por causa das questões de segurança. O Azure disponibiliza um modelo de armazenamento de dados grandes sem ter a necessidade de recorrer a um ambiente relacional.

4.2. Google App Engine

O App[8] Engine é a plataforma de desenvolvimento que permite que os aplicativos Web nele publicados sejam disponibilizados na infraestrutura do Google. Esta plataforma tem uma série de recursos aos desenvolvedores, como suporte a linguagens abertas de mercado, como o Java e o Phyton, modelos de aplicação, APIs para integração aos recursos do Google e fornecimento de um ambiente com ajustes e balanceamentos de carga automáticos.

A plataforma computacional do Google trabalha sobre um sistema de cotas por desenvolvedor para o qual é possível disponibilizar gratuitamente até dez aplicações com 500MB de limite de armazenamento e cinco milhões de visitação por mês. Estas cotas também possuem uma série de limitantes, como um timeout de 30 segundos por requisição e um limite de mil linhas de retorno por consultas. Caso haja uma extrapolação dos limitantes no período de gratuidade, a aplicação poderá ser retirada do ar (GOOGLE, 2010).

Além disso, a Google disponibiliza uma série de APIs de integração com serviços já existentes do Google e outros exclusivos para os assinantes. Estes serviços variam desde sistemas para envio de e-mails e edição de imagens até APIs para autenticação, utilizando uma conta integrada ao Google, agendamento de funções, *cache* e uma recuperação de URL, conforme mostra a figura 6.

Também é possível disponibilizar através do domínio próprio do usuário os serviços do Google conhecidos como Apps, por exemplo, o Gmail ou o Google Docs (GOOGLE, 2010).

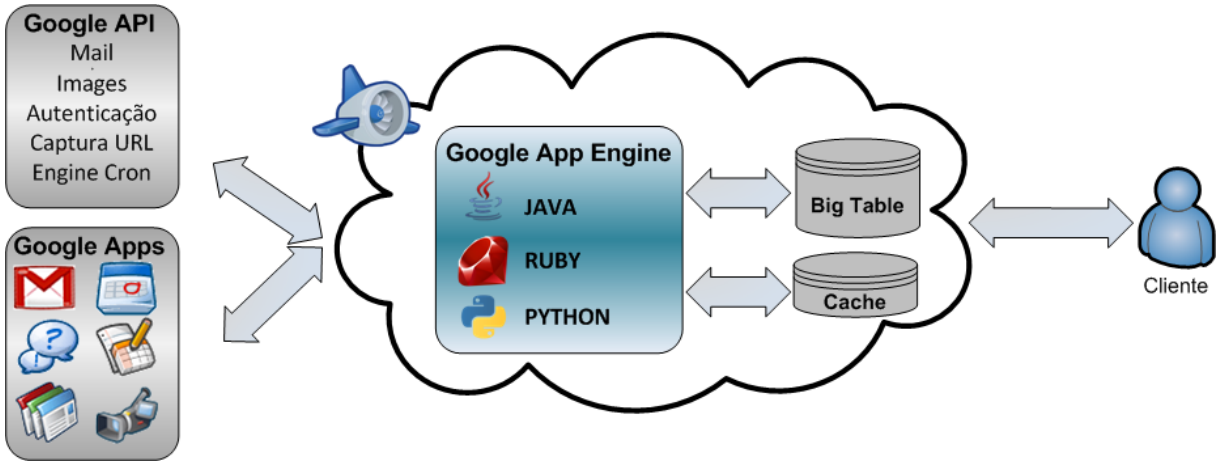


Figura 6. Arquitetura do Google

[8] Aplicação

Também o Google disponibiliza um serviço de armazenamento de dados distribuído, com alto desempenho não relacional, baseados no conceito de entidades sem esquema definido. As propriedades destas são criadas a partir do código da aplicação, ainda citando a figura 6.

4.3. Amazon Elastic Compute Cloud (Amazon EC2)

A Amazon foi a pioneira no mercado da computação em nuvens no ano de 2006, com o lançamento de sua plataforma de Web Services, que proviam serviços baseados na nuvem, e em 2008 com o lançamento do Amazon EC2 (*Elastic Computing Cloud*).

