



Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Engenharia de Software

Implantação de Softwares como Serviço em uma Nuvem Privada para a Fábrica de Software da FGA

Autor: Guilherme Fay Vergara
Orientador: Sérgio Antônio Andrade de Freitas, Prof. Dr.

Brasília, DF
2014



Guilherme Fay Vergara

Implantação de Softwares como Serviço em uma Nuvem Privada para a Fábrica de Software da FGA

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Sérgio Antônio Andrade de Freitas, Prof. Dr.

Coorientador: Edna Dias Canedo, Profa. Dra.

Brasília, DF

2014

Guilherme Fay Vergara

Implantação de Softwares como Serviço em uma Nuvem Privada para a Fábrica de Software da FGA/ Guilherme Fay Vergara. – Brasília, DF, 2014-

73 p. : il. (algumas color.) ; 30 cm.

Orientador: Sérgio Antônio Andrade de Freitas, Prof. Dr.

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB
Faculdade UnB Gama - FGA , 2014.

1. Computação em Nuvem. 2. Fábrica de Software. I. Sérgio Antônio Andrade de Freitas, Prof. Dr. . II. Edna Dias Canedo, Profa. Dra.. III. Universidade de Brasília. IV. Faculdade UnB Gama. V. Implantação de Softwares como Serviço em uma Nuvem Privada para a Fábrica de Software da FGA

CDU

Guilherme Fay Vergara

Implantação de Softwares como Serviço em uma Nuvem Privada para a Fábrica de Software da FGA

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Trabalho aprovado. Brasília, DF, 25 de Junho de 2014:

Sérgio Antônio Andrade de Freitas,
Prof. Dr.
Orientador

Edna Dias Canedo, Profa. Dra.
Coorientadora

Marcos Martins Melo, SERPRO
Convidado 1

Brasília, DF
2014

Dedico este trabalho à minha família, por sua capacidade de acreditar e investir em mim. Mãe, seu cuidado e dedicação foi que deram, em alguns momentos, a esperança para seguir. Pai, sua presença significou segurança e certeza de que não estou sozinho nessa caminhada.

Agradecimentos

Agradeço primeiramente a Deus por ter me dado saúde e força para superar as dificuldades.

Ao SERPRO e ao CEBRASPE por serem parceiros neste trabalho.

Aos meus orientadores Sérgio e Edna, pelo suporte no desenvolvimento deste trabalho, pelos incentivos, correções e puxões de orelha durante todo este processo, não somente por terem me ensinado, mas por terem me feito aprender, meus sinceros agradecimentos.

A minha mãe, que me deu apoio e incentivo nas horas difíceis, de desânimo e abatimento. Ao meu pai que apesar de todas as dificuldades sempre esteve ao meu lado me dando conselhos e sempre servindo de inspiração. E ao meu irmão pela sua amizade e ajuda nos estudos, e também pela ajuda nas correções de tradução deste trabalho. Meus eternos agradecimentos.

E a todos que direta ou indiretamente fizeram parte da minha formação, o meu muito obrigado.

*Se A é o sucesso, então A é igual a X mais Y mais Z.
O trabalho é X; Y é o lazer; e Z é manter a boca fechada.
(Albert Einstein)*

Resumo

O objetivo deste trabalho é trazer ao leitor primeiramente o estado da arte sobre computação em nuvem, posteriormente, uma breve indicação de ferramentas para computação em nuvem, passando por todos os níveis de arquitetura, e posteriormente a proposição de um modelo de implantação de computação em nuvem para a fábrica de software da faculdade do Gama e por último, a implantação de parte deste modelo. A implantação destes serviços é dentro de uma nuvem privada, alocada dentro dos servidores do CEBRASPE. Os principais serviços disponíveis são: O expresso que é um sistema que é mantido principalmente pelo SERPRO, e o Owncloud um software que implanta o compartilhamento de arquivos na nuvem. Estes serviços servem principalmente para aumentar a produtividade dos membros da fábrica, e aumentar a sua colaboratividade nos projetos desenvolvidos.

Palavras-chaves: Computação em nuvem. SaaS. Fábrica de Software.

Abstract

The objective of this final course design is bring the reader first state of the art on cloud computing later a brief explanation of tools for cloud computing, through all levels of architecture, and later the proposition of a deployment model of computing cloud to the software factory and finally the deployment of this model. The deployment of such services is within a private cloud, allocated within the CEBRASPE servers. The main services are available: Express which is a system that is primarily maintained by SERPRO, and owncloud, software that implements the sharing of any files in the cloud. These services increase the productivity of the factory members and increase their collaboration in projects developed internally.

Key-words: Cloud Computing. SaaS. Software Factory

Lista de ilustrações

Figura 1 – Três Elementos da Solução de Computação em Nuvem (VELTE; VELTE; ELSENPETER, 2009).	28
Figura 2 – Arquitetura da Computação em Nuvem (ZHOU et al., 2010)	31
Figura 3 – Papeis da computação em nuvem (MARINOS; BRISCOE, 2009b)	33
Figura 4 – Xen client	50
Figura 5 – Log-in no Expresso	51
Figura 6 – Tela principal do Expresso	52
Figura 7 – Log-in no Owncloud	52
Figura 8 – tela principal do Owncloud	53
Figura 9 – Processo de implantação (VERGARA; CANEDO; FREITAS, 2014)	54
Figura 10 – Papéis da computação em nuvem (VERGARA; CANEDO; FREITAS, 2014)	57
Figura 11 – Arquitetura física de Serviços (VERGARA; CANEDO; FREITAS, 2014)	58
Figura 12 – Arquitetura lógica de Serviços (VERGARA; CANEDO; FREITAS, 2014)	60

Lista de abreviaturas e siglas

IaaS	Infraestrutura como serviço
PaaS	Plataforma como serviço
SaaS	Software como serviço
VM	Máquina Virtual
TI	Tecnologia da Informação
SERPRO	Serviço Federal de Processamento de Dados
CEBRASPE	Centro Brasileiro de Pesquisa em Avaliação e Seleção e de Promoção de Eventos
TCC	Trabalho de conclusão de curso

Sumário

1	INTRODUÇÃO	23
1.1	Motivação	24
1.2	Objetivos do trabalho	24
1.3	Metodologia de pesquisa	24
1.4	Organização do trabalho	24
2	ESTADO DA ARTE E REVISÃO BIBLIOGRÁFICA	27
2.1	A computação em nuvem	27
2.1.1	Introdução	27
2.1.2	Elementos que compõe a nuvem	28
2.1.3	Características da Computação em Nuvem	28
2.1.4	Vantagens e Desvantagens	30
2.1.5	Arquitetura da computação em nuvem	30
2.1.5.1	Infraestrutura como Serviço	30
2.1.5.2	Plataforma como Serviço	32
2.1.5.3	Software como Serviço	32
2.1.6	Papéis na Computação em Nuvem	32
2.1.7	Modelos de implantação	33
2.1.7.1	Modelo privado	33
2.1.7.2	Modelo comunidade	34
2.1.7.3	Modelo público	34
2.1.7.4	Modelo Híbrido	34
2.1.8	Distribuição de arquivos na nuvem	34
2.1.8.1	Google File System (GFS)	35
2.1.8.2	Amazon S3	35
2.1.8.3	Microsoft Azure	36
2.2	Redes de Computadores e Internet	37
2.2.1	A camada de Transporte	37
2.2.1.1	TCP/IP	37
2.2.1.2	LDAP	37
2.2.2	A camada de Aplicação	38
2.2.2.1	DNS	38
2.2.2.2	HTTP	39
2.2.3	Segurança de Redes	39
2.2.3.1	VPN	39

2.2.3.2	Criptografia	39
2.2.3.3	<i>Firewall</i>	40
2.3	Discussões sobre segurança da informação	40
2.3.0.4	Os cinco riscos de segurança aos quais devemos ficar atentos	41
2.3.0.5	Como reduzir os riscos	42
2.4	A Fábrica de Software	43
3	FERRAMENTAS PARA COMPUTAÇÃO EM NUVEM	45
3.1	IAAS	45
3.1.1	Microsoft Windows AZURE	45
3.1.2	OpenStack	46
3.1.3	Xen	46
3.1.4	Eucalyptus	46
3.2	PAAS	47
3.2.1	Tsuru	47
3.2.2	Heroku	47
3.3	SAAS	48
3.3.1	Owncloud	48
3.3.2	Expresso	48
4	O MODELO DE IMPLANTAÇÃO PARA A FÁBRICA DE SOFTWARE	49
4.1	Modelo IaaS	49
4.2	Modelo PaaS	50
4.3	Modelo SaaS	50
4.4	Processo	53
5	A IMPLANTAÇÃO	57
5.1	Implantação	57
5.1.1	Papéis	57
5.1.2	Arquitetura Física	57
5.1.2.1	Servidores	57
5.1.2.2	<i>Firewalls</i>	59
5.1.3	Arquitetura de Serviços	59
5.2	Análise da implantação	59
6	CONCLUSÃO E TRABALHOS FUTUROS	61
6.1	Trabalhos Futuros	61
	Referências	63

	APÊNDICES	65
	APÊNDICE A – PRIMEIRO APÊNDICE	67
A.1	Detalhes dos servidores da nuvem	67
	APÊNDICE B – INSTALAÇÕES	69
B.1	Instalação redmine	69
B.2	Integração Moodle com AD	70
B.3	Integração OPENVPN com AD	70
B.4	Integração OWNCLOUD com AD	73

1 Introdução

Com o avanço tecnologia, serviços básicos e essenciais são quase todos entregues de uma forma completamente transparente. Serviços de utilidade pública como água, eletricidade, telefone e gás tornaram-se fundamentais para nossa vida diária e são explorados por meio do modelo de pagamento baseado no uso (VECCHIOLA; CHU; BUYYA, 2009).

As infraestruturas existentes permitem aos seus usuários receber tais serviços em praticamente qualquer lugar, e a qualquer hora, da forma que, quando precisamos de água, simplesmente abrimos a torneira e lá está, ou ao acender a luz já estamos consumindo um serviço. O uso destes serviços é cobrado conforme o uso, podendo o cliente pagar somente pelo que foi utilizado. Recentemente a mesma ideia vem sendo pesquisada e aplicada na computação, e uma mudança consistente neste sentido tem sido feita com a disseminação da *Cloud Computing* ou Computação em Nuvem.

A computação em nuvem é um novo paradigma da computação cujo o principal objetivo é entregar ao cliente final serviços de informática através da internet pelo modelo de pagamento onde o cliente pagará somente pelo o que usar.

Instituições de grande porte já enxergavam que era mais barato, prático e fácil contratar serviços à medida que lhes fosse necessário ao invés de investir em parques tecnológicos que muitas das vezes ficavam obsoletos sem que ao menos fossem utilizados. As primeiras a investirem nessa ideia foram grandes como IBM, Google e Microsoft. É um mercado muito recente, surge em meados dos anos 90, e em alto desenvolvimento e crescimento. Dado seu elevado potencial e complexidade e tendo em vista a crescente adoção da computação em nuvem pelo mercado, torna-se notória a necessidade desse conhecimento ser difundido, umas dessas maneiras e tornando mais viável a criação e simulação de nuvens, mesmo por aqueles que não dominam ainda o assunto.

A computação em nuvem é comercializada pelo sistema *Pay-per-use*, significa na prática que a empresa provedora da infraestrutura da nuvem será paga de acordo com a necessidade do cliente em relação à quantidade de Processamento, quantidade de memória do tipo RAM, quantidade de memória para armazenamento, tipos de serviços prestados e softwares na nuvem. Os tipos de serviços mais comuns são Infraestrutura como Serviço (IaaS), Plataforma como Serviço (Paas) e Software como Serviço (SaaS). Os usuários não precisam se preocupar com backups, pois se os componentes falharem, o provedor é responsável por substituí-los e tornar os dados disponíveis em tempo hábil por meio de réplicas (BRANTNER et al., 2008).

1.1 Motivação

As motivações estão baseadas em duas esferas centrais. Uma delas é a curiosidade por uma das áreas da computação que mais cresce nos últimos anos, segundo o *Gartner*. A segunda esfera está relacionada ao estudo de caso deste Trabalho de Conclusão de Curso, que é a fábrica de software do gama. Juntando estas duas motivações temos a motivação central, que é fornecer a fábrica um ambiente mais seguro e colaborativo do que as soluções atuais existentes.

1.2 Objetivos do trabalho

A computação em nuvem é uma área que vem cada vez mais se desenvolvendo atraindo diversos públicos, cada vez mais as organizações vêm adotando soluções baseadas em computação em nuvem. Além de manter o foco na implantação de uma nuvem privada no contexto da fábrica de software, os objetivos deste trabalho podem ser resumidos em:

- Fazer um estudo das tecnologias existentes em computação em nuvem com aplicação para a fábrica de software da faculdade do gama.
- Criar um modelo de implantação para a fábrica de software da faculdade do gama.
- Contribuir para um ambiente colaborativo para a fábrica de software.

1.3 Metodologia de pesquisa

A metodologia de pesquisa proposta foi dividida em três grandes fases. Primeiramente uma fase para levantamento bibliográfico e estudo sobre computação em nuvem, o que é, quais seus tipos, como é sua arquitetura e etc. Posteriormente passou-se para uma fase de aplicar esse conhecimento adquirido para um estudo de caso real, no caso a fábrica de software do gama, nesta fase foi feita uma avaliação de que tipo de serviços e qual os respectivos softwares que beneficiariam a fábrica nas suas atividades, pensou-se em soluções entre as três principais áreas da computação em nuvem, e por dentre todas as ferramentas analisadas foi escolhida a solução que trouxesse o melhor custo benefício para a fábrica, tendo como parâmetros facilidade de implantação X relevância X tempo para implantação.

1.4 Organização do trabalho

Para uma melhor compreensão deste trabalho explica-se agora a sua organização.

No capítulo 2 é feita uma revisão dos conceitos de Computação em Nuvem e Redes de Computadores e Internet.

