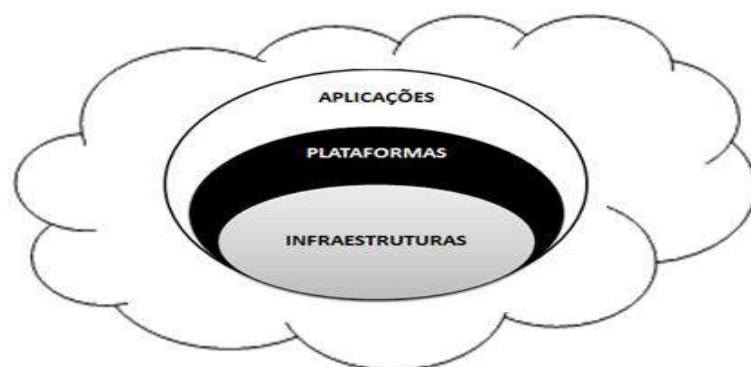




INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

**Área Departamental de Engenharia Eletrónica
e Telecomunicações e de Computadores**



Estudo do Paradigma Computação em Nuvem

Carina Nobre Gomes
(Licenciado)

Dissertação de natureza científica realizada para obtenção do grau de
Mestre em Engenharia Informática e de Computadores

Orientador

Professor Porfírio Pena Filipe

Júri

Presidente: Professor Fernando Manuel Gomes de Sousa

Vogais: Professor Carlos Jorge de Sousa Gonçalves

Professor Porfírio Pena Filipe

Dezembro de 2012

Na atual conjuntura económica, onde a globalização convive com a crise, as empresas confrontam-se com dois indeclináveis desafios, a expansão para novos mercados e a redução dos custos. A inevitabilidade de lidar com uma crescente quantidade de informação, na manutenção dos serviços prestados e na implementação de outros, obriga a uma sofisticada evolução dos meios informáticos. Para evoluir de forma pouco onerosa, é imprescindível a adoção de infraestruturas computacionais ágeis.

Neste contexto emergem estratégias sustentadas na adoção do paradigma Computação em Nuvem (CN). Este paradigma sugere infraestruturas virtuais, escaláveis e com gestão automática de recursos, partilhadas no mesmo modelo de negócio. A forma de definir os custos, designada por *pay as you go*, é baseada no uso. Procurando garantir uma constante adaptação às exigências do negócio, a CN proporciona confiança e qualidade de serviço, reduzindo o risco associado ao lançamento de aplicações e o tempo de resposta.

O objetivo deste trabalho é estudar o paradigma CN e perscrutar a sua projeção num futuro próximo, analisando as suas vantagens e inconvenientes. Nesse âmbito, é proposta uma arquitetura para integrar equipamentos de bilhética empregues para, designadamente, vender, validar e fiscalizar títulos de transportes. Para avaliar a arquitetura proposta foi implementado um demonstrador na plataforma *Windows Azure*.

PALAVRAS-CHAVE

Computação em Nuvem, Integração de Equipamentos, Bilhética, Transportes Públicos Coletivos de Passageiros.

ABSTRACT

Actual economic conjuncture, where globalization keeps pace with crisis, companies face two undeniable challenges: growth into new markets and reduction of costs. The unavailability of dealing with a growing amount of information, about maintenance of accomplished services and implementation of others, makes a sophisticated improvement of the computational means come true. In order to succeed in a less expensive way, it is inevitable to adopt agile computing infrastructures.

In this context sustained strategies emerge in adopting the paradigm of Cloud Computing (CC). It proposes the support of virtual infrastructures, scalable and with automatic management of resources, shared in the same business model. The way to define its payment, named ‘pay as you go’, is based on use. Looking ensures a constant adaptation to the demands of the business, the CC provides reliable and quality service, reducing the risk associated with launching applications and the response time.

The purpose of this work is to study the CC and to scrutinize its projection into the near future, analyzing their advantages and handicaps. In this context, is proposed architecture to integrate equipment used for ticketing, namely, sell, validate and monitor transport titles. To assess the architecture was implemented on a demonstrator on the Windows Azure platform.

KEYWORDS

Cloud Computing, Integrated Equipment, Ticketing, Public Collective Passenger Transport.

AGRADECIMENTOS

Agradeço às pessoas e entidades que tornaram possível a realização deste trabalho, às quais manifesto a minha especial gratidão:

- Ao Doutor Porfírio Pena Filipe, pela sua irrepreensível orientação;
- Ao Engenheiro Manuel Messias por toda a disponibilidade;
- Aos colegas Vítor Tomás, Rui Santos e Agostinho Baía, por todas as sugestões que tiveram a gentileza de me apresentar;
- À empresa *Link* (1), pela atenção dedicada e apoio prestado;
- À minha família, e em especial ao meu marido, pelo constante encorajamento, e também pela sua tolerância para com a minha reduzida disponibilidade familiar durante a realização deste trabalho;
- A todos os demais que, de alguma forma, contribuíram para o desenvolvimento deste trabalho.

ACRÓNIMOS

AAA	<i>Authentication, Authorization and Accounting Protocol</i>
AC	<i>Access Control</i>
ACL	<i>Access Control List</i>
ACS	<i>Access Control Service</i>
AMI	<i>Amazon Machine Image</i>
ARPANet	<i>Advanced Research Projects Agency Network</i>
AWS	<i>Amazon Web Services</i>
BLOB	<i>Binary Large Object</i>
capex	<i>Capital expenditure</i>
CC	<i>Computação em Cluster</i>
CDN	<i>Content Delivery Network</i>
CG	<i>Computação em Grade</i>
CN	<i>Computação da Nuvem</i>
CRUD	<i>Operações: Create, Read, Update e Delete.</i>
CU	<i>Computação de Utilidade</i>
DFS	<i>Distributed and replicated File System</i>
DOM	<i>Data Object Model</i>
EAV	<i>Entity-Attribute-Value</i>
EBS	<i>Elastic Block Store</i>
EC2	<i>Elastic Compute Cloud</i>
EGT	<i>Entity Group Transaction</i>
ERP	<i>Enterprise Resource Planning</i>
ESB	<i>Enterprise Service Bus</i>
FastCGI	<i>Fast Common Gateway Interface</i>
FIFO	<i>First In, First Out</i>
GAE	<i>Google App Engine</i>
GUID	<i>Globally Unique Identifier</i>
GRS	<i>Geo Redundant Storage</i>
HITs	<i>Human Intelligence Tasks</i>
HTML	<i>HyperText Markup Language</i>