O EC2 baseia-se principalmente na disponibilização de máquinas virtuais individuais, conhecidas como instâncias, com uma interface de controle baseada em Webservices que visam gerar o maior controle com a menor dificuldade. Este controle pode variar para ações básicas como uma consulta, até o provisionamento automático de outras instâncias (AMAZON, 2010).

A arquitetura do EC2 é a mais heterogênea em termos de sistemas operacionais e softwares, abrangendo os mais variados cenários. Por exemplo, máquinas virtuais para ambiente Microsoft Windows Server 2008 e suporte a .Net Framework[9] ou um ambiente totalmente código aberto, utilizando Linux e Java. As instâncias do EC2 também variam quanto ao nível do hardware (processador, arquitetura, memória física e virtual) sendo classificada em pequena, grande e extragrande.

Além disso, segundo Amazon (2010), a plataforma é composta por uma série de funcionalidades que visam o aprimoramento e a facilitação da gerência das instâncias disponibilizadas. Podemos entender como principais funcionalidades:

- **CloudWatch:** Webservice que permite aos clientes monitorarem em tempo real todos os recursos da instância como hardware e rede;
- **Escalonamento Automático:** permite a mudança automática de recursos de hardware para adaptar a instâncias a mudanças no trafego da rede;

[9]A .Net Framework é o motor responsável pela compilação e execução de todos os sistemas desenvolvidos, utilizando a plataforma de desenvolvimento da Microsoft.

- **Endereço IP Elástico:** permite de mapeamento de um endereço IP para uma ou várias instâncias de forma fácil e imediata, sem a necessidade de espera de propagação de DNS.

Sobre os serviços de armazenamento do EC2, vemos que, por padrão, as instâncias contém um serviço não relacional de caráter temporário que mantém sua memória salva a re-inicializações da máquina, mas que pode vir a perder os dados gravados em caso de erros.

Para um armazenamento persistente e de caráter relacional, a Amazon disponibiliza uma série de sistemas de banco de dados dos mais variados fabricantes, variando de Microsoft SQL Server 2005, IBM DB2 até Oracle 11G e MYSQL (AMAZON, 2010).

5. Metodologia

Para este trabalho, escolheu-se uma metodologia exploratória, que buscou explorar a utilização da computação em nuvens no mercado de TI em uma visão de melhorias de processo, demonstrando aspectos da implementação das plataformas, a fim de avaliar questões de usabilidade e facilidade de programação.

Portanto, devido à vasta gama de possibilidades, limitamo-nos a amostragem de duas plataformas de mercado: Windows Azure da Microsoft e App Engine da Google, comparando-as em codificação e vantagens de usabilidade.

Para este trabalho foi utilizado uma abordagem metodológica qualitativa, descritiva não quantificável, buscando um entendimento indutivo.

6. Análise comparativa em uma visão de codificação

O objetivo principal desta pesquisa foi realizar uma comparação entre duas principais plataformas de mercado de computação em nuvens, as quais se destacam, pois suas empresas fabricantes são líderes de mercado em seus respectivos segmentos, são elas:

- Microsoft Windows Azure;
- Google App Engine.

Para esta comparação entre plataformas, foi criada uma mesma aplicação nas duas plataformas. As principais funcionalidades desta aplicação é fornecer um formulário de cadastro para usuários, gravando e retornando seus dados dentro da infraestrutura disponibilizada. Esta aplicação será composta de uma página web, com um pequeno formulário e a lista dos dados já cadastrados.

6.1. Windows Azure

A plataforma de computação nas nuvens da Microsoft, anteriormente citada, é preferencialmente indicada para aplicações desenvolvidas em .Net framework,e que sejam desenvolvidas no Visual Studio. Em caso de migração de aplicações locais para as nuvens, esta não deverá passar por grandes mudanças para ser disponibilizada, pois não existem diferenças no framework[10] utilizado para a compilação da aplicação e nem no banco de dados, tendo em vista que o Windows Azure disponibiliza um banco de dados relacional com a possibilidade de importação de dados e conexões com o banco de dados local.