O capítulo 3 por sua vez traz um apanhado geral de ferramentas para implantação da computação em nuvem.

No capítulo 4 é apresentado um modelo de implantação para a fábrica de software, trazendo um modelo para cada tipo de implantação.

E finalmente no capítulo 5 é explicado como realmente foi feita a implantação, ou seja, a implantação em si.

2 Estado da arte e revisão bibliográfica

2.1 A computação em nuvem

Esta seção tem como foco fazer uma revisão bibliográfica acerca dos principais conceitos de computação em nuvem. Na seção 2.1.2 apresenta-se as principais definições e características da computação em nuvem na seção 2.1.3, e quais suas vantagens e desvantagens na seção 2.1.4, posteriormente na seção 2.1.5 define-se como a arquitetura da nuvem pode ser dividida. Para a divisão de responsabilidades para a nuvem define-se papéis, e para tal eles são descritos na seção 2.1.6. Na seção 2.1.7 descreve-se quais os possíveis modelos de implantação da nuvem, e finalmente na seção 2.1.8 apresenta-se uma breve introdução sobre como é a distribuição de arquivos na nuvem.

2.1.1 Introdução

Existe na literatura diversas definições de computação em nuvem apresenta-se aqui duas delas. O termo computação nas nuvens (do inglês *cloud computing*) segundo o [NIST] "a computação em nuvem é um modelo para acesso conveniente, sob demanda, e de qualquer localização, a uma rede compartilhada de recursos de computação (isto é, redes, servidores, armazenamento, aplicativos e serviços) que possam ser prontamente disponibilizados e liberados com um esforço mínimo de gestão ou de interação com o provedor de serviços".

De acordo com (BUYA; BROBERG; GOSCINSKI, 2010) a definição de computação em nuvem é a seguinte: "Nuvem é um sistema de computação paralela e distribuída que consiste em um conjunto de computadores interligados e virtualizados que são dinamicamente providos e apresentados como um ou mais recursos de computação unificada baseada em contratos de níveis de serviço estabelecidos através de negociação entre o prestador de serviço e os consumidores."

Ambas as definições expressam da maneira bem clara que uma das características marcantes da nuvem é a entrega de serviços de maneira bem simples e rápida para os clientes interessados. Esta característica foi um dos principais motivos para o desenvolvimento desta área, pois os clientes queriam de maneira cada vez mais rápida o acesso a recursos de software e hardware, e é neste domínio em que a computação em nuvem entra, pois para entregar estes serviços de software e hardware de maneira mais rápida apelou-se para a internet e o seguinte questionamento surgiu: "e se pudéssemos disponibilizar um recurso de hardware, por exemplo um computador inteiro, para o meu cliente através da internet?" Bem, isto se tornou possível graças a evolução de tecnologias como

a virtualização e a internet.

2.1.2 Elementos que compõe a nuvem

Uma solução de computação em nuvem é composta de vários elementos, conforme apresentado na Figura. Estes elementos formam as três partes de uma solução de computação em nuvem: cliente, datacenter e servidores distribuídos (VELTE; VELTE; ELSENPETER, 2009). Cada elemento tem uma finalidade e possui um papel específico em oferecer um aplicativo funcional baseado em nuvem, como representado na Figura 1.

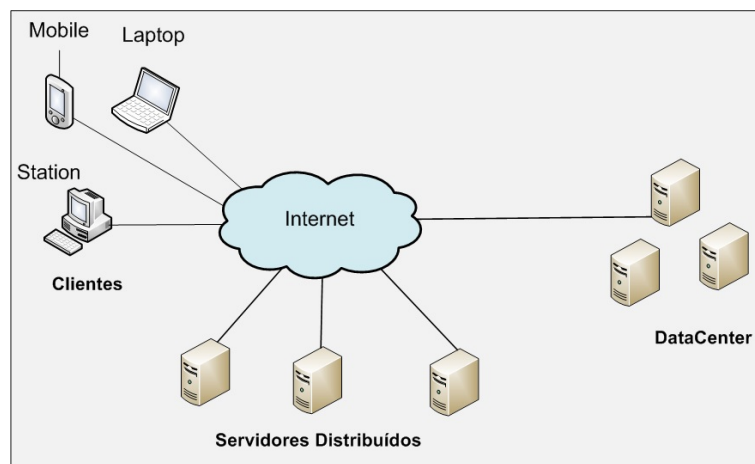


Figura 1: Três Elementos da Solução de Computação em Nuvem (VELTE; VELTE; ELSENPETER, 2009).

- Clientes - Os clientes são em uma arquitetura de computação em nuvem, exatamente o que eles são em uma rede local (LAN). São os dispositivos com os quais os usuários finais interagem para gerenciar sua informação na nuvem.
- Data Center - É um conjunto de servidores onde o aplicativo (CRM, ERP, financeiro, etc) é armazenado. Uma tendência crescente no mundo da TI é a virtualização de servidores, isto é, o software pode ser instalado permitindo que vários servidores virtuais sejam usados.
- Servidores Distribuídos - Os servidores não têm que estar alocados em um mesmo local. Normalmente, os servidores estão em diferentes posições geográficas, o que permite ao provedor de serviços maior flexibilidade nas opções e na segurança, por exemplo, a Amazon possui uma solução de nuvem no mundo inteiro. Se algo acontecer em um local, causando uma falha, o serviço poderá ser acessado através de outro local. Novos hardwares podem ser adicionados à nuvem, conforme a necessidade.

2.1.3 Características da Computação em Nuvem

Segundo (BADGER et al., 2011) A computação em nuvem possui algumas características importantes, as principais s seguintes:

1. Autoatendimento sob demanda (*On-Demand Self-Service*)

Este talvez seja uma das principais características da nuvem, este aspecto traz a nuvem a capacidade de prover ao consumidor, caso ache necessário, a opção de aumentar ou diminuir as capacidades computacionais alocadas como tempo de servidor, armazenamento de rede, tudo isso sem precisar de interação humana com o provedor de serviços.

2. Amplo acesso à rede (*Ubiquitous Network Access*) Todos os serviços da nuvem podem ser acessados de maneira padronizada pela grande maioria de aparelhos que possuem acesso a internet. A nuvem se utiliza de mecanismos padrões que promovem o uso de plataformas heterogêneas. Este tipo de acesso facilita o uso dos usuários que já estão acostumados com certo padrão de interface, e facilita aos programadores, pois não precisam desenhar uma interface nova para cada tipo de dispositivo

3. Pool de Recursos (*Resource Pooling*) Este modelo visa atender a múltiplos clientes simplesmente atribuindo ou retribuindo os recursos virtuais dinamicamente conforme sua demanda através do modelo multi-inquilino (textitmulti-tenancy). O serviço disponibilizado na nuvem, principalmente no modelo SaaS, tem que ser capaz de gerenciar a adesão de novos clientes, enquanto por exemplo, garante a segurança de dados (para que um cliente não acesse os dados dos outros) ou a escalabilidade (para que a infraestrutura suporte o aumento de carga inerente a novas adesões) O cliente não possui um controle ou conhecimento sobre o local onde seus recursos estão sendo alocados, tendo somente uma informação mais ampla como o país em que se encontra, o estado ou o Data Center, Os tipos de recursos são: armazenamento, processamento, memória, banda e máquinas virtuais.

4. Transparência O usuário pode não conhecer onde estão alocados seus recursos computacionais, sejam eles VM's, arquivos, programas, etc., tendo uma visão somente da "ponta do iceberg", ou seja a nuvem mascara toda a infraestrutura e oferece ao cliente somente uma interface para acesso, tornando assim tudo transparente.

5. Elasticidade Rápida (*Rapid Elasticity*) A elasticidade está associada a necessidade de se aumentar ou diminuir os recursos alocados rapidamente. Muitas vezes é feito de maneira automática, visando um maior aproveitamento das capacidades disponíveis. Por vezes esta característica da a ilusão de recursos computacionais infinitos. Essa é uma das principais características que tornam a computação em nuvem um serviço muito atrativo.

6. Serviços Mensuráveis (*Measured Service*) Todos os serviços podem ser monitorados e controlados automaticamente pela nuvem, isso permite que o fornecedor da nuvem pode cobrar do cliente exatamente o que foi usado por ele, e ao cliente ajuda-o a controlar seus gastos.

2.1.4 Vantagens e Desvantagens

As grandes vantagens da computação em nuvem são que o cliente poderá ter acesso a arquivos/aplicações em qualquer lugar, bastando ter uma conexão com a internet. Os custos com Hardware são menores, pois o cliente não precisa mais de um supercomputador para executar sua aplicação, quem deve se preocupar com isso agora é o *Back End*. Os clientes também não precisam mais se preocupar com compra de licença de softwares para cada um dos terminais que possuir.

A grande preocupação em relação à computação em nuvem hoje está relacionada a segurança e privacidade. Os arquivos dos clientes, muitas vezes altamente sigilosos, não são mais armazenados nos computadores da própria empresa, estão armazenados nos computadores dos servidores prestadores da infraestrutura da nuvem.

2.1.5 Arquitetura da computação em nuvem

Segundo (DIKAIKOS et al., 2009) a arquitetura da computação em nuvem pode ser dividida em três camadas abstratas, a camada de infraestrutura é a camada mais baixa, e é através dela que os prestadores de infraestrutura disponibilizam os serviços de rede e armazenamento da nuvem, dessa forma, fazem parte dela servidores, sistemas de armazenamento, como os *data centers*, e roteadores, por exemplo.

A camada de plataforma possui uma abstração mais elevada e provê serviços para que as aplicações possam ser desenvolvidas, testadas, implementadas e mantidas no ambiente da nuvem pelos prestadores de serviços. Finalmente, a camada de aplicação é a de mais alto nível de abstração, e aquela que oferece diversas aplicações como serviços para os usuários.

A figura 2 apresenta alguns serviços de computação como, rede, servidores, sistemas operacionais e etc. No primeiro arranjo temos a infraestrutura como serviço IaaS, nessa estrutura o cliente do serviço não precisa se preocupar com toda a parte de hardware e rede necessária para uma infraestrutura comum, sendo estes providos pelo provedor de serviços na nuvem. Um nível acima de abstração tem a PaaS, que muito semelhante ao anterior mascara toda a camada de hardware do usuário, porém neste por exemplo caso o usuário não controla os *webservers* usados, como (IIS, JBoss, Tomcat, etc.) deixando o usuário com foco na gerencia da aplicação e dados. E por último o SaaS que provavelmente é o mais comumente usado, neste caso o usuário não precisa se preocupar com nada, ele

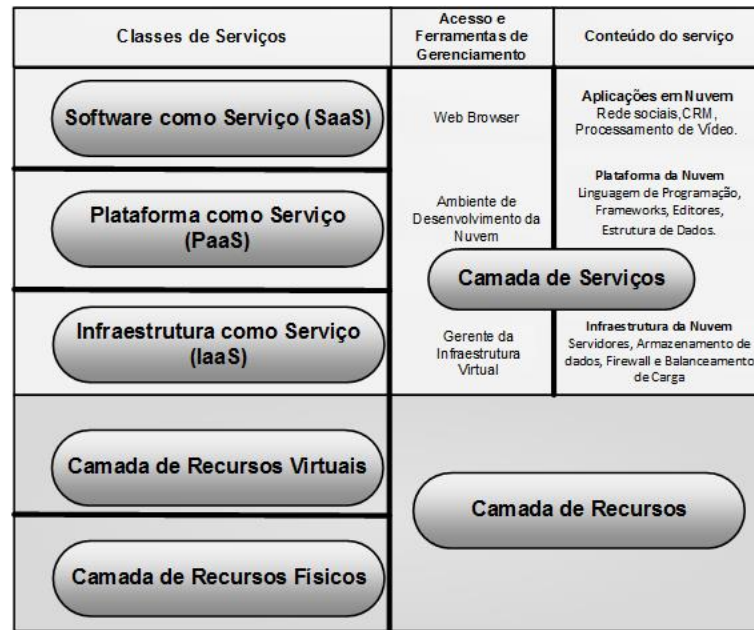


Figura 2: Arquitetura da Computação em Nuvem (ZHOU et al., 2010)

somente usa o sistema. A seguir é explicado cada camada de serviço.

2.1.5.1 Infraestrutura como Serviço

IaaS – *Infrastructure as a Service* – Conhecido também como HaaS *Hardware as a service* - É o tipo de modelo que oferece a virtualização de recursos tanto computacionais como de armazenamento e de comunicação. Permite prover serviços sobre demanda, que podem ser executados por diferentes sistemas operacionais e permite uma pilha de aplicativos customizáveis. Esse serviço oferece máquinas virtuais com pilha de aplicativos customizáveis, da mesma forma que um servidor físico comum pode ser customizado.

Este tipo de serviço elimina as preocupações com servidores, sistemas operacionais, armazenamento e etc. Vide figura 1, assim os desenvolvedores concentram-se mais na produção e não na infraestrutura, permitindo um desenvolvimento mais rápido e reduz bastante os gastos da empresa com tais recursos computacionais.

O termo IaaS refere-se a um tipo de serviço baseado em técnicas de virtualização de recursos computacionais oferecidas através da internet. Tais recursos podem ser escalados conforma a necessidade do cliente, e esta característica torna este serviço muito atrativo, pois os clientes pagam somente pelo que usarem, não pagando pelo tempo que seus servidores ficam ociosos.

Sua utilização é recomendada quando se tem uma demanda volátil, como por exemplo em lojas virtuais, que vendem muito no natal, e no resto do ano há pouca procura. Também é aconselhável para empresas que tendem a crescer rapidamente e não tem capital suficiente para uma infraestrutura.

Não recomenda-se a utilização de IaaS quando as aplicações da empresa necessitam de hardware específico, ou os níveis de desempenho necessários para as aplicações tenham limite pelo provedor. Também não é aconselhado para empresas em que suas legislações não permitam o armazenamento de dados fora da empresa.