HTTP(S)	<i>Hypertext Transfer Protocol (Secure)</i>
IaaS	<i>Infrastructure as a Service</i>
IDE	<i>Integrated Development Environment</i>
IIS	<i>Internet Information Services</i>
IPSec	<i>Internet Protocol Security</i>
IPv4	<i>Internet Protocol version 4</i>
IPv6	<i>Internet Protocol version 6</i>
JSON	<i>JavaScript Object Notation</i>
LAN	<i>Local Area Network</i>
NAT	<i>Network Address Translation</i>
NTFS	<i>New Technology File System</i>
OData	<i>Open Data Protocol</i>
opex	<i>Operational Expenditure</i>
PaaS	<i>Platform as a service</i>
PAYG	<i>Pay as you go</i>
QoS	<i>Quality of service</i>
REST	<i>Representational State Transfer</i>
ROI	<i>Return on Investment</i>
S3	<i>Simple Storage Service</i>
SaaS	<i>Software as a service</i>
SB	<i>Service Bus</i>
SBT	<i>Service Bus Topics</i>
SLA	<i>Service Level Agreement</i>
SDK	<i>Software Development Kit</i>
SOA	<i>Service Oriented Architecture</i>
SOAP	<i>Simple Object Access Protocol</i>
SSRS	<i>SQL Server Reporting Services</i>
SSL	<i>Secure Sockets Layer</i>
STRIDE	<i>Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege</i>
TCO	<i>Total Cost of Ownership</i>
TI	<i>Tecnologias de Informação</i>
URL	<i>Unified Resource Locator</i>
WAFC	<i>Windows Azure Fabric Controller</i>

WA	<i>Windows Azure</i>
WAC	<i>Windows Access Control</i>
WAN	<i>Wide Area Network</i>
WAQ	<i>Windows Azure Queue</i>
WASABi	<i>Windows Azure Autoscaling Application Block</i>
WASBT	<i>WA Service Bus Topic</i>
WASBQ	<i>Windows Azure Service Queue</i>
WCF	<i>Windows Communication Foundation</i>
WIF	<i>Windows Identity Foundation</i>
WSDL	<i>Web Services Description Language</i>
XML	<i>eXtensible Markup Language</i>
XSD	<i>XML Schema Definition</i>

CONVENÇÕES TIPOGRÁFICAS

Apresentam-se as convenções tipográficas empregues neste documento:

- Aplica-se no texto normal a fonte *Times New Roman*, o tamanho 12 e o espaçamento entre linhas 1,5 cm

Exemplo: Texto normal;

- Aplicam-se parêntesis curvos nas referências bibliográficas

Exemplo: (Nome1, Nome2 e Nome 3, ano da publicação);

- Aplica-se texto em itálico, delimitado por aspas, nas frases ou expressões ilustrativas

Exemplo: “*Frase ilustrativa*”;

- Aplica-se texto em itálico nas designações em inglês

Exemplo: *Cloud Computing*;

- Aplica-se texto em negrito para definir acrónimos e siglas

Exemplo: *Software as a Service* (**SaaS**).

ÍNDICE DO CONTEÚDO

1	PRÓLOGO	1
1.1	Motivação	1
1.2	Objetivo e Contribuição	2
1.3	O Paradigma da Computação em Nuvem.....	2
1.4	Organização do Documento	3
2	COMPUTAÇÃO EM NUVEM.....	5
2.1	História	5
2.2	Evolução do Paradigma Computação em Nuvem.....	6
2.2.1	Virtualização	7
2.2.2	Computação em <i>Cluster</i>	9
2.2.3	Computação em Grade.....	10
2.2.4	Computação de Utilidade.....	12
2.2.5	Computação em Nuvem.....	13
2.3	Análise SWOT	16
3	TIPOS DE CARREGAMENTO DA NUVEM	17
3.1	Introdução	17
3.2	Nuvem Privada.....	18
3.3	Nuvem Pública.....	20
3.4	Nuvem Híbrida.....	21
4	NÍVEIS DE SERVIÇO DA NUVEM	23
4.1	Introdução	23
4.2	<i>Software as a Service</i>	24
4.3	<i>Platform as a Service</i>	26
4.4	<i>Infrastructure as a Service</i>	27
5	O MERCADO.....	29
5.1	<i>Google App Engine</i>	29
5.2	<i>Amazon Web Services</i>	30
5.3	<i>Joyent</i>	32
5.4	<i>Eucalyptus</i>	32
5.5	<i>Windows Azure</i>	33
5.6	Comparação de Fornecedores	35
6	A PLATAFORMA <i>WINDOWS AZURE</i>	37

6.1	Arquitetura dos Serviços.....	38
6.2	Sistema <i>Windows Azure</i>	40
6.3	<i>Windows Azure AppFabric</i>	40
6.4	<i>SQL Database</i>	41
6.5	<i>Windows Azure Marketplace</i>	44
6.6	Processadores	44
6.7	Filas de Mensagens	45
6.8	<i>Windows Azure Auto-Scaling Block</i>	46
7	PROPOSTA	47
7.1	Intervenientes	47
7.2	Arquitetura	48
7.2.1	Equipamento.....	50
7.2.2	Protocolo de Integração	51
7.2.3	Protocolo de Interação	54
7.2.4	Mensagens	55
7.2.5	Modelo dos Objetos de Dados.....	56
7.3	Demonstrador.....	58
7.3.1	Equipamento Genérico.....	58
7.3.2	Aplicação de Administração	59
7.3.3	Componentes da Plataforma <i>Windows Azure</i>	60
7.4	Considerações Finais	64
8	EPÍLOGO	67
8.1	Avaliação de Custos	67
8.2	Conclusão.....	69
8.3	Perspetivas Futuras.....	69
	ANEXO A – REGRAS DE VALIDAÇÃO DA MENSAGEM.....	71
	ANEXO B – REGRAS DE VALIDAÇÃO DOS OBJETOS DE DADOS	73

ÍNDICE DE FIGURAS

Figura 1 - Evolução da computação	6
Figura 2 - Esquema de evolução do paradigma CN.....	7
Figura 3 - Modelo de camadas da estrutura CC.....	10
Figura 4 - Cenário “Ligado e Desligado”	14
Figura 5 - Cenário "Crescimento Rápido"	14
Figura 6 - Cenário "Carga Imprevista"	15
Figura 7 - Cenário "Carga Sazonal"	15
Figura 8 - Tipos de carregamento da nuvem	17
Figura 9 - Níveis de serviço da nuvem.....	23
Figura 10 - Componentes da plataforma WA	37
Figura 11 - Arquitetura dos serviços da plataforma WA	38
Figura 12 - Níveis de serviços da arquitetura proposta	47
Figura 13 - Arquitetura proposta.....	48
Figura 14 - Diagrama de sequência de um equipamento genérico	50
Figura 15 - Diagrama de classes do equipamento.....	51
Figura 16 - Diagrama de classes da nuvem	52
Figura 17- Modelo do ficheiro de configuração do equipamento.....	53
Figura 18 - Diagrama de sequência do protocolo de interação.....	54
Figura 19 - Modelo das mensagens.....	55
Figura 20 - Modelo do objeto <i>Register</i>	56
Figura 21 - Modelo do objeto <i>DeviceType</i>	57
Figura 22 - Modelo do objeto <i>LogTypeN</i>	57
Figura 23 - Equipamento genérico	59
Figura 24 - Aplicação de administração	60
Figura 25 - Arquitetura no contexto da plataforma WA	61