Inicialmente, para testes comportamentais e de funcionalidades, não foi necessário a aquisição de uma assinatura no Windows Azure. Para possibilitar o desenvolvimento de aplicações de forma a não gerar cobranças e facilitar os testes foi disponibilizada uma plataforma de desenvolvimento integrada a IDE (ambiente de desenvolvimento integrado) Visual Studio 2008 e 2010. O SDK[11] (kit de desenvolvimento de software) do Azure é gratuito e contempla em um único instalador tanto o ambiente de simulação web (*Development Fabric),* como o ambiente relacional do SQL Azure (*Development Storage)*. Por padrão, o Visual Studio não suporta trabalhar com as tecnologias do Windows Azure, sempre sendo preciso a utilização dos SDKs e Plugins para complementá-los.

O projeto web criado pelo Visual Studio e suportado pelo Azure segue todas as características de qualquer página web que utiliza a tecnologia *.Net* com a diferença de que a página é criada em cima de uma instância chamada de *Web Role.* Com esta instância é possível realizar o provisionamento simplificado de uma ou mais instâncias.

---

[10] Framework: compreende de um conjunto de classes implementadas em uma linguagem específica, usadas para auxiliar o desenvolvimento de software.

[11] Link para download do Windows Azure SDK: http://msdn.microsoft.com/en-us/windowsazure/cc974146.aspx

6.1.1. Definido a estrutura da entidade

Conforme já foi citado neste artigo, o Windows Azure dispõe de três modelos de dados não relacionais para serem utilizados na persistência das aplicações. Para este exemplo, utilizaremos o modelo conhecido como *table* que se assemelha bastante a idéia de uma tabela de banco de dadas, só que diferencia-se principalmente no fato de não termos uma ligação entre as tabelas (relacionamento), bem como nosso modelo de dados é definido pela estrutura de nossas entidades. Outro ponto importante é que para a classe, que será persistida, faz-se necessário herdar a super classe *TableServiceEntity*. A figura 7 contém o código-fonte que define a estrutura da entidade cliente com seus atributos e propriedades.

```
Public class Cliente:TableServiceEntity {
Public Cliente(string partitionKey, string rowKey):base(partitionKey, rowKey) {}
Public Cliente():this(Guid.NewGuid().ToString(), string.Empty) {}
//Propriedades
Public Int64? CPF { get; set; }
Public Int64? RG { get; set; }
Public string Nome { get; set; }
Public DateTime? DtNascimento { get; set; }
Public string Nacionaldiade { get; set; }
Public Int64? TelCelular { get; set; }
Public Int64? TelResidencial { get; set; }}
```

Figura 7. Entidade cliente

Após a criação da entidade, que será persistido, se faz necessário criarmos o serviço de contexto, classe responsável por criar uma interface de comunicação entre a aplicação e o banco de dados, persistindo os objetos. Ressaltamos que obrigatoriamente a classe deverá herdar a classe *TableServiceContext.* Segue figura 8 com o código-fonte da classe responsável pela persistência dos dados na aplicação.

```
Public class ClienteDataContext:TableServiceContext{
Public ClienteDataContext(string enderecoBase, StorageCredentials credenciais)
: base(enderecoBase, credenciais){ }
Public const string TabelaNome = "Cliente";
Public IQueryable<Cliente> Clientes{get {return
this.CreateQuery<Cliente>(TabelaNome);}}}
```

Figura 8. Contexto de serviço

Para finalizar o desenvolvimento, criaram-se as classes e métodos que utilizam os objetos de persistência, recebendo ou enviando os dados do cliente quando necessário. Segue, na figura 9, o diagrama de classe da aplicação com as classes citadas acima mais a classe que representa a interface Web da aplicação.