2.1.5.2 Plataforma como Serviço

PaaS – *Platform as a Service* – Seguindo os moldes do modelo de software como serviço, ele fornece todos os recursos necessários para construir aplicações e serviços, sem ter que se realizar a transferência ou instalação de nenhum software. Plataforma como serviço fornece aplicações de design, teste, desenvolvimento entre outras.

O usuário não administra ou controla a infraestrutura, ou seja, a camada de IaaS torna-se invisível para o usuário da PaaS, não se preocupando assim com tamanho hardware, memória, entre outros, porém ele tem controle sobre as aplicações implantadas. (MELL; GRANCE, 2009)

É aconselhável o uso quando se há necessidade de trabalhos em equipe, integração e triagem de serviços e integração de banco de dados. Aconselha-se também quando há a necessidade de um ambiente complexo de um ambiente para a aplicação.

A grande desvantagem desse modelo é a falta de portabilidade entre provedores quando a utilização de linguagem proprietárias.

2.1.5.3 Software como Serviço

SaaS – *Software as a Service* É o modelo de serviço no qual as aplicações são fornecidas como *host* para o cliente, o qual as acessa via internet. Sua vantagem na perspectiva do cliente é que quando o software é fornecido através de um *host*, o cliente não tem que manter nem dar suporte a esse software, essa obrigação agora é do prestador da infraestrutura da nuvem. O provedor faz todas as atualizações necessárias e mantém a infraestrutura do software em funcionamento.

O SaaS ajuda a reduzir os custos, pois dispensa-se a aquisição de licenças de sistemas, o usuário somente paga pelo que realmente usa. Um fator positivo para o SaaS é que por ser disponibilizado através da internet, os softwares por ele oferecidos podem ser acessados através de qualquer dispositivo com internet, podendo ser usado por *tablets*, *smartphones*, etc.

Sua utilização é recomendada quando há a necessidade de acesso das aplicações através de acesso remoto (dispositivos móveis) e também quando algum software de forma sazonal, como softwares de colaboração de projetos.

O SaaS não se recomenda quando as aplicações precisam de um processamento rápido, ou em tempo real, ou quando os requisitos de segurança são críticos.

2.1.6 Papéis na Computação em Nuvem

Os papéis são importantes para definir responsabilidades, acesso e perfil para os diferentes usuários que fazem parte e estão envolvidos em uma solução de computação em nuvem. (MARINOS; BRISCOE, 2009a).

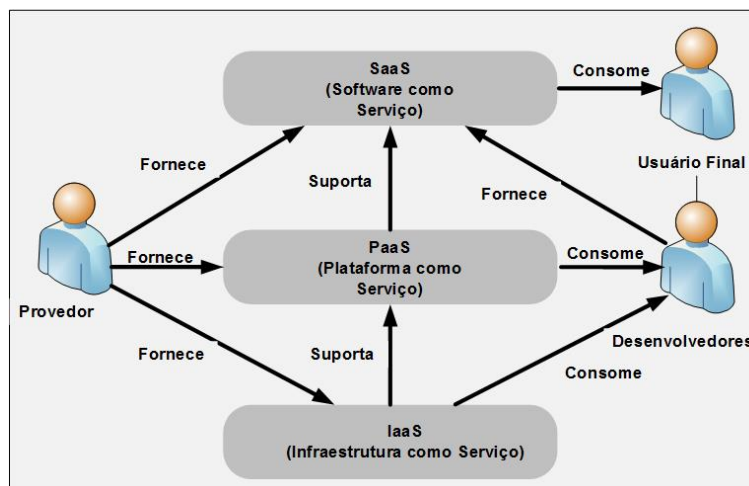


Figura 3: Papéis da computação em nuvem (MARINOS; BRISCOE, 2009b)

O provedor de infraestrutura é responsável por toda a gerência das soluções em computação em nuvem, é ele quem possui os *data centers* para alocação das suas máquinas virtuais.

Cada uma das instâncias criadas pelo provedor de infraestrutura é negociada com os provedores de serviço, que utilizam as suas máquinas virtuais para proporcionar um determinado tipo de serviço para os usuários do Serviço.

2.1.7 Modelos de implantação

Existem diferentes modos de se implementar um serviço de computação em nuvem, o modelo de implantação irá depender da necessidade da aplicação cliente. Segundo (MELL; GRANCE, 2009) “Os modelos de implantação da computação em nuvem podem ser divididos em nuvens públicas, privadas, comunidades e híbridas.”

2.1.7.1 Modelo privado

O modelo de implantação privado é aquele em que é construído, operado e mantido pela operadora da nuvem e compartilhado por toda a organização, este modelo é ideal para organizações que querem entrar no mundo da computação em nuvem mas não abrem mão do controle da mesma. Um dos principais motivos que tem levado as organizações a escolherem o modelo privado é a segurança que este modelo provê, pois segundo Cesar Taurion, diretor de novas tecnologias aplicadas da IBM Brasil o que a nuvem privada tem

de diferente é o fato de restringir acesso, pois se encontra atrás do *firewall* da organização. “É uma forma de aderir à tecnologia mantendo controle do nível de serviço e aderência às regras de segurança. Instituições financeiras como bancos, grandes empresas multinacionais como Microsoft e IBM, fazem o uso do modelo privado. Segundo (CHIRIGATI, 2009).” Uma nuvem privada é, em geral, construída sobre um data center privado.”

2.1.7.2 Modelo comunidade

Já o modelo comunidade é uma variação do modelo privado, usualmente usado por organizações que tem um objetivo de negócio em comum (por exemplo, para ser usada por um consórcio de compras) e compartilham do mesmo recurso (dados, software, hardware, etc.), este modelo normalmente acarreta em uma diminuição dos custos da implantação, pois os servidores não são exclusivos de uma organização.

Segundo (MELL; GRANCE, 2009) “Pode ser administrado por organizações ou por um terceiro e pode existir localmente ou remotamente”.

2.1.7.3 Modelo público

O modelo público é provido e designado para serviços de propósito geral, utiliza-se do modelo de comercialização *pay-per-use* ou seja, pague o quanto usar.

Pelo motivo dos clientes pagarem somente pelo o que usarem isso normalmente diminui os custos drasticamente, pois o cliente não precisa gastar com um *data-center* e mesmo se tivesse um o cliente não gasta com o tempo em que o servidor está ocioso.

Este modelo normalmente é usado por pequenas empresas, *startups* ou setores de TI dentro de empresas maiores de outro ramo, justamente pelo baixo custo e pela fácil aquisição de seus serviços, visibilidade limitada.

2.1.7.4 Modelo Híbrido

O modelo híbrido por sua vez, como o próprio nome diz, é uma composição entre os outros dois modelos. É onde os modelos de implantação público e privado acontecem ao mesmo tempo, o que permite a portabilidade de aplicações e dados. É um modelo que amplia os recursos de um modelo privado, podendo utilizar os recursos de um modelo público também.

Segundo (CHIRIGATI, 2009) “O termo “computação em ondas” é, em geral, utilizado quando se refere às nuvens híbridas”.

2.1.8 Distribuição de arquivos na nuvem

Uma das características mais interessantes da computação em nuvem vem do fato dela oferecer grande flexibilidade para os usuários, visto que toda a complexidade do gerenciamento é mascarado e invisível para quem utiliza, por exemplo, os bancos de dados podem ser transferidos para datacenters de grandes empresas especializadas, embora possa não ser muito confiável (CANEDO, 2013). Os usuários acabam ficando dependentes da disponibilidade e integridade fornecida pelos provedores de serviço de armazenamento. Assim, é necessária a utilização de modelos de armazenamento de dados seguros visando garantir a integridade dos dados dos usuários da nuvem (WANG et al., 2010).

O Sistema de arquivos distribuídos para nuvem é um sistema de arquivos que permite que muitos clientes tenham acesso aos mesmos dados/arquivos fornecendo importantes operações (criar, apagar, modificar, ler, escrever). Cada arquivo pode ser dividida em vários pedaços. Cada pedaço é armazenado em tipicamente em máquinas remotas, os dados são armazenados em arquivos em uma árvore hierárquica onde os nós representam os diretórios. Assim, facilita a execução em paralelo de aplicações.

Nas seções seguintes é apresentado uma visão geral dos sistemas de arquivos, Google File System, Amazon S3, Microsoft Azure e Hadoop.

2.1.8.1 Google File System (GFS)

O Google File System GFS é um sistema de arquivos escalável para aplicações de distribuição intensiva de dados (GHEMAWAT; GOBIOFF; LEUNG, 2003). Ele tem as mesmas características de outros sistemas de arquivos distribuídos como performance, escalabilidade, resiliência e disponibilidade. A Google utiliza o GFS para organizar e manipular grandes arquivos e permitir que aplicações consigam usar os recursos necessários. O desenvolvimento de um novo sistema de arquivos distribuído surgiu em 2003 a partir da ideia de armazenar dados de forma escalável, confiável, com alto desempenho e disponibilidade mesmo em máquinas não confiáveis, devendo atender às necessidades de uso do Google de gerar e manter grandes quantidades de dados. Em comparação com os sistemas de arquivos tradicionais, o GFS foi projetado e otimizado para ser executado em centros de dados e fornecer elevada vazão, baixa latência e tolerância a falhas individual de servidores (GHEMAWAT; GOBIOFF; LEUNG, 2003).

2.1.8.2 Amazon S3

O *Amazon Simple Storage Service* (S3) (S3, 2014) é um sistema de armazenamento distribuído desenvolvido com base no *Dynamo* (DECANDIA et al., 2007). O Amazon S3 fornece armazenamento através de serviços web interfaces (REST , SOAP , e BitTorrent). (S3, 2014) Amazon lançou S3, seu primeiro serviço web disponível ao público, nos Estados Unidos março 2006 e na Europa em novembro de 2007.

O *Dynamo* é o nome dado a um conjunto de técnicas que, quando tomados em conjunto podem formar uma alta disponibilidade de valor-chave do armazenamento estruturado de um sistema (DECANDIA et al., 2007) ou de um armazenamento de dados distribuídos. Possui propriedades de ambos os bancos de dados e tabelas de hash distribuídas (DHT). Amazon DynamoDB deriva seu nome do *Dynamo*, mas não é nada relacionado de qualquer outra forma. *Dynamo* tem um design multi-mestre exigindo que o cliente para resolver conflitos de versão enquanto DynamoDB tem um design único mestre. A API do *Dynamo* possui duas operações *get()* e *put()*. A operação *get(chave)* localiza as réplicas do objeto associado com a chave no sistema de armazenamento e retorna um objeto único ou uma lista de objetos com versões conflitantes, ao longo de um contexto. A operação *put (chave, contexto, objeto)* determina onde as réplicas do objeto devem ser colocadas com base na chave associada, e escreve a réplica no disco. As informações de contexto são armazenadas junto com o objeto para que o sistema possa verificar a validade do contexto do objeto fornecido na requisição (CANEDO, 2013).

2.1.8.3 Microsoft Azure

Microsoft SQL Azure é composto por um conjunto de serviços para o armazenamento e processamento de dados em nuvem (MICROSOFT, 2014). O SQL Azure juntamente com o Windows Azure Storage compõem a solução de gerenciamento de dados em nuvem da Microsoft (CANEDO, 2013). O objetivo do Windows Azure Storage é oferecer um armazenamento escalável, durável, altamente disponível e proporcionar ao usuário o pagamento sob demanda. Permite fácil acesso aos dados, oferecendo uma interface simples, disponíveis remotamente e em datacenters. Os serviços de armazenamento do Windows Azure Storage são oferecidos em quatro níveis de abstração:

- BLOBs – Fornece uma interface simples para armazenamento de grandes itens de dados. Um BLOB é um par (nome, objeto) que permite armazenar objetos com tamanho de até 50 GB;
- Tabelas – Fornecem um conjunto de entidades, que contêm um conjunto de propriedades. Um aplicativo pode manipular as entidades e consultar qualquer uma das propriedades armazenadas em uma tabela. São diferentes das tabelas relacionais e são compostas de entidades. Elas não são acessadas usando a linguagem SQL, mas por meio de serviços de dados;
- Filas – Fornece armazenamento confiável para a entrega de mensagens, propiciando expedição assíncrona de trabalhos para habilitar a comunicação entre os serviços de diferentes partes (papéis) de sua aplicação. Sua função principal é fornecer um serviço de troca de mensagens persistentes e confiável;

- Drives – Fornece volumes NTFS (Sistema de Arquivos de Nova Tecnologia) duráveis para aplicações. O Windows Azure Storage inclui armazenamento persistente por meio dos BLOBs, tabelas e filas. O acesso ao armazenamento e o balanceamento de carga é realizado automaticamente através de um conjunto de nós responsáveis pelo armazenamento físico, proporcionando escalabilidade e disponibilidade.

2.2 Redes de Computadores e Internet

Este capítulo visa introduzir alguns conceitos de Redes de Computadores e Internet

2.2.1 A camada de Transporte

A função básica da camada de transporte é aceitar dados da camada acima dela, dividi-los em unidades menores caso necessário, repassar essas unidades à camada de rede e assegurar que todos os fragmentos chegarão corretamente à outra extremidade. Além do mais, tudo isso deve ser feito com eficiência e de forma que as camadas superiores fiquem isoladas das inevitáveis mudanças na tecnologia de hardware. A camada de transporte também determina que tipo de serviço deve ser fornecido à camada de sessão e, em última análise, aos usuários da rede. O tipo de conexão de transporte mais popular é um canal ponto a ponto livre de erros que entrega mensagens ou *bytes* na ordem em que eles foram enviados. No entanto, outros tipos possíveis de serviço de transporte são as mensagens isoladas sem nenhuma garantia relativa à ordem de entrega e à difusão de mensagens para muitos destinos. O tipo de serviço é determinado quando a conexão é estabelecida.