ÍNDICE DE TABELAS

Tabela 1 - Vantagens e desvantagens da virtualização	8
Tabela 2 - Comparação entre os paradigmas CG e CN.....	11
Tabela 3 - Análise SWOT do paradigma CN	16
Tabela 4 - Vantagens da nuvem privada	19
Tabela 5 - Aspetos a considerar na nuvem híbrida	21
Tabela 6 - Comparação de fornecedores CN	35
Tabela 7 - Comparação entre os diferentes tipos de armazenamento na WA	43
Tabela 8 - Legenda do equipamento genérico	59
Tabela 9 - Legenda da interface do operador	60
Tabela 10 - Avaliação de custos de alojamento na WA.....	68

1 PRÓLOGO

Atualmente, no mundo das Tecnologias da Informação (TI), procura-se garantir a mobilidade dos utilizadores e a oportunidade de disporem de uma gama cada vez mais alargada de serviços. Como estratégia para atingir este objetivo, surgiu o paradigma da Computação em Nuvem (CN), com o propósito de disponibilizar serviços, públicos ou privados, através da Internet (2).

O paradigma CN pretende criar um horizonte de oportunidades no mundo das TI, pelo fornecimento via Internet de recursos escaláveis, fiáveis e virtuais, abundantes e orientados para os serviços pretendidos. As empresas que pagam pelo uso dos recursos pagam apenas o que usam, durante o período estabelecido, com reduzidos investimentos iniciais (3).

A evolução para a CN tem levado a modificações significativas da tecnologia, alterando o ambiente operacional dos gestores das TI. Um dos desafios para os fornecedores de tecnologia consiste na disponibilização de soluções e serviços móveis, que permitam uma fácil integração nas infraestruturas e nos processos de negócio atuais. O principal objetivo é evitar o desenvolvimento de processos complexos, de custo acrescido. A propriedade de elasticidade, inerente ao paradigma CN, permite ajustar dinamicamente os recursos às necessidades dos negócios (4).

1.1 Motivação

As palavras “*recessão*” e “*crise*”, exaustivamente repetidas no nosso quotidiano, sintetizam o atual estado da economia. O mundo das TI não escapa à contração generalizada das atividades económicas.

Na procura de modelos de negócio mais lucrativos, as empresas diminuem o investimento nos bens de capital, (*capital expenditures* – **capex**), necessários para a implementação de serviços, e reduzem custos com uma melhor gestão das despesas operacionais (*operational expenditure* – **opex**) (5). Esta estratégia tem um impacto direto nas TI. Os custos capex, com a aquisição de recursos (máquinas, dispositivos de armazenamento e licenças de *software*) e a contratação dos seus gestores, são

substituídos por fornecedores externos de serviços escaláveis e adequados à ótica de negócio (6) (7) (8).

No âmbito da bilhética de transportes públicos e coletivos, a dificuldade em garantir a existência de serviços e recursos necessários para responder aos novos requisitos do mercado, aumenta os problemas das empresas que se arriscam a pôr em causa a satisfação dos clientes, por inexistência de recursos alocáveis, ou a desperdiçar meios financeiros num aprovisionamento excessivo (9).

1.2 Objetivo e Contribuição

O objetivo deste trabalho consiste em estudar um paradigma computacional emergente, a CN, e averiguar a sua aplicabilidade ao domínio da bilhética de transportes públicos coletivos de passageiros, mais concretamente à gestão integrada de equipamentos de bilhética, usados para, nomeadamente: vender, validar e fiscalizar títulos de transporte.

Para o efeito, é proposta uma arquitetura de referência, para integrar equipamentos de bilhética, implementada para fins de demonstração, sobre a plataforma *Windows Azure (WA)* (10).

1.3 O Paradigma da Computação em Nuvem

Sobre o paradigma CN destacam-se as definições:

- a. *“A computação em nuvem é um modelo que permite o acesso à rede sob demanda para um pool de recursos computacionais configuráveis partilhados (por exemplo, redes, servidores, armazenamento, aplicativos e serviços) que podem ser rapidamente provisionados e libertados com mínimo esforço de gestão ou de interação com o fornecedor de serviços.”* (11).
- b. *“Recursos de TI e serviços que são abstraídas da infraestrutura subjacente e são fornecidos "on-demand" e "em escala" em um ambiente de vários utilizadores.”* (12).

No âmbito do presente trabalho adotou-se a definição da alínea “a.” por ser mais completa e referir considerações sobre o modelo de negócio que é um aspeto fundamental a ter em atenção para o sucesso deste tipo de soluções.

1.4 Organização do Documento

Este documento está organizado em sete capítulos.

No primeiro capítulo faz-se uma introdução e define-se o paradigma CN, descreve-se a motivação, o objetivo e a contribuição do trabalho proposto. No final, apresenta-se a estrutura organizativa.

No segundo capítulo apresenta-se o paradigma CN, a história, a evolução de acordo com o contributo dos paradigmas anteriores, a identificação e descrição dos cenários do modelo e, por fim, uma análise SWOT do referido paradigma.

No terceiro capítulo referem-se os três tipos de carregamento da nuvem, salientando as vantagens e contextos em que são aplicados.

No quarto capítulo apresentam-se os três níveis de serviço, destacando aspetos relevantes associados a cada um deles, assim como possíveis exemplos de utilização.

No quinto capítulo mencionam-se os fornecedores atuais, discriminando principalmente as tecnologias, mecanismos de autenticação, segurança e modelos de negócio. A finalizar apresenta-se uma comparação sumária.

No sexto capítulo é feita uma análise da plataforma WA, salientando aspetos relevantes da arquitetura e dos serviços que disponibiliza.

No sétimo capítulo é apresentada a arquitetura proposta para a integração de equipamentos, assim como, a estrutura das mensagens e os protocolos de integração e de interação. Adicionalmente são apresentados o equipamento genérico e a interface do operador. Posteriormente é realizado o mapeamento da arquitetura proposta no modelo computacional da plataforma WA, salientando os componentes utilizados e respetivas justificações de emprego. Finalmente são realizadas considerações sobre o demonstrador vocacionado para a integração de equipamentos de bilhética.

No oitavo capítulo foca-se a avaliação de custos e são igualmente apresentadas conclusões e perspectivas de investigação futura.