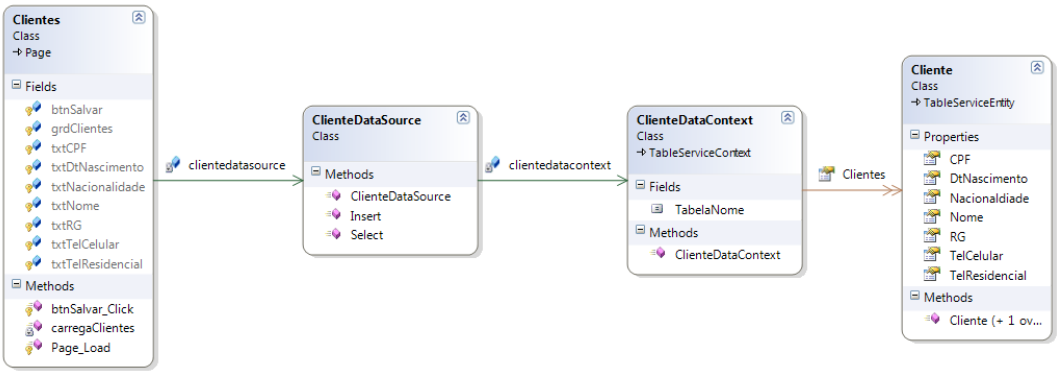


Figura 9. Diagrama de classe da Aplicação Azure

6.1.2. Testando a aplicação

A realização dos testes, conforme já foi dito, foi realizada de forma desconectada do ambiente em nuvens. Para isso, utilizamos as funcionalidades de provisionamento web do *visual Studio* junto com o *Development Fabric* e *Storage* que simulam o ambiente Azure localmente. Segue na figura 10 o formulário de cadastro da aplicação e na figura 11, após a execução do método de gravação dos dados, o retorno das informações.

Figura 10. Formulário de Cadastro

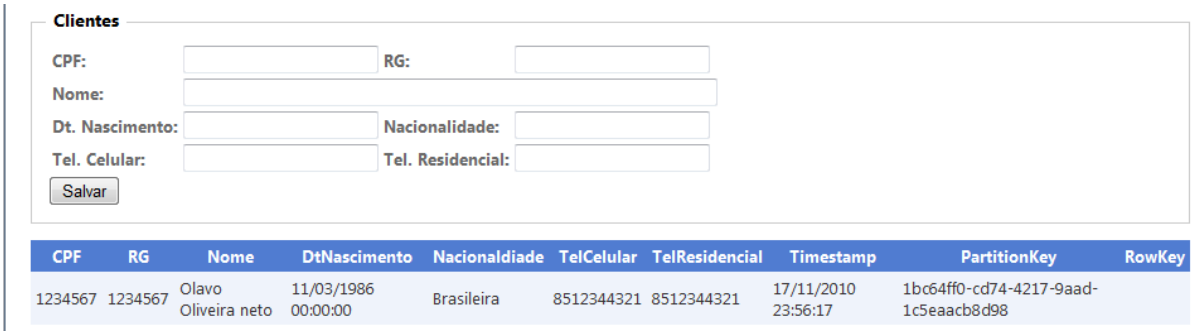


Figura 11. Aplicação trazendo os dados cadastrados

6.2. Google App Engine

O Google App Engine se destaca por ser uma plataforma simples com a possibilidade de implementação em duas linguagens abertas. Para este teste foi escolhido o Java, pois se trata de uma linguagem de mercado de forte aceitação.

O desenvolvimento para App Engine pode ser feito em qualquer IDE que tenha suporte para a linguagem Java. O Google recomenda que todo o desenvolvimento seja feito em cima da plataforma Eclipse, utilizando o *plugin* disponibilizado no site[12]. Nativamente, o Eclipse não tem suporte a implementação para o ambiente em nuvens.

As páginas do App Engine podem ser feitas em HTML puro ou em JSP[13], utilizando *servlets*[14] do lado servidor para receber e enviar as informações. Toda a aplicação pode ser executada de forma local, utilizando o servidor de aplicações disponibilizado junto ao Eclipse ou publicando nos servidores do Google de forma gratuita.