2.2.1.1 TCP/IP

O TCP/IP (também chamado de pilha de protocolos TCP/IP) é um conjunto de protocolos de comunicação entre computadores em rede. Seu nome vem de dois protocolos: o TCP (*Transmission Control Protocol* - Protocolo de Controle de Transmissão) e o IP (*Internet Protocol* - Protocolo de Internet, ou ainda, protocolo de interconexão). O conjunto de protocolos pode ser visto como um modelo de camadas (Modelo OSI), onde cada camada é responsável por um grupo de tarefas, fornecendo um conjunto de serviços bem definidos para o protocolo da camada superior. As camadas mais altas estão logicamente mais perto do usuário (chamada camada de aplicação) e lidam com dados mais abstratos, confiando em protocolos de camadas mais baixas para tarefas de menor nível de abstração.

2.2.1.2 LDAP

Lightweight Directory Access Protocol, ou LDAP, é um protocolo para atualizar e pesquisar diretórios rodando sobre TCP/IP. Um diretório LDAP geralmente segue o

modelo X.500, que é uma árvore de nós, cada um consistindo de um conjunto de atributos com seus respectivos valores. O LDAP foi criado como uma alternativa ao muito mais incômodo *Directory Access Protocol* (DAP).

Um diretório LDAP tende a refletir vários limites políticos, geográficos e/ou organizacionais, dependendo do modelo adotado. A utilização do LDAP hoje em dia tende a se basear nos nomes já existentes do sistema *Domain Name System* (DNS), na estruturação dos níveis mais básicos de hierarquia. Mais profundamente, podem aparecer estruturas representando pessoas, unidades organizacionais, impressoras, documentos, grupos de pessoas ou qualquer outra coisa que represente um nó.

2.2.2 A camada de Aplicação

A camada de aplicação é a camada que a maioria dos programas de rede usa de forma a se comunicar através de uma rede com outros programas. Processos que rodam nessa camada são específicos da aplicação; o dado é passado do programa de rede, no formato usado internamente por essa aplicação, e é codificado dentro do padrão de um protocolo.

Alguns programas específicos são levados em conta nessa camada. Eles proveem serviços que suportam diretamente aplicações do usuário. Esses programas e seus correspondentes protocolos incluem o HTTP (navegação na *World Wide Web*), FTP (transporte de arquivos), SMTP (envio de email), SSH (login remoto seguro), DNS (pesquisas nome <-> IP) e muitos outros.

Existem diversos protocolos nesta camada. Como exemplo de alguns deles podemos citar:

- SMTP (Simple Mail Transport Protocol) é utilizado para a comunicação entre serviços de correio eletrônico na Internet.
- POP (Post Office Protocol) é utilizado para recuperação de mensagens de correio eletrônico via Internet.
- IMAP (Internet Mail Access Protocol) - também é utilizado para recuperação de mensagens de correio eletrônico via Internet, mas de forma mais avançada que o POP3.
- HTTP (Hypertext Transport Protocol) – utilizado para a publicação de sites WEB na Internet.
- FTP (File Transfer Protocol) – utilizado para publicação de arquivos na Internet.
- DNS (Domain Name System) - Utilizado para a distribuição de nomes de domínio

2.2.2.1 DNS

O DNS (Domain Name System - Sistema de Nomes de Domínios) é um sistema de gerenciamento de nomes hierárquico e distribuído visando resolver nomes de domínios em endereços de rede (IP).

O sistema de distribuição de nomes de domínio foi introduzido em 1984, e com ele, os nomes de *hosts* residentes em um banco de dados podem ser distribuídos entre servidores múltiplos, diminuindo assim a carga em qualquer servidor que provê administração no sistema de nomeação de domínios. Ele baseia-se em nomes hierárquicos e permite a inscrição de vários dados digitados além do nome do *host* e seu IP. Em virtude do banco de dados de DNS ser distribuído, seu tamanho é ilimitado e o desempenho não degrada tanto quando se adiciona mais servidores nele. Este tipo de servidor usa como porta padrão a 53. A implementação do DNS-Berkeley, foi desenvolvido originalmente para o sistema operacional BSD UNIX 4.3.

2.2.2.2 HTTP

O Hypertext Transfer Protocol (HTTP), em português Protocolo de Transferência de Hipertexto, é um protocolo de comunicação (na camada de aplicação segundo o Modelo OSI) utilizado para sistemas de informação de hipermídia, distribuídos e colaborativos.¹ Ele é a base para a comunicação de dados da World Wide Web.

Hipertexto é o texto estruturado que utiliza ligações lógicas (hiperlinks) entre nós contendo texto. O HTTP é o protocolo para a troca ou transferência de hipertexto.

2.2.3 Segurança de Redes

2.2.3.1 VPN

Rede Privada Virtual é uma rede de comunicações privada normalmente utilizada por uma empresa ou um conjunto de empresas e/ou instituições, construída em cima de uma rede de comunicações pública (como por exemplo, a Internet). O tráfego de dados é levado pela rede pública utilizando protocolos padrão, não necessariamente seguros.

Uma VPN é uma conexão estabelecida sobre uma infraestrutura pública ou compartilhada, usando tecnologias de tunelamento e criptografia para manter seguros os dados trafegados. VPNs seguras usam protocolos de criptografia por tunelamento que fornecem a confidencialidade, autenticação e integridade necessárias para garantir a privacidade das comunicações requeridas. Quando adequadamente implementados, estes protocolos podem assegurar comunicações seguras através de redes inseguras.

Deve ser notado que a escolha, implementação e uso destes protocolos não é algo trivial, e várias soluções de VPN inseguras são distribuídas no mercado. Advertem-se os

usuários para que investiguem com cuidado os produtos que fornecem VPNs.

2.2.3.2 Criptografia

A Criptografia tem suas origens a muitos anos. Comenta-se que o imperador Romano Júlio César teria sido o primeiro a empregá-la quando enviava cartas criptografadas, pois não confiava no mensageiro e havia o risco dele ser capturado, no caso de uma guerra. O método utilizado por César era simples: ele rescrevia a carta somando 3 a posição da letra, ou seja, o "A"(1) passaria a ser "D"(4), o "B"(2) "E"(5) e assim sucessivamente, imaginado as letras dispostas em círculo, ou seja, a lista não termina no "Z" mas continua daí no "A" novamente.

Hoje a Criptografia utiliza técnicas muito mais complexas que as de César Augusto mais a sua ideia ainda é empregada. Uma ligeira modificação de sua ideia original consiste, em vez de trocar cada caractere por ele + 3, troca-lo por ele + "n". Para descriptografar, portanto, o receptor deverá saber o valor de "n". Introduzimos ai o conceito de "chave". Já não basta o receptor conhecer apenas o método empregado, mas também deve conhecer a chave. Esta técnica, no entanto, é fácil de ser quebrada (manualmente, pois com o auxílio de computadores é muito fácil testar se um "A" foi substituído por algum dos outros 25 caracteres existentes) tendo-se em vista uma tabela de frequência para língua utilizada. Por exemplo, no inglês estudos comprovam que as letras mais usadas são e, t, o, a, n, i, etc. Precisaríamos apenas de uma tabela com suas frequências.

2.2.3.3 Firewall

Um *firewall* (em português: Parede de fogo) é um dispositivo de uma rede de computadores que tem por objetivo aplicar uma política de segurança a um determinado ponto da rede. O *firewall* pode ser do tipo filtros de pacotes, proxy de aplicações, etc. Os firewalls são geralmente associados a redes TCP/IP.1 ..

Este dispositivo de segurança existe na forma de software e de hardware, a combinação de ambos é chamado tecnicamente de "appliance". A complexidade de instalação depende do tamanho da rede, da política de segurança, da quantidade de regras que controlam o fluxo de entrada e saída de informações e do grau de segurança desejado.

2.3 Discussões sobre segurança da informação

A computação em nuvem é um modelo que utiliza a Internet para disponibilizar seus serviços. Isso se torna mais complexo visto que os recursos computacionais utilizam diferentes domínios de redes, sistemas operacionais, software, criptografia, políticas de segurança, entre outros (CANEDO, 2013). Questões de segurança devem ser consideradas para prover a autenticidade, confidencialidade e integridade. No que diz respeito à confi-

abilidade e responsabilidade, o provedor deve fornecer recursos confiáveis, especialmente se a computação a ser realizada é crítica e deve existir uma delimitação de responsabilidade entre o provedor e o usuário. Dessa forma, devem-se ter meios para impedir o acesso não autorizado a informações e que os dados sensíveis permaneçam privados, pois estes podem ser processados fora das empresas (AGRAWAL; DAS; ABBADI, 2010). Em geral, cada sistema tem seu próprio modelo de dados e política de privacidade destes dados (COOPER et al., 2010). Quando ocorre a movimentação de dados entre sistemas, deve-se garantir a privacidade dos dados mesmo com a mudança entre modelo de dados diferente e que aplicações multi-inquilino acessem.

2.3.0.4 Os cinco riscos de segurança aos quais devemos ficar atentos

Quando você usar a computação em nuvem, é importante saber onde os dados estão, como estão protegidos e quem pode acessá-los. Infelizmente, muitos provedores de serviços de computação em nuvem não compartilham esses detalhes. E o que é pior: muitos não fazem nenhum tipo de promessa no que diz respeito à proteção de seus dados. Se uma abordagem de vários fornecedores de segurança corporativa de TI dificultar a existência de um único ponto de responsabilidade, a computação em nuvem será ainda mais arriscada. Se escolher o provedor de serviços errado, aí que você não terá de quem cobrar a responsabilidade. Antes de poder avaliar que tipo de segurança um provedor de serviços de computação em nuvem oferece, compreenda os maiores riscos.

1. Proteção de dados e gerenciamento de privacidade - Muitos provedores de serviços de computação em nuvem não oferecem acordos de nível de serviço (SLAs). Isso significa que você fica sem garantia quanto à disponibilidade dos dados, privacidade ou proteção das informações.
2. Governança, risco e conformidade - Confiar seus dados a um provedor de serviços de computação em nuvem não significa que você está isento da responsabilidade de assegurar a proteção desses dados. A computação em nuvem aumenta riscos que alguns provedores de serviços podem não cuidar. Por exemplo, as políticas de retenção e registro de um provedor de serviços de computação em nuvem podem não atender às suas obrigações regulamentares. Se o provedor de serviços de computação em nuvem não estiver fazendo o registro completo ou exato dos dados, você poderá ter problemas em uma auditoria de segurança.
3. Gerenciamento de identidades - Quando seus dados estiverem dentro do *firewall* do provedor de serviços, quem terá acesso a eles e em quais circunstâncias? Com que rapidez seu provedor de serviços pode conceder acesso? E, mais importante, com que rapidez ele cancela acesso administrativo e de usuário? Suas próprias políticas

de autorização de dados podem ser excepcionalmente rígidas. Mas as políticas do seu provedor de serviços podem ficar fora do seu controle.

4. Segurança da infraestrutura - Os aplicativos e os dados confiados a um provedor de computação em nuvem ficam em servidores e armazenamento que você não escolheu ou que não mantém pessoalmente. A maioria dos fornecedores não dá visibilidade além de seus recursos virtuais. Então, como saber o nível de segurança que os equipamentos físicos realmente têm? Como saber se seus aplicativos estão sendo executados em um sistema operacional com *patches* perfeitos e não em um repleto de buracos?
5. Preparação - Inserir arbitrariamente uma aplicação na computação em nuvem não é uma forma inteligente de avaliar a prontidão dela. No entanto, poucos provedores de serviços oferecem o tipo de avaliação necessária para definir se a aplicação faz sentido para a computação em nuvem.

2.3.0.5 Como reduzir os riscos

A computação em nuvem não precisa ser repleta de riscos. Com o provedor de serviços certo, a computação em nuvem pode cumprir a promessa de serviços de TI mais flexíveis e mais fáceis de gerenciar, com preços mais acessíveis. No entanto, muito depende de seu preparo e de sua escolha dos provedores de serviços.

1. Classificação - Quando estiver pensando em serviços de computação em nuvem, primeiro classifique seus dados para determinar a adequação deles para a computação em nuvem. Uma parte importante desse processo é fazer uma análise do custo-benefício. As economias geradas quando se colocam os dados em nuvem compensam os riscos de brecha de segurança ou regulamentações de privacidade?
2. Avaliação - Encontre um provedor de serviços que faça avaliações de segurança para definir se os aplicativos ou os dados estão prontos para a computação em nuvem. Os melhores provedores de serviços irão determinar as regulamentações de conformidade às quais você está sujeito e irão ajudá-lo a cumpri-las.
3. Comece pelas informações não confidenciais - Não comece sua aventura pela computação em nuvem com aplicativos que exponham os números de cartão de crédito ou informações sobre as contas bancárias de seus clientes. Comece por aplicações que ofereçam menos risco até você conseguir gerenciar com segurança o modelo e os serviços do seu provedor.
4. Avaliação crítica dos contratos do provedor de serviços - Descubra exatamente como o seu provedor de serviços pretende proteger seus dados e mantê-los privados na nuvem. Se os seus dados forem essenciais para os negócios, exija garantias satisfatórias

do provedor. Isso inclui termos de serviço (TOS) apropriados, políticas aceitáveis de uso (AUP) e contratos de nível de serviços (SLA).

5. Criptografia - Não deixe a criptografia para o seu provedor de serviços de computação em nuvem. Certifique-se de que você tenha um gerenciamento de ciclo de vida de chaves. Além disso, usando a sua classificação de dados como orientação, faça a criptografia dos dados conforme apropriado e necessário.
6. Insista na transparência - Exija a capacidade de saber o que está acontecendo na infraestrutura física subjacente à infraestrutura virtual.