2 COMPUTAÇÃO EM NUVEM

Neste capítulo aborda-se o paradigma CN, referindo aspetos históricos que motivaram a sua evolução. Conclui-se o capítulo fazendo uma comparação relativas aos paradigmas de computação.

2.1 História

No fim dos anos 60, a agência *Advanced Research Projects Agency Network*, (**ARPANet**), concebeu a criação de uma “*rede intergaláctica de computadores*”. A ideia consistia em interligar e aceder a qualquer recurso na Internet, a partir de qualquer local a qualquer momento. A agência pretendia garantir a interligação de tudo e todos, à escala mundial, via Internet (13) (14).

Na década de 70, o cientista de computação *John McCarthy* (criador do termo Inteligência Artificial e da linguagem *LIS Processing*) propõe o conceito de nuvem, que estaria tão disponível como os serviços de utilidade pública (15).

Em 1999, a empresa de aplicações empresariais *Customer Relationship Management* (**CRM**), com o endereço *salesforce.com*, fez surgir os primeiros serviços disponibilizados via Internet. Esta iniciativa introduziu uma nova ideia no mercado, a do fornecimento de aplicações empresariais com arquiteturas orientadas ao serviço (*Service Oriented Architecture* - **SOA**), através da Internet (16).

O passo seguinte, em 2002, foi o lançamento da *Amazon Web Services* (**AWS**). A partir da *Amazon Mechanical Turk*, a *Amazon* disponibilizou serviços ao nível do armazenamento, da computação e da inteligência humana (*Human Intelligence Tasks* - **HITs**). Quatro anos depois, a *Amazon* lançou no mercado um serviço comercial Web, o *ElasticComputeCloud* (**EC2**). Este serviço tinha o objetivo de facultar o aluguer de computadores a pequenos utilizadores empresariais, para a execução das suas aplicações informáticas (17) (18).

Em 2006 surgiu o *Simple Storage Service* (**S3**), um serviço Web da *Amazon* que disponibiliza uma infraestrutura de armazenamento escalável a custos reduzidos. A sua

forma de remuneração baseou-se no modelo *pay as you go* (**PAYG**), ou seja, o utilizador paga pelos recursos que utiliza durante o período contratado. Este modelo tornou-se no modelo de pagamento inerente à CN (19).

Após três anos, a empresa *Google* iniciou as suas ofertas de aplicações (*Google Apps*). Acedidas por navegadores, as aplicações disponibilizavam serviços fiáveis e fáceis de consumir pelos seus utilizadores. Com elas, e também com a gradual implementação da banda larga à escala mundial, deu-se o primeiro passo para a interligação e descoberta de serviços (20).

A Figura 1 ilustra os anos determinantes para o aparecimento do paradigma CN.

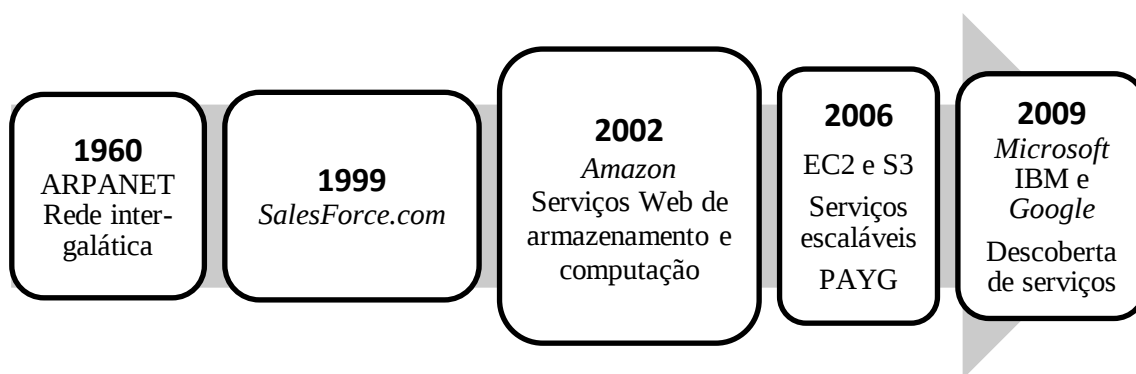


Figura 1 - Evolução da computação

Atualmente, a Internet é considerada um motor de crescimento económico. Dois dos marcos a destacar são o surgimento da política do *Internet Protocol version 6* (**IPv6**), devido à escassez de endereços *Internet Protocol version 4* (**IPv4**), e a definição de contratos controlados acerca da gestão da própria infraestrutura. Estas inovações proporcionaram o amadurecimento da tecnologia de virtualização, o desenvolvimento da banda larga ao nível mundial e mesmo o aparecimento de novos padrões de arquitetura, como é o caso da CN (21) (22).

2.2 Evolução do Paradigma Computação em Nuvem

O diagrama evolutivo da Figura 2 refere os paradigmas que contribuíram para o aparecimento da CN.

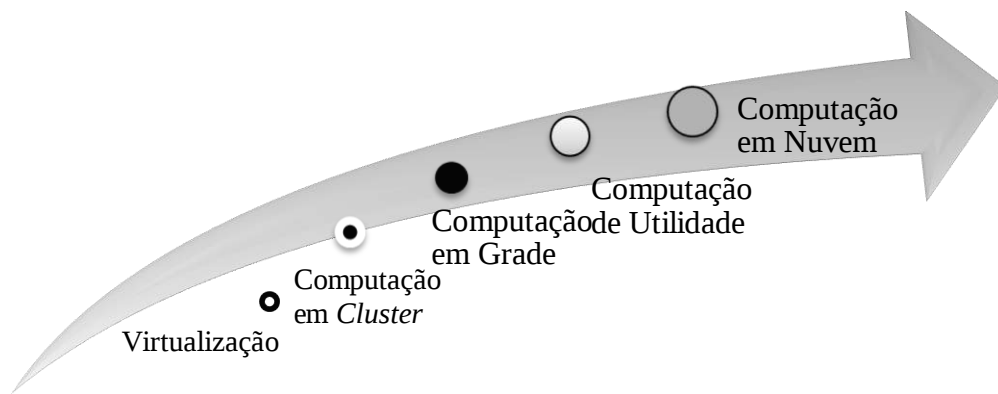


Figura 2 - Esquema de evolução do paradigma CN

Na secção seguinte faz-se uma análise, singular e comparada, dos modelos referindo os pontos fortes do paradigma CN, relativamente aos seus antecessores, com especial destaque na última subsecção (23).

2.2.1 Virtualização

A virtualização torna transparentes as características físicas de um ambiente computacional, decompondo-as em recursos lógicos, que posteriormente são rearranjados em máquinas virtuais. Com esta tecnologia interpõe-se uma camada *middleware* entre as camadas aplicacional e física, possibilitando acessos concorrentes à segunda. Simultaneamente, recursos virtuais e a sua abstração são disponibilizados aos utilizadores (24) (25).