6.2.1. Persistindo informações

A plataforma do Google contém uma sistemática de persistência que se assemelha muito a já citada no Windows Azure. Neste caso, também persistimos a informação de forma não relacional através do armazenamento do estado de objetos em memória. Ainda de forma semelhante, tivemos que criar para isto duas classes, uma que representasse a entidade cliente com algumas peculiaridades da plataforma como um atributo chamado *key,* que será a representação da chave primária da entidade após o armazenamento, e outra classe para persistência dos dados. Segue, na figura 12, o código que representa o objeto Java (JDO) que será persistido.

```
Package entidades;
import java.util.Date;
import javax.jdo.annotations.IdGeneratorStrategy;
import javax.jdo.annotations.PersistenceCapable;
import javax.jdo.annotations.Persistent;
import javax.jdo.annotations.PrimaryKey;
@PersistenceCapable public class Cliente_JDO {
@PrimaryKey @Persistent(valueStrategy = IdGeneratorStrategy.IDENTITY)
private Key key;
@Persistent private String cpf;
@Persistent private String rg;
@Persistent private String nome;
@Persistent private Date dtnascimento;
@Persistent private String nacionalidade;
@Persistent private String telcelular;
@Persistent private String telresidencial;
public Cliente_JDO(String nome, String cpf, String rg, Date dtnascimento, String
nacionalidade, String telcelular, String telresidencial){
       this.nome = nome; this.cpf = cpf; this.rg = rg; this.dtnascimento =
dtnascimento;
       this.nacionalidade = nacionalidade; this.telcelular = telcelular;
```

[12] Link para download: http://code.google.com/eclipse/docs/users_guide.html
[13] Java Server Pages, linguagem web para o desenvolvimento de aplicações web.
[14] Componente java responsável por enviar e receber dados da camada web.

```
this.telresidencial = telresidencial;}
public Key getKey(){return key;}
public String getNome(){return nome;}
public String getCPF(){return cpf;}
public String getRG(){return rg;}
public Date getdtNascimento(){return dtnascimento;}
public String getNacionalidade(){return nacionalidade;}
public String getTelCelular(){return telcelular;}
public String getTelResidencial(){return telresidencial;}}
```

Figura 12. Entidade cliente

A classe de persistência necessita utilizar um objeto do tipo *PersistenceManagerFactory,* classe que entende o contexto no qual a aplicação está rodando e automaticamente já realiza o armazenamento do objeto no servidores do Google, não existindo a necessidade de apontamento de base de dados. A figura 13 demonstra o código da classe de persistência que retorna uma instância do objeto necessário para as ações no banco de dados.

```
Package appEngine;
Import javax.jdo.JDOHelper;
Import javax.jdo.PersistenceManagerFactory;
Public final class PMF {private static final PersistenceManagerFactory pfmInstance
= JDOHelper.getPersistenceManagerFactory("transactions-optional");
private PMF(){}
public static PersistenceManagerFactory get(){return pfmInstance;}}
```

Figura 13. Classe de persistência

As classes da aplicação apesar de longas, são simples e não envolvem uma programação muito complexa. Segue abaixo, na figura 14, o diagrama de classe da aplicação desenvolvida com um detalhe importante: todas as classes são instâncias a partir do *servlet*.

Figura 14. Diagrama de classe da aplicação Java

6.2.2. Consultando os dados

A consulta dos objetos persistidos é realizada através da classe PMF que foi embutida no código HTML da página para que, de forma dinâmica, pudesse montar a página. Segue codificação na figura 15, demonstrando principalmente que a

consulta é feita utilizando comandos parecidos com os utilizados nos bancos de dados tradicionais.

```
<%PersistenceManager 20pm = PMF.get().getPersistenceManager();
String query = "select from "+Cliente_JDO.class.getName();
List<Cliente_JDO>clientes = (List<Cliente_JDO>)pm.newQuery(query).execute();
if(clientes.isEmpty()){%>
<p>Nenhum cliente cadastrado.</p>
<%}else%>
<table><thead><tr><td>Nome</td><td>RG</td><td>CPF</td></tr></thead>
<tbody>
<%{for(Cliente_JDO c : clientes){%>
<tr>
<td><%=c.getNome() %></td>
<td><%=c.getRG() %></td>
<td><%=c.getCPF() %></td>
</tr>
<%}}pm.close();%>
</tbody></table></div></form>
```

Figura 15. Consulta dos dados cadastrados

6.2.3. Executando a aplicação

O teste da aplicação pode ser feito de forma local, utilizando o servidor de aplicações do Google ou através da publicação facilitada nos servidores do App Engine. A figura 16 mostra o formulário de cadastro e o retorno de dados salvos.