2.4 A Fábrica de Software

O projeto de Pesquisa e Desenvolvimento, Fábrica de Software do CEBRASPE (aprovado no conselho da FGA e na CPP do DPP/UnB) é parceria entre a UnB e o CEBRASPE (Antigo CESPE, desde Out/2013). Seu principal objetivo é construir um ambiente de pesquisa e desenvolvimento de ferramentas e metodologias na área de Engenharia de Software. Um dos focos da Fábrica de Software, além de total apoio ao desenvolvimento de soluções para o CEBRASPE, é o de trazer alunos do curso de Engenharia de Software em um estágio ainda cedo no curso e já inseri-los em um ambiente de desenvolvimento, fomentando assim a criatividade e o poder de criação em grupo, para que futuramente os alunos que por aqui passarem se sintam inspirados a criar suas próprias empresas *start-ups*

Este ambiente é composto por:

- Pesquisadores da área de engenharia de software;
- Alunos de graduação dos cursos de computação;
- Colaboradores do CEBRASPE;
- Alunos do ensino fundamental e médio os quais queiram estar se capacitando na área de computação;
- Comunidade aberta dos que queiram produzir/capacitar-se na produção de software de qualidade.

Equipe da Fábrica de Software

- Beatriz Rezener Dourado Matos (FGA)
- Cristovão de Lima Frinhani (FGA)
- Caique Pereira (FGA)

- Guilherme Fay Vergara (FGA)
- João Gabriel de Britto e Silva (FGA)
- Sérgio Antônio Andrade de Freitas (Coordenador na FGA)
- Jorge H. C. Fernandes (Coordenador no Darcy Ribeiro)

3 Ferramentas para Computação em Nuvem

Este capítulo apresenta algumas soluções existentes para computação em nuvem que podem ser aplicadas à Fábrica de Software da Faculdade do Gama, salientando que benefícios esta solução pode trazer. Este capítulo está dividido por tipo de arquitetura, IaaS, PaaS e SaaS para uma melhor comparação.

3.1 IAAS

Soluções de IAAS visam principalmente fornecer máquinas virtuais com todas as características da nuvem para a fábrica de software. Esta solução beneficia a fábrica pois não precisa de máquinas potentes em todas as estações de trabalho, e possibilita aos usuários da fábrica utilizarem suas estações em qualquer lugar que desejarem. Outro benefício que esta solução traz é de comercialização destas máquinas, podendo futuramente vender para futuros cliente.

3.1.1 Microsoft Windows AZURE

O Microsoft Azure ([MICROSOFT, 2014](#)) é uma plataforma para a implementação de computação em nuvem que oferece um conjunto específico de serviços para desenvolvedores [Azure]. Esta plataforma pode ser usada por aplicações em execução em nuvem ou fora desta. A plataforma Azure é formada pelo sistema operacional Windows Azure e um conjunto de serviços: *Live Services*, *.NET Services*, *SQL Services*, *SharePoint Services* e *Dynamics CRM Services*.

O Windows Azure é um sistema operacional para serviços na nuvem que é utilizado para o desenvolvimento, hospedagem e gerenciamento dos serviços dentro do ambiente Azure. Microsoft *.NET Services* é um conjunto de serviços escaláveis, orientados ao desenvolvedor e que oferecem os componentes necessários para a maioria das aplicações baseadas em nuvem. Estes serviços possibilitam o desenvolvimento focado na lógica da aplicação ao invés de ter a necessidade de construir e disponibilizar o próprio serviço de infraestrutura na nuvem. O *Live Services* é um conjunto de componentes dentro do Azure para o tratamento de dados do usuário e recursos da aplicação. *Live Services* possibilita aos desenvolvedores construir aplicações ricas que podem conectar com usuários do Windows Live.

O *Live Services* inclui as tecnologias do *Live Mesh* para sincronização de dados dos usuários e possibilita a extensão de aplicações Web entre múltiplos dispositivos. O *SQL Services* é um serviço de armazenamento de dados e de processamento de consultas

escalável, sendo construído com base na tecnologia do SQL Server. O componente SharePoint Services permite colaborar e criar aplicações Intranet e o *Dynamics CRM Services* é um sistema totalmente integrado de CRM.

3.1.2 OpenStack

De forma simples o OpenStack (OPENSTACK. . . , 2014) é um software de código aberto, capaz de gerenciar os componentes de múltiplas infraestruturas virtualizadas, assim como o sistema operacional gerencia os componentes de nossos computadores, o OpenStack é chamado de Sistema Operacional da Nuvem, por cumprir o mesmo papel em maior escala.

O OpenStack é um conjunto de projetos de software *open source* que empresas e provedores de serviços podem usar para configurar e operar sua infraestrutura de computação e armazenamento em nuvem. A Rackspace (provedor de infraestrutura americano) e a NASA (agência espacial americana) foram os principais contribuidores iniciais para o projeto. A Rackspace forneceu sua plataforma *Cloud Files* para implementar o aspecto de armazenamento (*Object Storage*) do OpenStack, enquanto que a NASA entrou com o "Nebula" para implementar o lado computacional (Compute).

3.1.3 Xen

Xen (XEN. . . , 2014) é um projeto desenvolvido em comunidade com código livre, sendo um hypervisor do tipo 1 o que torna possível executar várias instâncias de um sistema operacional ou até mesmo diferentes sistemas operacionais em paralelo em uma única máquina (ou *host*). O hypervisor é usado como base para uma série de diferentes aplicações comerciais e *open source*, tais como: virtualização de servidores, infraestrutura como Serviço.

Xen foi originalmente desenvolvido como um projeto de pesquisa na Universidade de Cambridge, liderado por Ian Pratt, fundador da XenSource, Inc. Em 15 de agosto de 2007, a XenSource foi adquirida pela *Citrix System Inc.* pelo valor de 500 milhões de dólares. A Citrix agora suporta o desenvolvimento do projeto *open source* e vende também uma versão "empresarial" do software. O primeiro lançamento público do Xen foi disponibilizado em 2003.

3.1.4 Eucalyptus

O projeto Eucalyptus (LIU; LIANG; BROOKS, 2007) é uma infraestrutura de código aberto que fornece uma interface compatível com o Amazon EC2, S3, *Elastic Block Store* (EBS) e permite aos usuários criarem uma infraestrutura e experimentar

a computação em nuvem. A arquitetura do Eucalyptus é simples, flexível e modular e contém uma concepção hierárquica que reflete os recursos comuns do ambiente.

O Eucalyptus tem como objetivo auxiliar a pesquisa e o desenvolvimento de tecnologias para computação em nuvem e possui as seguintes características: interface compatível com o EC2, instalação e implantação simples usando ferramentas de gerenciamento de *clusters*, apresenta um conjunto de políticas de alocação extensível de nuvem, sobreposição de funcionalidade que não requer nenhuma modificação em ambiente Linux, ferramentas para administrar e auxiliar a gestão do sistema e dos usuários e capacidade de configurar vários *clusters*, cada um com endereços privados de rede interna em uma única nuvem. Esse sistema permite aos usuários iniciar, controlar o acesso e gerenciar todas as máquinas virtuais utilizando uma emulação do protocolo SOAP do Amazon EC2 e interfaces de consulta. Neste sentido, os usuários interagem com o Eucalyptus utilizando as ferramentas e interfaces exatamente do mesmo modo que eles interagiriam com o Amazon EC2.

3.2 PAAS

As soluções em PaaS oferecem uma plataforma para que os usuários possam de forma simples e rápida colocar seus programas em produção, proporcionando assim um ambiente para testes de forma rápida.

3.2.1 Tsuru

Tsuru ([GLOBO.COM, 2014](#)) é uma plataforma de computação em nuvem poliglota e *open source* desenvolvida pela Globo.com desde 2012, e começou a ser oferecida em uma versão preliminar em 2013. Como outras plataformas, o Tsuru facilita o desenvolvimento de aplicações web sem o custo de um ambiente de servidores.

Tsuru usa Juju para orquestração de serviços e tira proveito dos atrativos da sua arquitetura. Linguagens de programação de desenvolvimento suportados incluem Go, Java, Python e Ruby

3.2.2 Heroku

Heroku ([HEROKU, 2014](#)) é uma plataforma de *cloud* como serviço com suporte para várias linguagens de programação. Heroku foi adquirida pela Salesforce.com em 2010. O Heroku é uma das primeiras plataformas de nuvem, já está em desenvolvimento desde junho de 2007, quando apoiou apenas a linguagem de programação Ruby, mas, desde então, adicionado suporte para Java, Node.js, Scala, Clojure e Python e Perl. O sistema operacional base é Debian ou, no mais recente o Ubuntu baseado em Debian.

3.3 SAAS

Soluções baseadas em Software como serviço talvez sejam as mais interessantes para uma fábrica de software, visto no processo de desenvolvimento de um software é utilizado várias ferramentas, que principalmente ajudem na colaboração entre os membros da fábrica, como serviços de troca de arquivos, e-mail, entre outros. Um papel importante que este tipo de arquitetura pode trazer é a de que a fábrica agrega mais segurança a todo o seu processo, visto que não depende de servidores alocados em algum lugar no mundo, e sim dentro de sua própria infraestrutura.

3.3.1 Owncloud

O ownCloud ([OWNCLOUD, 2014](#)) é um aplicativo web gratuito e de código aberto para a sincronização de dados, compartilhamento de arquivos e armazenamento remoto de documentos está escrito em PHP e linguagens de script Java Script. Para a gestão da mudança, que emprega SabreDAV, um servidor WebDAV *open source* projetou-se para funcionar com vários sistemas de gerenciamento de banco de dados, incluindo o SQLite, MariaDB, MySQL, banco de dados Oracle e PostgreSQL.

3.3.2 Expresso

O Expresso Livre ([EXPRESSO... , 2014](#)) é uma solução para comunicação corporativa, totalmente baseada em software livre, que reúne Email, Agenda, Catálogo de Endereços, Workflow e Mensagens Instantâneas em um único produto.

Características:

- Alta Escalabilidade
- Baixo custo
- Utilização de protocolos padrões
- Integração com outros sistemas Web
- Independência de fornecedor
- Independência de plataforma cliente
- Independência de plataforma servidor
- Mobilidade dos usuários

4 O Modelo de Implantação para a Fábrica de Software

A partir do ferramental apresentado no capítulo 3, apresenta-se neste capítulo um modelo de implantação para a fábrica de software (VERGARA; CANEDO; FREITAS, 2014). Para este trabalho foram escolhidos dois softwares que pudessem contribuir no trabalho colaborativo da fábrica. Tais softwares devem contribuir para que os membros da fábrica possam compartilhar documentos de projetos, tarefas, agendas compartilhadas e demais ferramentas para gerenciamento de projetos. Um fator importante a ser considerado na hora de adoção de serviços na nuvem é a segurança, fato tal que tem despertado diversas discussões no Governo Federal Brasileiro ao ponto de ter sido expedido um decreto presidencial (decreto N° 8.135, de 4 de novembro de 2013). O primeiro artigo deste decreto é descrito a seguir:

"Art. 1º As comunicações de dados da administração pública federal direta, autárquica e fundacional deverão ser realizadas por redes de telecomunicações e serviços de tecnologia da informação fornecidos por órgãos ou entidades da administração pública federal, incluindo empresas públicas e sociedades de economia mista da União e suas subsidiárias."

Este artigo mostra com clareza esta preocupação do Governo, que começou a ser bastante comentada depois de ser tornar pública os casos de espionagem aos e-mails da Presidente do Brasil.

4.1 Modelo IaaS

Um primeiro modelo proposto para a fábrica de software foi as soluções em IaaS (VERGARA; CANEDO; FREITAS, 2014), ou seja, fornecer para a fábrica de software toda a infraestrutura necessária de forma transparente e escalável. Porém a fábrica de software não provê hoje de servidores e *storage* necessários para se manter, tais recursos de *hardware* ainda estão passando por processo de licitação. E visando sanar este problema, a principal solução é que se utilize máquinas virtuais até que a fábrica tenha como se manter com seus próprios meios. Para o provimento de máquinas virtuais foi escolhido o XEN Hypervisor como solução, por ter de grande utilização do mercado, além de ser de código aberto e já ter sido estudado anteriormente.

A figura 4 que mostra a instalação de duas máquinas virtuais Linux no cliente Windows. Uma das máquinas é um servidor Linux, e a outra um Ubuntu com interface gráfica.

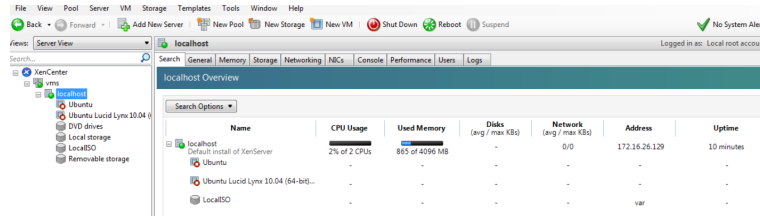


Figura 4: Xen client

O interessante que pode ser visto na figura é que podemos controlar isoladamente e totalmente a memória, o disco e a rede. Desta forma obter-se-á uma clara ideia se a VM foi bem provisionada.

4.2 Modelo PaaS

Por se tratar de uma fábrica de software, é interessante que a fábrica seja capaz de produzir protótipos o mais rápido possível, pois quanto antes o cliente tem um protótipo diminui-se os riscos de um software que não agregue funcionalidades úteis para o cliente e mais cedo o cliente tem uma visualização do software que será entregue, para isso muito importante que a fábrica tenha uma plataforma onde possa rapidamente colocar o software em produção e possa mostrar aos seus clientes. Para este tipo de serviço foram pesquisadas as ferramentas OpenShift e Tsuru, ambas não foram testadas localmente, porém pelas pesquisas o OpenShift mostrou-se uma poderosa ferramenta para tais funções, o OpenShift suporta as principais linguagens utilizadas atualmente e é uma das principais soluções *Open Source*.

4.3 Modelo SaaS

Por último porém não menos importante propõe-se um modelo de SaaS (VERGARA; CANEDO; FREITAS, 2014), e em um primeiro contato estas soluções pareciam não serem tão relevantes frente a IaaS e a PaaS, pois a internet está completamente cheia de softwares ofertados como serviço, tais como, Dropbox, Google Docs, porém um dos principais fatores que levam a que não se utilize tais softwares é a segurança que eles não provem. Por se tratar de uma fábrica de software, é de suma importância que seus projetos não sejam abertos de alguma forma para outros utilizadores.