Tipicamente, a virtualização é implementada a partir de uma entidade central, que atribui logicamente os recursos existentes, garantindo o isolamento entre utilizadores. Num ambiente de virtualização, a administração da infraestrutura é considerada um desafio, pois a inexistência de uma boa abordagem pode aumentar a complexidade e os custos. Para vencer este desafio, uma das possibilidades é a automação da entidade central, que tem como responsabilidade a gestão do ambiente físico, que fornece os recursos virtuais do servidor aos demais utilizadores (26) (27).

Esta tecnologia está associada ao modelo da CN, pois os centros de dados da nuvem aprovisionam uma rede de serviços, a utilizar quando solicitados. A sua disposição exige um suporte dinâmico que lhes é conferido pela virtualização. Desta forma, consegue-se a característica de escalabilidade, assim como a possibilidade de um melhor

aproveitamento e gestão da componente física dos servidores, face aos modelos tradicionais. A diminuição de custos, tanto ao nível de *hardware* como de energia, é devida ao mapeamento de um único recurso físico para várias máquinas virtuais. Recorrendo a estruturas de replicação, a virtualização possibilita a implementação de mecanismos de disponibilidade e de procedimentos de recuperação, após eventuais desastres (28) (29) (30).

O facto de o *hardware* ser usado por múltiplos sistemas operativos, pela execução de várias máquinas virtuais numa única máquina física, leva à redução da subutilização, assim como ao aumento de agilidade e distribuição, de recursos TI, adaptando-os às necessidades de negócio. Como consequência, esta tecnologia pode aumentar o retorno de investimento (*Return on Investment* - **ROI**) e reduzir o custo total de propriedade (*Total Cost of Ownership* - **TCO**) (31) (32).

Na Tabela 1 apresentam-se as vantagens e os inconvenientes do emprego da tecnologia de virtualização (16) (26) (33) (34).

Vantagens	Desvantagens
a. Redução do consumo global de energia, devido à existência de um menor número de servidores físicos. b. Favorecimento do tipo de ambiente multiutilizador e da partilha de recursos. c. Redução do tempo de inatividade do negócio, pelo realojamento dinâmico das aplicações. d. Escalabilidade e agilidade das aplicações, a partir de um ambiente virtual. As alterações são mais flexíveis do que nos modelos tradicionais. e. Acesso aos serviços, independentemente do local, viabilizado pela inexistência de vinculação com o suporte físico da infraestrutura. f. Redução das dificuldades do desenvolvimento das aplicações, conseguida pela diminuição do controlo da infraestrutura, através de metodologias de virtualização.	a. Problemática associada à gestão (e.g. consistência dos dados replicados) e segurança dos dados (e.g. mecanismos de acesso). b. Os servidores são considerados pontos críticos. c. Dificuldade no realojamento da infraestrutura, pois pode implicar a inatividade das máquinas virtuais. d. A energia e a refrigeração podem tornar-se inadequadas, após alteração dos padrões de desempenho do centro de dados. e. A eventual indisponibilidade dos servidores acarreta um serviço mais lento, insatisfatório para os utilizadores. Pode ocorrer redução do valor do negócio e perda de lucros.

Tabela 1 - Vantagens e desvantagens da virtualização

A partir de um ambiente virtual é possível, dinamicamente, criar, mover e gerir a dimensão da infraestrutura da nuvem. Estas características propiciam aos utilizadores funcionalidades, tais como: partilha, gestão e isolamento de recursos. A inexistência de vinculação aos recursos de *hardware* subjacentes facilita a movimentação das máquinas virtuais (35) (36) (37).

A criação de uma camada de abstração com todos os centros de dados, evitando a sua implementação como sistemas isolados, leva a que a virtualização seja uma estratégia cada vez mais adotada (3).

2.2.2 Computação em *Cluster*

A Computação em *Cluster* (*Cluster Computing* - CC) foi concebida para permitir o aumento da utilização, escalabilidade e disponibilidade de recursos computacionais individualizados. Fisicamente, um *cluster* é constituído por computadores interligados, que originam um único recurso computacional integrado. Este modelo é um sistema paralelo e distribuído, cujos recursos são localizados num domínio administrativo e gerido por uma organização (38).

Na arquitetura *cluster*, cada um dos computadores existentes é designado por nó existindo tipicamente um nó que gere e distribui tarefas entre os demais. Uma arquitetura típica de CC designa-se por *Beowulf*, trata-se de um aglomerado de computadores ou *clusters*, de computação paralela, usando computadores pessoais, não especializados e de baixo custo (6) (39) (40).

O modelo CC é um dos tipos de virtualização da nuvem. Neste, a partir de uma camada *middleware*, os servidores físicos são organizados como um servidor virtual. A camada gere os diversos componentes físicos, que dão origem a um único núcleo lógico representativo da tipologia *cluster* (28) (41).

A Figura 3 ilustra o modelo de camadas da estrutura CC.

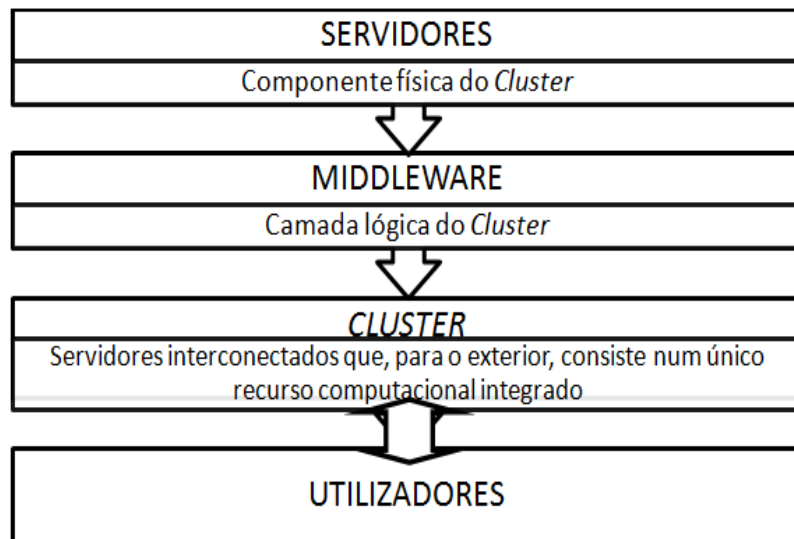


Figura 3 - Modelo de camadas da estrutura CC

Em síntese, o modelo admite mecanismos de redundância, virtualização e elasticidade de recursos, embora em escala diferente da do paradigma CN. Este último assegura a eficiente utilização do ambiente computacional, permitindo suportar diversas aplicações heterogêneas, inerentes à ótica de negócio do utilizador (42) (43).