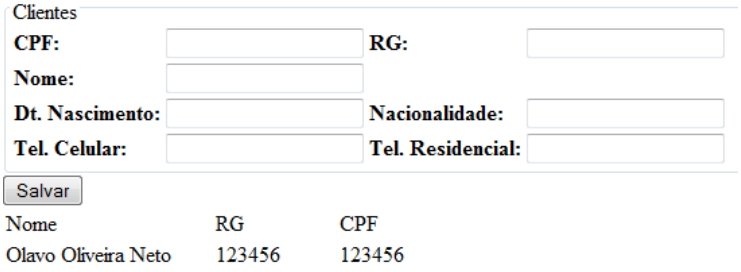


Figura 16. Tela da aplicação App Engine

6.3. Resultados

Nos estudos realizados, observamos que ambas as plataformas oferecem recurso parecidos, por exemplo: uma IDE especializada para o desenvolvimento, integrações com outros produtos e serviços, classes especiais voltadas ao monitoramento das aplicações publicadas e armazenamento não relacional. Para os testes realizados, concluímos que as plataformas funcionam de maneira muito parecida no desenvolvimento das aplicações, testes e publicação.

Essas semelhanças fazem com que cada plataforma se destaque em seus devidos ambientes, App Engine para aplicações Java e Azure para .Net.

De certa forma, as plataformas estão no mesmo nível de funcionalidade como de desenvolvimento. Fica a vantagem da Microsoft em cima da Google no quesito

banco de dados, pois o fornecimento de um ambiente banco de dados relacional é importante para as empresas, principalmente as que desejam migrar aplicações de um ambiente local para nuvens com o mínimo de alteração no código-fonte. O Google, por ser gratuito inicialmente, é destinado às empresas que desejam criar novos aplicativos já no ambiente Web, sem custos, e com toda garantia de serviço que a computação em nuvens dispõe.

7. Considerações finais

Este trabalho mostrou os benefícios da computação nas nuvens para o mercado de TI (Tecnologia da Informação) atual e demonstrou os serviços providos pela Microsoft, com o Windows Azure: o Google, com o App Engine e a Amazon, com o EC2. Também apresentou um estudo comparativo de funcionalidade entre as plataformas do Google e da Microsoft.

Como foi visto, a computação nas nuvens é um passo a ser dado na evolução da informática que vem possibilitando a disponibilização dos recursos não só nos navegadores independente de localização, mas também na entrega de sistemas como bens de consumo essenciais que serão pagos por uso e utilizados pelos mais variados tipos de dispositivos.

Segundo Brodkin (2008), encontramos como resistência a computação nas nuvens o medo que as empresas possam ter de que seus dados armazenados sejam violados ou utilizados pelo provedor do serviço em benefício próprio. Tratar estes aspectos de segurança, principalmente no que se diz respeito à visão do cliente quanto ao sigilo de seus dados, deverá ser o próximo passo para os provedores no intuito de desmitificar esses mitos e aumentar a aderência à plataforma.

A partir da pesquisa realizada neste artigo, poderão ser realizados estudos de codificação mais avançados, por exemplo: padrões de projeto para aplicações web, ou estudos das demais vantagens que as plataformas aqui citadas possam conter.

REFERÊNCIAS

AMAZON. Amazon Elastic Compute Cloud (Amazon EC2). **Amazon Web Services**, 2010. Disponivel em: <http://aws.amazon.com/ec2/#highlights>. Acesso em: 20 agosto 2010.

BRODKIN, J. Gartner:Seven cloud-computing security risks. **Security Central**, 2 Julho 2008. Disponivel em: <http://www.infoworld.com/d/security-central/gartner-seven-cloud-computing-security-risks-853>. Acesso em: 19 agosto 2010.