Destaca-se que a principal empresa de TI do Governo Federal, o Serviço Federal de Processamento de Dados - SERPRO (SERPRO, 2014), está diretamente relacionado

ao decreto comentado na introdução deste capítulo, tendo sido esta Empresa destinada a fornecer tais serviços, como se vê a seguir:

"Os órgãos e entidades da União a que se refere o caput deverão adotar os serviços de correio eletrônico e suas funcionalidades complementares oferecidos por órgãos e entidades da administração pública federal."

E é neste âmbito que surge o Expresso, plataforma de e-mail e demais funcionalidades em que o SERPRO é um dos principais desenvolvedores, e possui a alocação dos seus recursos totalmente internos, aumentando assim a segurança da aplicação.

Como comentado anteriormente, a fábrica precisa de soluções que agreguem valor ao trabalho colaborativo. Como solução para este problema, apresenta-se duas soluções de SaaS, são elas as seguintes:

- Expresso

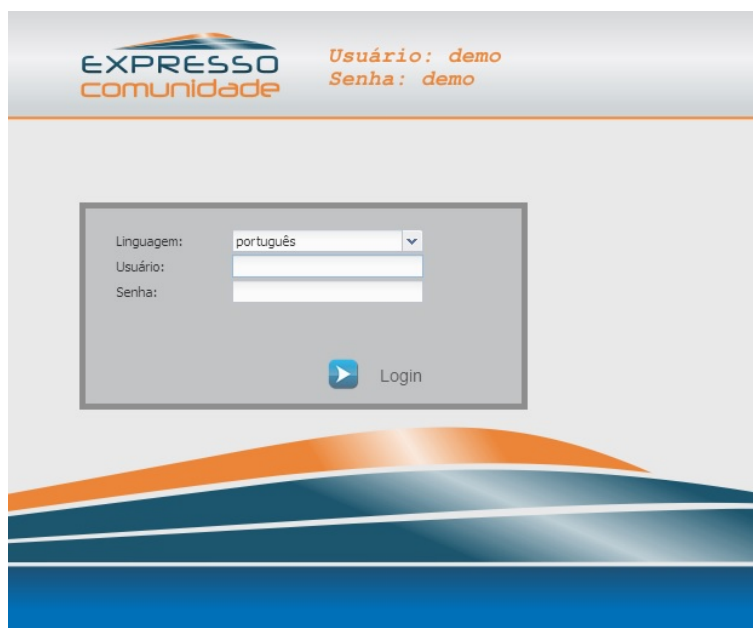


Figura 5: Log-in no Expresso

O expresso (EXPRESSO..., 2014) como comentado no capítulo anterior é uma solução completa de comunicação, que inclui e-mail, agenda, catálogo de endereços, workflow e mensagens instantâneas em um único ambiente. Esta solução facilita muito o trabalho dos membros da fábrica de software, pois eles podem marcar reuniões e marca-las em uma agenda compartilhada, facilitando a alocação de horários disponíveis para todos. Outro serviço muito interessante a ser utilizado pela fábrica é o de vídeo chamada, onde os membros do projeto podem ter reuniões sem sair

de casa, facilitando encontros e agilizando reuniões, que não mais precisam ser fisicamente ocorridas. Não podemos nos esquecer, é claro, do principal serviço do Expresso, que é de e-mail, que será muito mais seguro que em ambientes alocados na internet, como o GMAIL, HOTMAIL, etc. Na figura 6 temos a tela principal do Expresso

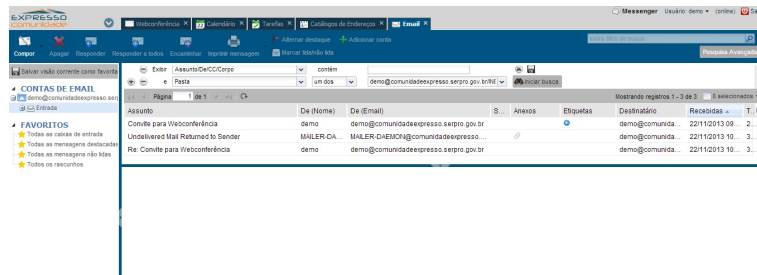


Figura 6: Tela principal do Expresso

Nota-se na parte de cima da figura pode as principais funcionalidades do Expresso. Primeiramente a aba selecionada evidencia a funcionalidade de e-mail, com várias opções de filtro, favoritos, etc. Logo a esquerda temos o catálogo de endereços, onde todos os contatos do usuário são salvos, seguindo com as abas temos uma funcionalidade muito interessante que é a de tarefas, onde por exemplo um gerente de alguma equipe pode delegar tarefas a algum de seus membros, de maneira simples e rápida. A funcionalidade de calendário depois da funcionalidade principal de e-mail, talvez seja a principal a ser usada pelas equipes, pois nela pode-se gerenciar de forma muito simples a agenda de cada um dos participantes, e facilitar a alocação de reuniões, deadlines, etc. E por último temos a funcionalidade de web conferência, onde os membros de uma reunião por exemplo, podem simplesmente através de uma solicitação via e-mail, se juntar para uma vídeo conferência, facilitando o acontecimento de reuniões de equipes distribuídas, ou com dificuldade de horário para encontros presenciais.

- Owncloud

Além da implantação do serviço de e-mail pretende-se disponibilizar aos membros da fábrica, um repositório de arquivos, onde os poder-se-á compartilhar documentos importantes dentro de um ambiente mais seguro que as soluções disponíveis, pois os arquivos estarão dentro da própria infraestrutura da fábrica.

Tal Software escolhido é o OwnCloud ([OWNCLOUD, 2014](#)), que é uma solução parecida com o conhecido Dropbox, porém OwnCloud é um software livre de código aberto, e conta com grande apoio da comunidade. O OwnCloud possui como funcionalidade principal a possibilidade do usuário guardar seus arquivos, de forma que ele possa acessar em qualquer computador com o cliente, tanto *desktops* quanto *smartfones*



Figura 7: Log-in no Owncloud

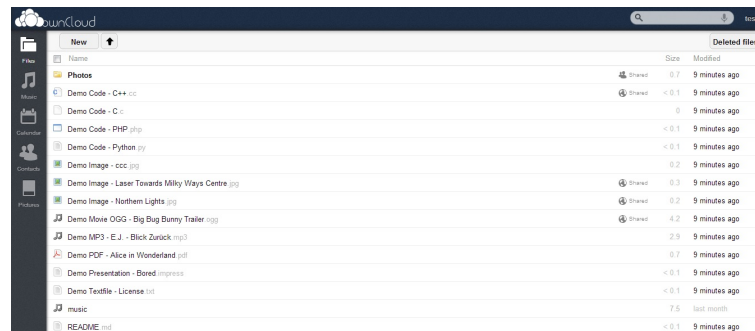


Figura 8: tela principal do Owncloud

Na figura 8 notamos as opções de alocação do OwnCloud, como pode ser visto, o primeiro arquivo é um exemplo de código escrito na linguagem C++, arquivos deste tipo podem ser visualizados na própria interface web, já sendo aceito pela aplicação e com visualização semelhante a IDE's para esta extensão, podendo ser editados diretamente do *browser*, facilitando assim a vida dos programadores da fábrica. O mesmo se aplica para todas as outras linguagens mostradas na figura e também para arquivos txt, impress, entre outros. Ao se clicar nas imagens um pop-up aparece com a imagem selecionada, facilitando a visualização da mesma. E por último temos as músicas, que podem ser tocadas diretamente da internet.

Todos estes arquivos podem ser baixados de duas formas: Primeira, quando passado o mouse em cima do arquivo clica-se em download, e o download é efetuado. Ou a segunda forma, que é baixando o cliente do OwnCloud, assim você mantém todos os seus arquivos atualizados.

4.4 Processo

A implantação dos serviços é um processo árduo, visto que muitos imprevistos normalmente acontecem, e que demanda muito tempo para ser executado. A figura 9 é uma proposta de modelo de um processo pra a implantação de serviços em uma nuvem.

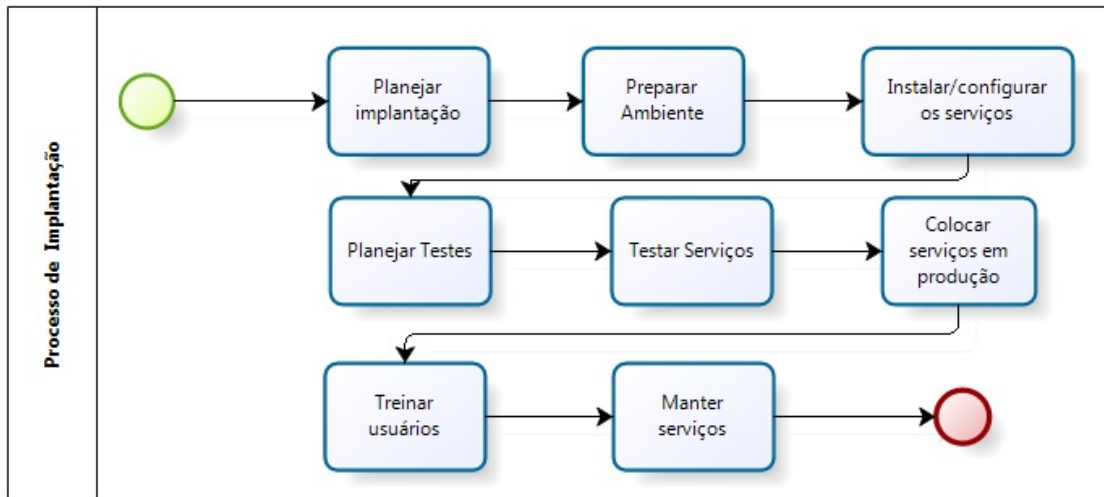


Figura 9: Processo de implantação (VERGARA; CANEDO; FREITAS, 2014)

1. **Planejar Implantação** Este é o início do processo para a implantação dos serviços. Esta fase destina-se a um estudo mais "baixo nível" de que materiais (servidores, *storage*) que serão necessários para a implantação. Neste caso necessita-se um estudo sobre os softwares que serão implantados, para ficar bem claro quais serão os elementos que precisam ser instalados e configurados.
2. **Preparar Ambiente** Nesta fase, é indispensável que já se tenha uma boa visão dos elementos essenciais para a instalação dos softwares. Visa-se, nesta fase, que se "junte" a maior parte dos materiais necessários para a instalação, como máquinas, *storages*, entre outros. Normalmente é nessa fase onde deve-se configurar os servidores, instalando o sistema operacional de sua escolha. Esta fase pode ser facilitada por um provedor de infraestrutura, que para uma maior segurança deve ser um interno a fábrica ou um de confiança da organização.
3. **Instalar serviços** Esta fase se destina a instalar/configurar os serviços escolhidos, sendo talvez, uma das fases mais demoradas, visto que ainda não há um conhecimento amplo sobre o processo da instalação dos serviços. Imprevistos sempre podem ocorrer, acabando por tornar esta fase bastante morosa. Para minimizar tal problema os mantenedores do serviço podem ter um bom manual de instalação disponível.

4. Planejar testes Depois dos serviços já instalados, planeja-se os testes da implantação do software, pode-se criar nesta fase um documento especificando como todos os testes serão executados, bem como os resultados esperados e comparando-os com os resultados encontrados na próxima fase.
5. Testar serviços Após os serviços estarem totalmente instalados, e os testes programados, deve-se executar cada um deles em busca de algum problema que possa ter ocorrido. Caso algo tenha sido encontrado deve-se corrigir antes de passar para a próxima fase.
6. Colocar serviços em produção Esta fase junto com a instalação, talvez seja a mais difícil e demorada. Pois devemos garantir que os serviços estão funcionando de maneira correta, e procurar que os clientes estejam utilizando-os. Deve-se procurar estar sempre junto com os clientes, buscando *feedbacks* sobre melhorias e possíveis problemas.
7. Treinar usuários Logo após e os serviços terem sido colocados em produção, os usuários precisam de um pequeno treinamento sobre como utilizar as ferramentas. Como normalmente os usuários já são habituados com a utilização de softwares, esta fase pode ser um pouco facilitada. Porém, se a fábrica deseja implantar estes softwares, nos seus clientes, esta fase se torna crucial para a adoção pelo lado do cliente. Porém como os serviços precisaram de manutenção futura, também inclui-se nesta fase a transferência de conhecimento para algum(s) membro(s) da fábrica para que ele possa continuar na operação dos serviços.
8. Manutenção dos serviços Como a maioria dos softwares evoluem e apresentam problemas sempre há a necessidade de continua manutenção e atualizações dos mesmos. Esta fase dura até o fim da vida útil dos serviços. A manutenção dos serviços é feita primeiramente pelo(s) membro(s) orientado(s) no treinamento anterior.

5 A Implantação

Este capítulo apresenta a implantação do modelo proposto no capítulo anterior, aqui porém focamos na implantação do SAAS, os outros modelos de implantação não foram implantados por questão de tempo. Os serviços implantados são o Expresso, o OwnCloud e o, Open meetings.

5.1 Implantação

Nesta seção destaca-se como foi feita a implantação em seus detalhes. Principalmente, quais são os papéis neste processo de implantação, e como ficaram as arquiteturas tanto física, quanto de serviços.

5.1.1 Papéis

A implantação dos softwares na fábrica de software do gama foi uma parceria entre três instituições, o SERPRO (SERPRO, 2014), a UnB e o CEBRASPE (CEBRASPE, 2014). O CEBRASPE assumiu um papel de provedor de infraestrutura, provendo as máquinas virtuais e toda a infraestrutura de hardware necessária para a implantação do SaaS, a fábrica se encaixa no papel de provedor de serviços tanto para a própria fábrica quanto para seus clientes. Esta interação pode ser melhor representada pela figura 10.