2.2.3 Computação em Grade

O paradigma Computação em Grade (*Grid Computing* – **CG**) surgiu com o objetivo de integrar recursos heterogêneos e geograficamente distribuídos. A infraestrutura CG é sustentada por um agregado de *clusters*, constituídos por recursos heterogêneos e geograficamente dispersos (3).

O paradigma CC está focado na integração de serviços, proporcionando um suporte para redes de investigação para a produção de novas funcionalidades, nos domínios da Ciência e da Engenharia (14).

A CG é favorável ao trabalho colaborativo, à partilha de recursos computacionais e ao contexto económico, pois ajuda a criar novos modelos de negócio, através da comercialização de plataformas e *software* como serviços na Internet (16) (39).

Na Tabela 2 comparam-se os paradigmas CG e CN (44) (45).

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

Tabela 2 - Comparação entre os paradigmas CG e CN

Tal como a CC, a CG garante a virtualização e elasticidade de recursos através do suporte físico de computadores interligados. Contudo, tipicamente, os computadores que da arquitetura CC estão localizados num único sítio tratando-se de um ambiente homogéneo, onde todas as máquinas têm o mesmo sistema operativo e o mesmo *hardware*. Quanto à CG, consiste num ambiente heterogéneo, pois os computadores podem executar diferentes sistemas operativos, podem diferir no *hardware*, e estão distribuídos através de uma *Local Area Network*, (**LAN**), ou *Wide Area Network*, (**WAN**) (16) (39) (46).

No paradigma CG, as máquinas são fracamente acopladas, de ambiente heterogéneo para a gestão de tarefas distribuídas. Quanto ao CC, trata-se de um sistema fortemente acoplado, transparecendo para o utilizador a ideia de um sistema único e homogéneo para a realização de trabalho centralizado (47).

Nos dois paradigmas CG e CN justifica-se destacar a redução dos custos de computação e a flexibilidade no uso de recursos disponibilizados por terceiros. Porém, a CN está focada no fornecimento de conjuntos de recursos, a partir de um aprovisionamento dinâmico, com a consequente redução de custos, em vez de permitir o acesso a todo o ambiente computacional, como acontece na CG (48).

2.2.4 Computação de Utilidade

Nas sociedades modernas, a maior parte dos serviços essenciais é disponibilizado de uma forma transparente. O abastecimento de água, eletricidade, gás e telefone, bens imprescindíveis no nosso dia-a-dia, têm esta característica. A sua exploração é feita com o modelo de pagamento baseado no uso, PAYG (49).

Os serviços disponibilizados no paradigma CN têm a característica da elasticidade, no caso, a resposta dinâmica do sistema à variabilidade das necessidades de recursos computacionais dos utilizadores, e o modelo de pagamento PAYG, tal como os serviços de utilidade pública. Esta analogia permite caracterizar a CN por Computação de Utilidade (*Utility Computing* - CU) (50).

2.2.5 Computação em Nuvem

A CN é um paradigma de disponibilização de recursos das TI, num ambiente virtual, escalável e multiutilizador. Tipicamente, cada *pool* de armazenamento tem a capacidade de alojar aplicações heterogêneas, com apropriados níveis de desempenho e segurança (26) (51).

Nos modelos tradicionais, as aplicações de negócio estão confinadas a uma infraestrutura particular. Tendencialmente, esta opção origina eficiência e flexibilidade baixas, além da dificuldade de manutenção dos complexos sistemas internos. Ao contrário, a CN proporciona a utilização das condições da infraestrutura mais adequadas à aplicação de negócio, caracterizando-se por escalabilidade, com parâmetros QoS e baixo custo (52).

A possibilidade de gestão de custos a partir do modelo PAYG é uma das características da CN. Por exemplo, é desnecessário pagar licença pelo uso integral do *software* pretendido. O acesso aos recursos da nuvem pode ser conseguido de qualquer lugar, utilizando um dispositivo de acesso à Internet. Assim, não é preciso um ambiente local de computação obrigando à manutenção da sua infraestrutura, nem a instalação e atualização de *software* nos computadores corporativos. Estas responsabilidades são transferidas para o fornecedor do serviço da nuvem (53) (54).

Fisicamente, a nuvem é constituída por servidores sustentados por uma infraestrutura orientada ao serviço, escalável e com componentes coesos e fracamente acoplados. O objetivo é favorecer a abstração de recursos computacionais, dando a ilusão de serem infinitos, com a possibilidade dos utilizadores usarem apenas a quantidade indispensável às suas necessidades de negócio. Deste modo, o paradigma introduz uma nova metodologia de negócio e de gestão de infraestrutura, onde o utilizador usa os recursos de *hardware*, *software* e rede, através de serviços disponibilizados na Internet (16) (55) (56) (57).

Sucintamente, dos benefícios associados à utilização dos serviços da nuvem destacam-se: a gestão centralizada, a redução de consumos energéticos e a diminuição dos custos inerentes à manutenção de infraestruturas tradicionais. A nuvem disponibiliza uma diversidade de serviços, onde a rapidez de acesso a recursos virtuais favorece a agilidade do negócio (33) (58) (59).

Deve ser feita uma análise cuidadosa das potenciais ameaças e desafios: dependência da estabilidade e da velocidade da Internet, desempenho, disponibilidade, segurança e eventuais restrições sobre a permissão de licenças. Fundamentalmente deve-se estar focado na produção de valor, na crescente satisfação do utilizador e na redução dos riscos existentes (50) (56) (60).

Por fim, destacam-se os cenários onde é possível aplicar o paradigma CN. O uso dos recursos da nuvem permite a utilização de serviços escaláveis, com possível redução significativa dos seus custos, dado ser desnecessário adquirir e gerir uma infraestrutura. Tipicamente, a escolha da CN, em detrimento dos outros paradigmas, é devida ao facto do nível de utilização de um determinado serviço se enquadrar num conjunto de cenários típicos (61) (62).

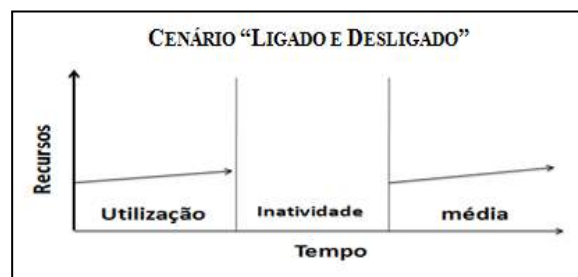


Figura 4 - Cenário "Ligado e Desligado"

Na Figura 4 apresenta-se o cenário "Ligado e Desligado", (*On and Off*) que corresponde a um ambiente onde ocorrem cargas temporárias de trabalho, com o consequente desperdício de recursos nos períodos inativos do serviço. No ambiente da nuvem, os recursos podem ser ligados ou desligados, de modo a libertar os que não são necessários nos momentos de ausência de trabalho. Em consequência, consegue-se a redução dos custos opex, (e.g. execução de rotinas) (63).