BUYYA, R. et al. Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility, Melbourne, 2009.

COMPUTAÇÃO em nuvem. **Wikipedia**. Disponivel em: <http://pt.wikipedia.org/wiki/Computa%C3%A7%C3%A3o_em_nuvem#Refer.C3.AAncias>. Acesso em: 18 agosto 2010.

COMPUTERWORLD US. Centralização de TI: modelo volta a ser usado pelas empresas. **CIO - Estratégias de negócios e TI para lideres corporativos**, 10 Março 2010. Disponivel em: <http://cio.uol.com.br/gestao/2010/03/09/centralizacao-de-ti-modelo-volta-a-ser-usado-pelas-empresas/>. Acesso em: 2010 setembro 20.

GARTNER. Gartner Says Cloud Computing Will Be As Influential As E-business. **Gartner**, 26 Junho 2008. Disponivel em: <http://www.gartner.com/it/page.jsp?id=707508>. Acesso em: 18 agosto 2010.

GOOGLE. Run your web apps on Google's infrastructure. **Google**, 2010. Disponivel em: <http://code.google.com/intl/en/appengine/>. Acesso em: 20 agosto 2010.

HOWE, D. Application Program Interface. **Free On-Line Dictionary Of Computing**, 2 Fevereiro 1995. Disponivel em: <http://foldoc.org/Application+Program+Interface>. Acesso em: 30 agosto 2010.

HURWITZ, J.; BLOOR, R.; KAUFMAN, M. **Cloud Computing for DUMMIES, HP Special Edition**. Indianapolis: Wiley Publishing, INC., 2010.

IBM. RESTful Web services: The basics. **IBM**, 06 Novembro 2008. Disponivel em: <https://www.ibm.com/developerworks/webservices/library/ws-restful/>. Acesso em: 3 setembro 2010.

LIMA, G. A quantidade de informação gerada no mundo vs a qualidade. **Coruja d TI**, 20 Setembro 2010. Disponivel em: <http://blog.corujadeti.com.br/a-quantidade-de-informacao-gerada-no-mundo-vs-a-qualidade/>. Acesso em: 8 outubro 2010.

LOCAWEB. Cloud Computing. **Locaweb**, 2010. Disponivel em: <http://www.locaweb.com.br/solucoes/cloud-computing.html>. Acesso em: 20 agosto 2010.

MICROSOFT. **Windows Azure™ Security Overview**. [S.l.]. 2010.

MICROSOFT MSDN - WINDOWS AZURE. Patterns para computação em nuvem. **Windows Azure**, 2009. Disponivel em: <http://msdn.microsoft.com/pt-br/windowsazure/ff384162.aspx>. Acesso em: 19 agosto 2010.

MICROSOFT MSDN. Cloud Computing e o Windows Azure. **Windows Azure**, 2010. Disponivel em: <http://msdn.microsoft.com/pt-br/windowsazure/dd637844.aspx>. Acesso em: 19 agosto 2010.

MOHAMED, A. A history of cloud computing. **ComputerWeekly.com**, 27 Março 2009. Disponivel em: <http://www.computerweekly.com/Articles/2009/06/10/235429/A-history-of-cloud-computing.htm>. Acesso em: 25 agosto 2010.

RHOTON, J. **Cloud Computing Explained:** Implementation Handbook for Enterprises. [S.l.]: Recursive Press, 2009.

SNEDAKER, S. **Business Continuity and Disaster Recovery Planning for IT Professionals**. Burligton: Syngress, 2007.

SOUSA, F. R. C.; MOREIRA, L. O.; MACHADO, J. C. **Computação em Nuvem: Conceitos, Tecnologias, Aplicações e Desafios**. Fortaleza. 2009.

SUN MICROSYSTEMS. Introduction to Cloud Computing architecture White Paper, Santa Clara, junho 2009.

VECCHIOLA, C.; CHU, X.; BUYY, R. **Aneka: A Software Platform for.NET-based Cloud Computing**. Melbourne. 2009.