Figura 10: Papéis da computação em nuvem (VERGARA; CANEDO; FREITAS, 2014)

5.1.2 Arquitetura Física

A arquitetura física da implantação é em um total de 5 servidores, sendo 4 Linux e 1 Windows, todos cedidas pelo CEBRASPE. Todos esses servidores podem ser vistos na figura 11

5.1.2.1 Servidores

- Servidor Linux - VPN: Este servidor talvez seja um dos mais importantes de todos, este servidor serve de porta de entrada para todos os outros servidores. Este servidor

serve como uma terceira parte ao sistema como um todo, protegendo os outros servidores.

- Servidor Windows - AD: O servidor de AD serve para centralizar todas as informações de usuários dos serviços, como seus nomes, logins e senhas criptografadas. Esta aplicação é muito importante pois não precisamos a cada novo usuário criar um login em cada sistema implantado.
- Servidor Linux - Desenvolvimento: Este servidor serve para instalarmos os softwares para testes internos, para que quando estiverem prontos para serem usados possam passar para produção e colocados no próximo servidor.
- Servidor Linux - Produção: Após os softwares passarem pelo servidor de testes eles são colocados em produção neste servidor, para que possam ser usados por todos os interessados.
- Servidor Linux - NAS: Este servidor reúne todos os arquivos de banco de dados dos softwares que estão em produção.

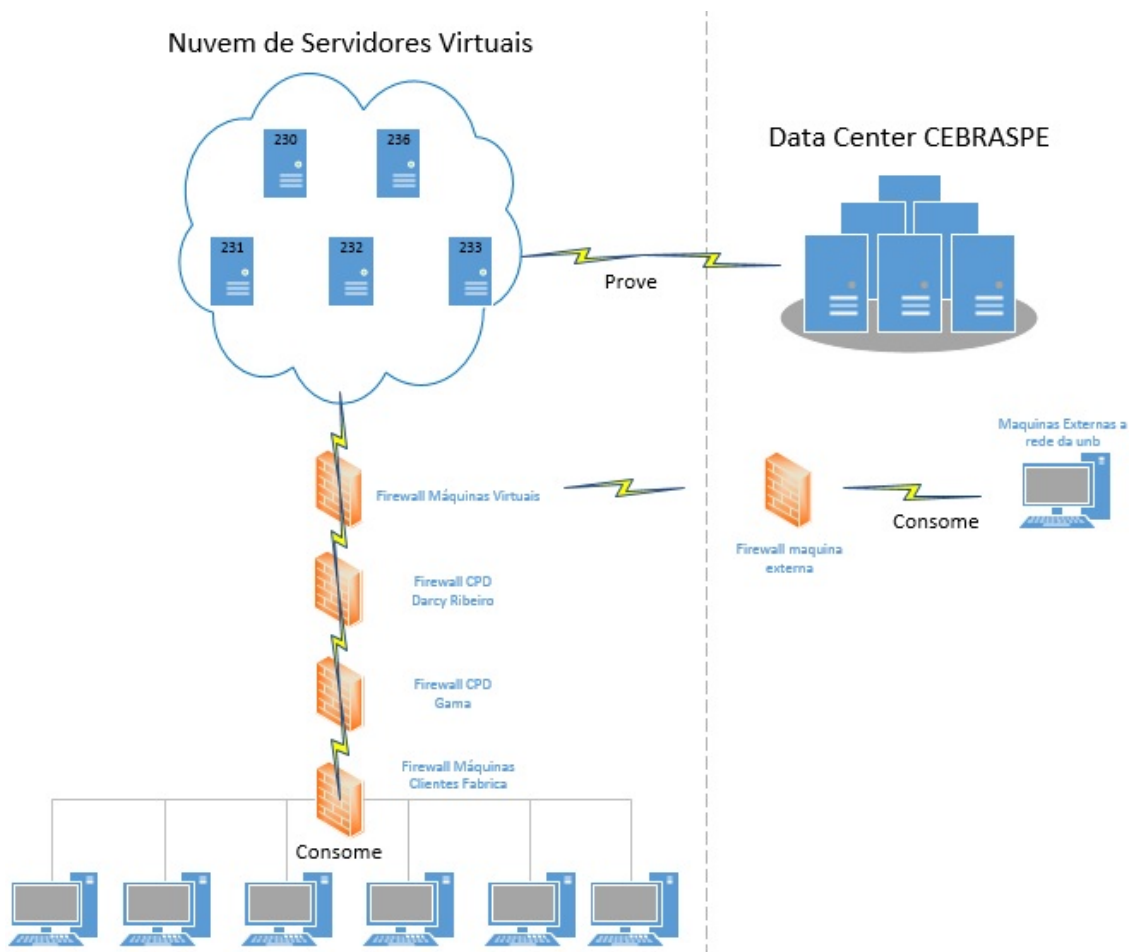


Figura 11: Arquitetura física de Serviços (VERGARA; CANEDO; FREITAS, 2014)

5.1.2.2 Firewalls

Uma das barreiras de proteção desta implantação são os firewalls que protegem a rede, quando um usuário da fábrica tenta usar uma máquina virtual ele passa por até 5 firewalls diferentes, e eles são descritos abaixo:

1. *Firewall* das máquinas da Fábrica de software: O usuário da fábrica de software, a partir de sua estação de trabalho, passará por 4 *firewalls* principais. O Primeiro deles é o
2. *Firewall* da própria máquina da estação de trabalho.
3. *Firewall* CPD UnB Gama Por estar presente dentro da rede da UnB Gama, existe um *firewall* que protege todos os computadores da Rede UnB Gama.
4. *Firewall* CPD UnB Darcy Após passar pelo *firewall* da UnB Gama, os dados passam para o seu *firewall* hierarquicamente mais alto. Este *firewall* protege todos os computadores da rede da UnB inteira. Talvez este seja o *firewall* mais difícil de ser controlado, pois é o mais longe do nosso domínio.
5. *Firewall* das máquinas virtuais. Após passar por todos os *firewalls* chegamos ao mais importante deles, O *firewall* das máquinas virtuais. Este *firewall* é o *firewall* presente em cada máquina virtual da nuvem, sendo este o *firewall* mais importante de todos os *firewalls*, pois ao se acessar estas máquinas de uma rede externa a rede da UnB, este será o único *firewall* entre o usuário mal intencionado e os nossos servidores.

5.1.3 Arquitetura de Serviços

Na figura 12 pode ser observado como as aplicações estão "conversando" entre si, ou seja, quais são as relações lógicas entre elas. Para utilizar um serviço da fábrica, basta o usuário digitar no seu navegador o IP do servidor de Produção e escolher qual a aplicação ele deseja utilizar. Porém para proteger os servidores de ataques eles são bloqueados de acesso por fora, via SSH por exemplo, somente é possível acessar algum servidor através do log-in no servidor de VPN com usuário, senha e certificado digital de chave privada, protegendo assim a rede.

5.2 Análise da implantação

Considerando o que foi apresentado no modelo de implantação e o que realmente foi implantado verifica-se que tudo o que era esperado que fosse implantado referente a SaaS realmente foi, e ainda com considerações que não eram esperadas no modelo.

Arquitetura Lógica de Serviços

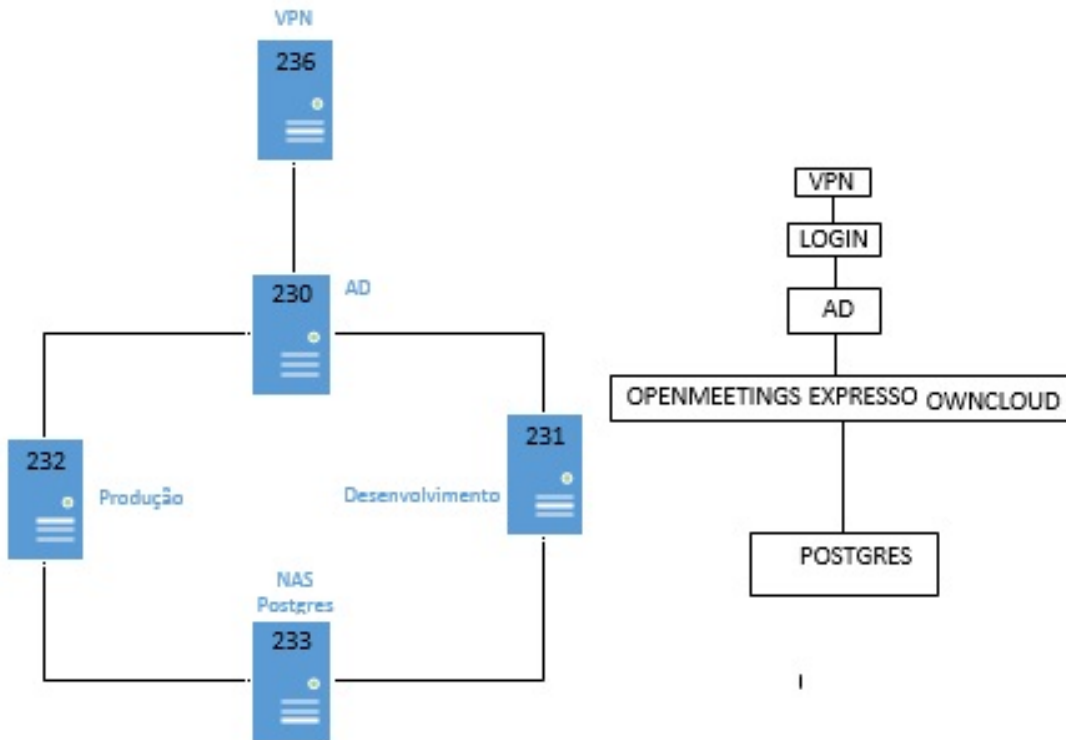


Figura 12: Arquitetura lógica de Serviços (VERGARA; CANEDO; FREITAS, 2014)

Primeiramente em relação aos serviços, nesta implantação do modelo temos a ideia de dois servidores, um para desenvolvimento e um para produção. Atualmente o Open meetings e o OwnCloud já estão em produção enquanto o Espresso, por razões de a fábrica ainda não ter um servidor de SMTP ainda está em desenvolvimento. Um fator importante que deve ser considerado é o fato de que todos os logins estão integrados em um servidor de log-in, isso facilita imensamente desde a criação de novos usuários nos serviços da fábrica quanto na deleção dos mesmos, não necessitando a criação/deleção de usuários em cada aplicação. E por último um dos fatores mais importantes que acabou surgindo foi a criação do servidor de VPN, que primeiramente surgiu para sanar um problema de acesso a porta 23 (SSH) e acabou sendo uma porta de entrada para todos que desejaram acessar algum servidor, sendo assim uma proteção a mais para a rede.

6 Conclusão e Trabalhos Futuros

A cada dia que passa a computação em nuvem tem estado mais presente em nossas vidas, não só para uso pessoal, mas cada vez mais no uso profissional. Por mais que a computação em nuvem traga muitos benefícios, ela traz também inúmeros desafios. Um dos desafios importantes é em relação a segurança, principalmente depois dos escândalos no Brasil e no mundo em relação a espionagem.

Neste trabalho foi feito um apanhado geral de ferramentas que implementam os 3 principais tipos de arquitetura da nuvem, e posteriormente a criação de um modelo de implantação para a Fábrica de Software do Gama. Este trabalho teve como foco a implantação dos Softwares como Serviço propostos neste modelo e que com base na análise feita pode-se dizer que foram feitas de maneira bem sucedida, pois além de fazer tudo o que estava planejado que era a implantação do OwnCloud e do Expresso, esta implantação trouxe vários elementos principalmente de segurança.

Ficou evidente neste trabalho que a implantação de serviços em uma nuvem privada não é um trabalho simples e que precisa de muita dedicação e paciência, principalmente se você deseja que ela esteja cada vez mais segura. Neste trabalho é mostrado algumas coisas que podem ser feitas para que a segurança seja aumentada, como o acesso pela terceira parte (VPN) mascarando os servidores da rede, e a proteção através de *firewalls*.

Uma boa evidência do sucesso deste trabalho é que não só os usuários da fábrica estão utilizando o OwnCloud como outros laboratórios de Engenharia de Software do Gama também se interessaram e já estão utilizando o sistema para armazenamento de arquivos, por ser mais seguro. Outro fator relevante deste trabalho é que o modelo de implantação proposto gerou um artigo que foi publicado em um congresso internacional de Cloud Computing.

6.1 Trabalhos Futuros

Como proposta de trabalhos futuros ficam principalmente 2 pontos principais.

O primeiro é terminar a implantação do SaaS proposto no modelo de implantação para a fábrica. Para terminar esta implantação falta a criação de um servidor de SMTP para que o Expresso tenha seu funcionamento de troca de emails funcional, e também da instalação de um servidor de BigBlueButton para a instalação do módulo de vídeo conferência do Expresso. Desta maneira a implantação do SaaS pode ser considerada completa.

Um segundo ponto para trabalho futuro é a implantação do IaaS e do PaaS, previstos no modelo de de implantação para a fábrica. Estes serviços não foram implantados por não serem tão importantes frente aos serviços de SaaS e por haver pouco tempo para a implantação.