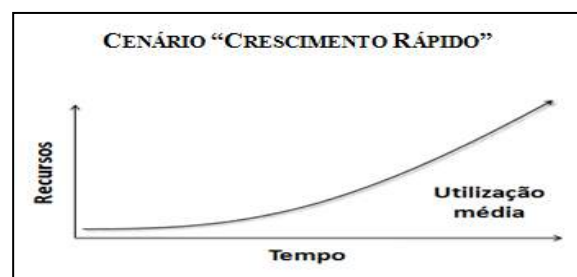


Figura 5 - Cenário "Crescimento Rápido"

A Figura 5 ilustra o cenário “Crescimento Rápido”, (*Growing Fast*) que traduz um aumento progressivo da necessidade de recursos, face a um número crescente de pedidos realizados ao longo do tempo. Tendo em conta a escalabilidade de recursos que é possível ter no ambiente CN, o utilizador pode aumentar o número de instâncias necessárias para responder a pedidos crescentes num curto espaço de tempo, sem penalização dos requisitos de negócio (64).



Figura 6 - Cenário "Carga Imprevista"

A Figura 6 corresponde ao cenário “Carga Imprevista”, (*Unpredictable Bursting*) que se enquadra num ambiente onde ocorrem picos de trabalho inesperados, com consequente impacto no desempenho e disponibilidade do serviço. O paradigma da CN permite que o utilizador dê respostas eficazes num curto espaço de tempo, com o aumento da quantidade de recursos, afim de não comprometer os requisitos de negócio, sem necessidade de investir em *hardware* local. Após os picos, o utilizador pode ajustar os recursos, garantindo uma melhor gestão dos custos opex do seu negócio (65).



Figura 7 - Cenário "Carga Sazonal"

Por último, na Figura 7 ilustra-se o cenário “Carga Sazonal”, (*Predictable Bursing*) ao qual corresponde um aumento de picos de trabalho, periodicamente, para os quais o aprovisionamento de recursos poderia conduzir ao seu desperdício. Recorrendo ao paradigma CN, não é necessária a aquisição antecipada de recursos, porque é possível

num curto intervalo de tempo, adaptar a quantidade dos recursos disponibilizados pelo fornecedor da nuvem às necessidades do seu negócio (61).

2.3 Análise SWOT

Nesta secção apresenta-se uma análise do modelo computacional estudado, a CN, destacando pontos fortes e fracos, mas também as oportunidades e ameaças, tendo em conta a sua evolução e integração, assim como as perspetivas de mercado. Na Tabela 3 apresenta-se a análise SWOT (25) (66).

<u>Pontos fortes</u> - Serviço escalável em curto prazo; - Redução dos custos: iniciais, de gestão de energia e de servidores subutilizados; - Configurações predefinidas de servidores e de máquinas virtuais, com controlo dos recursos adquiridos; - O acesso e a gestão do sistema organizacional, a partir de API, são disponibilizados pelos fornecedores, através de dispositivos com inferiores capacidades computacionais.	<u>Pontos fracos</u> - Os consumidores podem perder o controlo dos dados alojados na nuvem; - Os fornecedores não garantem total qualidade e disponibilidade de serviço, durante todo o ano, o que pode ser insuficiente para as empresas; - Possibilidade de falha nos serviços disponibilizados pelos fornecedores da nuvem.
<u>Oportunidades</u> - Sem necessidade dos tradicionais gastos iniciais, as empresas têm facilitada a criação e a expansão de negócios; - Criação de <i>mashups</i>, páginas Web que combinam diversos serviços externos, originando novos serviços e novas oportunidades de negócio; - Uso mais racional da energia e dos recursos naturais.	<u>Ameaças</u> - Possibilidade de falência dos fornecedores da nuvem; - Preocupação com a segurança, a confiabilidade e o desempenho dos serviços da nuvem; - Regulamentação variável e sujeita a eventuais alterações, conforme as decisões nacionais, no que respeita à privacidade, aos requisitos de auditoria e de localização dos dados.

Tabela 3 - Análise SWOT do paradigma CN

Considerando esta análise conclui-se que antes da escolha de uma plataforma de CN, é necessário analisar e comparar os custos da aquisição, gestão e de manutenção de uma infraestrutura local com os custos associados à utilização dos recursos de terceiros. Deve-se prever o tempo de duração do negócio, de modo a diminuir os custos capex e opex, ajustando as características dos componentes arquiteturais aos resultados pretendidos.

3 TIPOS DE CARREGAMENTO DA NUVEM

Neste capítulo são abordados os três tipos de carregamento associados à CN. Inicia-se com uma introdução e posterior indicação de características e vantagens de utilização de cada tipo de carregamento.

3.1 Introdução

O diagrama da Figura 8 ilustra os três tipos de carregamento da nuvem, designadamente: privado, público ou híbrido.

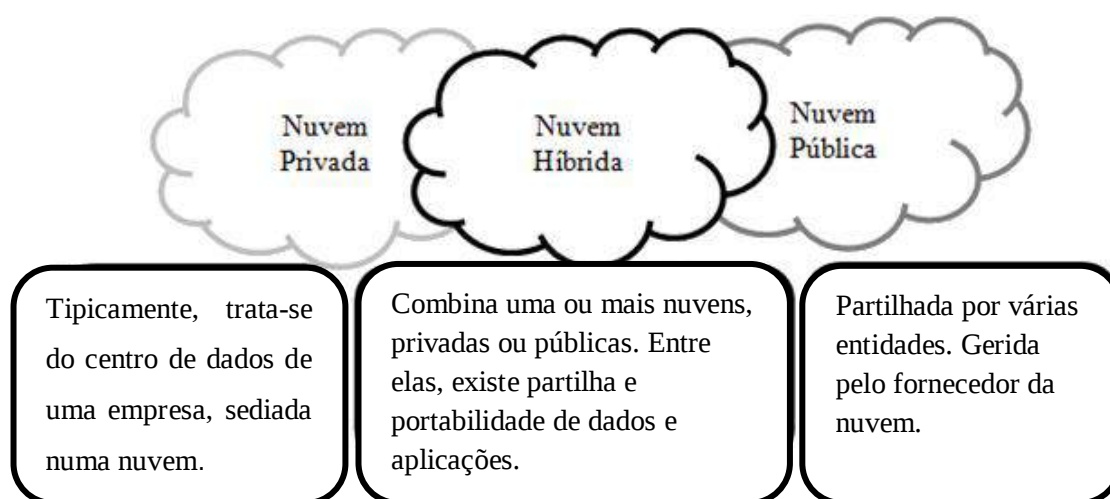


Figura 8 - Tipos de carregamento da nuvem

A nuvem privada destina-se a empresas, que pretendem usar as suas tecnologias e nela alojar as suas aplicações multiutilizador, juntamente com as componentes de dados e de processamento (67) (68).