Referências

- AGRAWAL, D.; DAS, S.; ABBADI, A. E. Big data and cloud computing: new wine or just new bottles? *Proceedings of the VLDB Endowment*, VLDB Endowment, v. 3, n. 1-2, p. 1647–1648, 2010. Citado na página 40.
- BADGER, L. et al. Draft cloud computing synopsis and recommendations. *Recommendations of the National Institute of Standards and Technology*, 2011. Citado na página 28.
- BRANTNER, M. et al. Building a database on s3. ACM, 2008. Citado na página 23.
- BUYYA, R.; BROBERG, J.; GOSCINSKI, A. M. *Cloud computing: Principles and paradigms*. [S.l.]: John Wiley & Sons, 2010. Citado na página 27.
- CANEDO, E. D. Modelo de confiança para a troca de arquivos em uma nuvem privada. 2013. Citado 3 vezes nas páginas 35, 36 e 40.
- CEBRASPE. 2014. Disponível em: <<http://www.cespe.unb.br/>>. Citado na página 57.
- CHIRIGATI, F. S. Computação em nuvem. *Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ*, 2009. Citado na página 34.
- COOPER, B. F. et al. Benchmarking cloud serving systems with ycsb. In: ACM. *Proceedings of the 1st ACM symposium on Cloud computing*. [S.l.], 2010. p. 143–154. Citado na página 41.
- DECANDIA, G. et al. Dynamo: amazon’s highly available key-value store. In: ACM. *ACM SIGOPS Operating Systems Review*. [S.l.], 2007. v. 41, n. 6, p. 205–220. Citado na página 35.
- DIKAIAKOS, M. D. et al. Cloud computing: distributed internet computing for it and scientific research. *Internet Computing, IEEE*, IEEE, v. 13, n. 5, p. 10–13, 2009. Citado na página 30.
- EXPRESSO Livre. 2014. Disponível em: <<http://www.expressolivre.org/>>. Citado 2 vezes nas páginas 48 e 51.
- GHEMAWAT, S.; GOBIOFF, H.; LEUNG, S.-T. The google file system. In: ACM. *ACM SIGOPS Operating Systems Review*. [S.l.], 2003. v. 37, n. 5, p. 29–43. Citado na página 35.
- GLOBO.COM. *Tsuru*. 2014. Disponível em: <<http://www.tsuru.io/>>. Citado na página 47.
- HEROKU. 2014. Disponível em: <<https://www.heroku.com/>>. Citado na página 47.
- LIU, S.; LIANG, Y.; BROOKS, M. Eucalyptus: a web service-enabled e-infrastructure. 2007. Citado na página 46.
- MARINOS, A.; BRISCOE, G. Community cloud computing. In: *Cloud Computing*. [S.l.]: Springer, 2009. p. 472–484. Citado na página 33.

- MARINOS, A.; BRISCOE, G. Community cloud computing. In: *Cloud Computing*. [S.l.]: Springer, 2009. p. 472–484. Citado 2 vezes nas páginas 15 e 33.
- MELL, P.; GRANCE, T. Draft nist working definition of cloud computing. *Referenced on June. 3rd*, v. 15, 2009. Citado 3 vezes nas páginas 32, 33 e 34.
- MICROSOFT. *Microsoft Azure*. 2014. Disponível em: <<http://azure.microsoft.com/pt-br/>>. Citado 2 vezes nas páginas 36 e 45.
- OPENSTACK Cloud Software. 2014. Disponível em: <<https://www.openstack.org/>>. Citado na página 46.
- OWNCLOUD. 2014. Disponível em: <<http://owncloud.org/>>. Citado 2 vezes nas páginas 48 e 53.
- S3. 2014. Disponível em: <<http://aws.amazon.com/pt/s3/>>. Citado na página 35.
- SERPRO. 2014. Disponível em: <<https://www.serpro.gov.br/>>. Citado 2 vezes nas páginas 50 e 57.
- VECCHIOLA, C.; CHU, X.; BUYYA, R. Aneka: a software platform for .net-based cloud computing. *High Speed and Large Scale Scientific Computing*, IOS Press, Amsterdam, Netherlands, p. 267–295, 2009. Citado na página 23.
- VELTE, T.; VELTE, A.; ELSENPETER, R. *Cloud computing, a practical approach*. [S.l.]: McGraw-Hill, Inc., 2009. Citado 2 vezes nas páginas 15 e 28.
- VERGARA, G.; CANEDO, E.; FREITAS, S. Deployment of secure collaborative softwares as a service in a private cloud to a software factory. In: *CLOUD COMPUTING 2014, The Fifth International Conference on Cloud Computing, GRIDs, and Virtualization*. [S.l.: s.n.], 2014. p. 123–129. Citado 7 vezes nas páginas 15, 49, 50, 54, 57, 58 e 60.
- WANG, C. et al. Privacy-preserving public auditing for data storage security in cloud computing. In: IEEE. *INFOCOM, 2010 Proceedings IEEE*. [S.l.], 2010. p. 1–9. Citado na página 35.
- XEN Hypervisor Project. 2014. Disponível em: <<http://www.xenproject.org/developers/teams/hypervisor.html>>. Citado na página 46.
- ZHOU, M. et al. Security and privacy in cloud computing: A survey. In: IEEE. *Semantics Knowledge and Grid (SKG), 2010 Sixth International Conference on*. [S.l.], 2010. p. 105–112. Citado 2 vezes nas páginas 15 e 31.

Apêndices

APÊNDICE A – Primeiro Apêndice

A.1 Detalhes dos servidores da nuvem

- AD

IP: 164.41.255.230

SO name | Microsoft Windows Server 2012 Datacenter processador | Intel Xeon x5660 2,8 GHZ 2 Cores 2 logical processors Memoria fisica RAM | 900MB Memoria virtual | 1,69GB Disk: 100GB Expansível

- NAS

IP: 164.41.255.231

SO name | Debian 7 processador | Intel Xeon x5660 2,8 GHZ 4 Cores Memoria fisica RAM | 4054668 kB Memoria Swap | 1952764 kB

Disk: Sist. Arq. Tam rootfs 14G /dev/disk/by-uuid/99b9b14f-7f14-4f61-b76e-dc705ba3fce6 14G /dev/sda4 19G /dev/sda3 65G

- Desenvolvimento

IP: 164.41.255.232

SO name | Debian 7 processador | Intel Xeon x5660 2,8 GHZ 4 Cores Memoria fisica RAM | 4042288 kB Memoria virtual | 521212 kB

Disk: Filesystem Tam /dev/sda1 98G udev 2,0G

- Aplicação

IP: 164.41.255.233

SO name | Debian 7 processador | Intel Xeon x5660 2,8 GHZ 4 Cores Memoria fisica RAM | 4042288 kB Memoria virtual | 1952764 kB

Disk: Sist. Arq. Tam rootfs 14G /dev/disk/by-uuid/7faf5e63-f9ba-4a6b-8cad-1c76b5e8d13f 14G /dev/sda4 694G /dev/sda3 276G

APÊNDICE B – Instalações

B.1 Instalação redmine

1. Seguir os passos de instalação tal como colocado no link:

http://www.redmine.org/projects/redmine/wiki/HowTo_Install_Redmine_in_Ubuntu

2. Depois criar o arquivo `/usr/share/redmine/public/.htaccess` (owner `www-data.www-data`), com o seguinte conteúdo:

```

1 # General Apache options
2 AddHandler fcgid-script .fcgi
3 AddHandler cgi-script .cgi
4 Options +SymLinksIfOwnerMatch +ExecCGI
5
6 # If you don't want Rails to look in certain directories,
7 # use the following rewrite rules so that Apache won't rewrite
8 #   certain requests
9 #
10 # Example:
11 #   RewriteCond %{REQUEST_URI} ^/notrails.*
12 #   RewriteRule .* - [L]
13
14 # Redirect all requests not available on the filesystem to Rails
15 # By default the cgi dispatcher is used which is very slow
16 #
17 # For better performance replace the dispatcher with the fastcgi
18 #   one
19 #
20 # Example:
21 #   RewriteRule ^(.*)$ dispatch.fcgi [QSA,L]
22 RewriteEngine On
23
24 RewriteBase /redmine
25 RewriteRule ^$ index.html [QSA]
26 RewriteRule ^([\^.]*)$ $1.html [QSA]
27 RewriteCond %{REQUEST_FILENAME} !-f
28 RewriteRule ^(.*)$ dispatch.fcgi [QSA,L]

```

```
28 # In case Rails experiences terminal errors
29 # Instead of displaying this message you can supply a file here
    which will be rendered instead
30 #
31 # Example:
32 #   ErrorDocument 500 /500.html
33
34 ErrorDocument 500 "<h2>Application error</h2>Rails application
    failed to start properly"
```

B.2 Integração Moodle com AD

Valores de configuração da integração do MOODLE com o AD

```
1 URL: ldap://ad.fabrica.unb.br:389
2 Versao: 3
3 TLS: Nao
4 codificacao LDAP: cp1252
5 Tamanho pagina: 250
6 Esconder senhas: Nao
7 Nome Distinto: CN=VPNReadOnlyUser,OU=Fabrica,DC=fabrica,DC=unb,DC
    =br
8 Senha
9 Tipo de usuario: MS Active Directory
10 Contextos: ou=fabrica,dc=fabrica,dc=unb,dc=br
11
12 Procurar por subcontexto: Sim Atributo de usuario: samaccountname
13 Atributo de membro: <vazio> Atributo do membro usando DN:
14 (&(|(objectclass=group))(|(cn=ADMINS)(cn=FSW)(cn=MC)(cn=OTRS)))
15 classe do objeto: objectclass=*
```

B.3 Integração OPENVPN com AD

Valores de configuração da integração do OPENVPN com o AD

Arquivo /etc/openvpn/server.conf

```
1 mode server
2 port 443
3 proto tcp
4 dev tun
5
```



```

6 ca /etc/openvpn/easy-rsa/keys/ca.crt # generated keys
7 cert /etc/openvpn/easy-rsa/keys/server.crt
8 key /etc/openvpn/easy-rsa/keys/server.key # keep secret
9 dh /etc/openvpn/easy-rsa/keys/dh1024.pem
10
11 script-security 2
12 plugin /usr/lib/openvpn/openvpn-auth-ldap.so "/etc/openvpn/ldap-
    auth.config"
13
14 server 10.9.8.0 255.255.255.0 # internal tun0 connection IP
15 ifconfig-pool-persist ipp.txt
16
17 push "route 164.41.255.230 255.255.255.255"
18 push "route 164.41.255.231 255.255.255.255"
19 push "route 164.41.255.232 255.255.255.255"
20 push "route 164.41.255.233 255.255.255.255"
21 push "route 164.41.255.234 255.255.255.255"
22 push "route 164.41.255.235 255.255.255.255"
23 push "route 164.41.255.236 255.255.255.255"
24
25 keepalive 10 120
26
27 comp-lzo # Compression - must be turned on at both end
28 persist-key
29 persist-tun
30
31 status /var/log/openvpn-status.log
32 log-append /var/log/openvpn-messages.log
33 log /var/log/openvpn-logs.log
34
35 verb 3 # verbose mode
36 client-to-client

```

Arquivo /etc/openvpn/ldap-auth.config

```

1 <LDAP>
2 # LDAP server URL
3 URL ldap://ad.fabrica.unb.br:389
4
5 #BindDN uid=admin,ou=Users,dc=test,dc=com
6 #BindDN "CN=VPNReadOnlyUser,DC=fabrica,DC=unb,DC=
    br"
7 BindDN VPNReadOnlyUser@fabrica.unb.br

```

```
8
9     # Bind Password
10    Password      2014@FSW
11
12    # Network timeout (in seconds)
13    Timeout       15
14
15    # Enable Start TLS
16    TLSEnable     no
17
18    # Follow LDAP Referrals (anonymously)
19    FollowReferrals yes
20 </LDAP>
21
22 <Authorization>
23     # Base DN
24     #BaseDN      "CN=Users,DC=test,DC=com"
25     #BaseDN      "DC=fabrica,DC=unb,DC=br"
26     BaseDN       "DC=fabrica,DC=unb,DC=br"
27
28     # User Search Filter
29     #SearchFilter "(&(uid=%u)(accountStatus=active))"
30     #SearchFilter "(&(sAMAccountName=%u)(msNPAllowDialin=
31     TRUE))"
32     SearchFilter  "(&(sAMAccountName=%u))"
33
34     # Require Group Membership
35     RequireGroup true
36
37     # Add non-group members to a PF table (disabled)
38     #PFTable     ips_vpn_users
39
40     <Group>
41         BaseDN      "DC=fabrica,DC=unb,DC=br"
42         SearchFilter "(&(CN=VPN))"
43         MemberAttribute "member"
44         # Add group members to a PF table (disabled)
45         #PFTable     ips_vpn_eng
46     </Group>
47 </Authorization>
```

B.4 Integração OWNCLOUD com AD

Valores de configuração da integração do OWNCLOUD com o AD

```
1 Server
2     URL: ad.fabrica.unb.br:389
3     Nome Distinto: CN=VPNReadOnlyUser,OU=Fabrica,DC=fabrica,
4         DC=unb,DC=br
5     Senha
6     Contexto: DC=fabrica,DC=unb,DC=br
7
8 User Filter
9     classes de objeto: person, user
10    grupos: ADMINS, FSW, MC, OTRS
11    Filtro Raiz (direto): (&(|(objectclass=user)(objectclass=
12        user))(|(memberof=CN=ADMINS,OU=Fabrica,DC=fabrica,DC=
13        unb,DC=br)(memberof=CN=FSW,OU=Fabrica,DC=fabrica,DC=unb
14        ,DC=br)(memberof=CN=MC,OU=Fabrica,DC=fabrica,DC=unb,DC=br)
15        (memberof=CN=OTRS,OU=Fabrica,DC=fabrica,DC=unb,DC=br)
16        )))
17
18 Login Filter
19    Usuario LDAP: <check box ON>
20    Outros Atributos: sAMAccountName
21    Filtro Raiz (direto): (&(&(|(objectclass=person)(
22        objectclass=user))(|(memberof=CN=ADMINS,OU=Fabrica,DC=
23        fabrica,DC=unb,DC=br)(memberof=CN=FSW,OU=Fabrica,DC=
24        fabrica,DC=unb,DC=br)(memberof=CN=MC,OU=Fabrica,DC=
25        fabrica,DC=unb,DC=br)(memberof=CN=OTRS,OU=Fabrica,DC=
26        fabrica,DC=unb,DC=br)))(|(samaccountname=%uid)(|(
27        sAMAccountName=%uid))))
28
29 Group Filter
30    classe de objetos: group
31    grupos: ADMINS, FSW, MC, OTRS
32    Filtro Raiz (direto): (&(|(objectclass=group))(|(cn=
33        ADMINS)(cn=FSW)(cn=MC)(cn=OTRS)))
```