O conceito de nuvem pública alude à partilha de serviços e da infraestrutura física da nuvem, num ambiente multiutilizador. Os recursos são disponibilizados dinamicamente e numa perspetiva *self-service* a partir da Internet. Os riscos associados à colocação de recursos críticos na nuvem pública estão associados aos mecanismos de segurança e de controlo de acesso aos dados, visto estes serem partilhados pelas diversas entidades presentes na nuvem (34).

A nuvem híbrida consiste numa mescla dos outros dois tipos de carregamento, nuvem privada e pública. A hibridez proporciona a economia de custos e o acesso a uma infraestrutura escalável com mecanismos de segurança. Pela agregação das vantagens dos dois modelos, privado e público, esta metodologia é a opção mais interessante para a maioria das empresas. No entanto, as nuvens híbridas introduzem uma complexidade adicional, a distribuição de aplicações pelos dois modelos (69).

A virtualização possibilita o isolamento de recursos computacionais. Em consequência, os utilizadores finais têm facilidade de adaptação e agilidade, na escolha do modelo de implementação da nuvem mais apropriado. Antes de decidir, eles terão de ponderar a estratégia mais adequada às suas necessidades empresariais (23) (33).

3.2 Nuvem Privada

O conceito de nuvem privada aplica-se à infraestrutura de uma organização, alojada num fornecedor e que não está disponível ao público em geral. A capacidade computacional (e.g. processamento, armazenamento) e a consequente elasticidade dos recursos disponibilizados dependem do investimento realizado (10).

Na construção de uma nuvem privada devem ser satisfeitos aspetos de racionalização de recursos, de otimização da arquitetura e de implementação de serviços, a partilhar pelos utilizadores com autenticação e autorização válidas (14).

A racionalização determina o modo de utilização dos serviços das TI e a redução dos recursos, que não sejam usados procurando padronizar e homogeneizar o ambiente computacional da nuvem, a fim de facilitar a sua gestão, melhorar a sua agilidade e reduzir os custos (51).

A arquitetura da nuvem privada, definida pelos serviços pretendidos, deve ter características, nomeadamente: virtualização e a automação. A virtualização, pela abstração de recursos, transforma o típico modelo servidor/aplicação num modelo multiutilizador, de acordo com as solicitações dos utilizadores. A automação, face à intervenção humana ser residual, agiliza a infraestrutura. Em consequência, obtém-se a redução do tempo de resposta, face às constantes mudanças induzidas pelo utilizador (70).

Uma das vantagens da nuvem privada é a partilha de serviços pelos diversos departamentos das TI, conduzindo à redução de custos. A coesão e o fraco acoplamento entre os diferentes serviços são requisitos para a combinação de múltiplas aplicações, o que proporciona o melhor aproveitamento dos recursos e a simplificação das normas de segurança. A consolidação de serviços leva à redução da aquisição de servidores físicos, com a consequente redução do consumo de energia (71).

Na Tabela 4 são apresentadas as vantagens inerentes à utilização de uma nuvem privada (56) (72) (73).

Aspeto	Descrição
Redução de custos	Numa nuvem privada, a consolidação dos recursos partilhados permite baixos custos e um melhor aproveitamento da infraestrutura. Através da agregação de bases de dados num ambiente multiutilizador, a componente física tem menores gastos energéticos. Políticas de automação reduzem os custos operacionais.
Redução de complexidade	Pela racionalização, normalização e consolidação de recursos, os ambientes podem reduzir a complexidade de configuração e serviços. Através da normalização, os departamentos das TI podem definir configurações por omissão, proporcionando a escalabilidade nos seus ambientes, com componentes modulares.
Melhoramento dos parâmetros de qualidade (QoS)	Com uma nuvem privada, a camada de dados e outros recursos são monitorizados e geridos através de interfaces partilhadas. Estes parâmetros possibilitam a unificação e a segurança da infraestrutura, de acordo com um processo normalizado.
Departamentos TI ágeis e flexíveis	A procura de soluções ágeis e flexíveis é um objetivo dos departamentos das TI. A utilização de uma nuvem privada permite o carregamento da infraestrutura. A normalização de componentes e configurações, juntamente com uma gestão facilitada, permitem ter uma elasticidade adaptada às necessidades dinâmicas de negócio.

Tabela 4 - Vantagens da nuvem privada

As nuvens privadas são uma forma eficiente para disponibilizar serviços de bases de dados, pois direcionam os departamentos das TI para a consolidação da informação em componentes partilhados. O carregamento da camada de dados numa nuvem privada é benéfico para o utilizador. Este, face ao fornecimento escalável de recursos de acordo com os seus pedidos, beneficia no custo, na qualidade de serviço disponibilizado e na agilidade (23).

Interessa salientar que, tendo uma visibilidade menor, esta nuvem possibilita a existência de um controlo mais centralizado e a definição de políticas de segurança mais rígidas entre os utilizadores.

A nuvem privada é uma solução, destinada a uma única entidade, para resolver o problema do alojamento de recursos num ambiente virtual. Como o uso dos recursos é exclusivo, sem partilha com outras organizações, torna-se mais fácil conseguir segurança e privacidade. O risco é inferior àquele que é inerente a modelos multiutilizador (56).

Usualmente aloja-se o ambiente de uma nuvem privada numa infraestrutura com recursos dedicados. Pelos mecanismos de virtualização, os utilizadores usufruem de máquinas virtuais, dirigidas para as suas necessidades de negócio, e dos benefícios de elasticidade, associados ao modelo de CN (74).

3.3 Nuvem Pública

A nuvem pública consiste na coordenação e combinação, de serviços de plataformas externas da nuvem com a infraestrutura interna da empresa. A escalabilidade é um dos benefícios associados a este tipo de modelo. Pela agregação de diversos centros de dados, com as suas capacidades de processamento, memória e armazenamento, os utilizadores têm a ilusão de possuir recursos ilimitados. Estes podem ser geridos uniformemente, com base nas mudanças dos requisitos comerciais (72).

Se, eventualmente, houver necessidade de mais recursos computacionais, a nuvem pública pode fornecê-los, sem ser preciso recorrer à tradicional compra de *hardware*. Uma outra vantagem da nuvem pública é o modelo de pagamento. A possibilidade de usar recursos de terceiros durante o tempo pretendido, evitando a saturação ou acumulação de recursos inutilizados, permite uma melhor gestão dos custos associados (75).

A partilha, por vários utilizadores, da infraestrutura e dos recursos da nuvem pública torna necessária a implementação de mecanismos de segurança (e.g. criptografia). Normas de funcionamento têm também de ser definidas, são precisas para estabelecer o

modo como os vários componentes interagem entre si, de forma a originar um ambiente coerente e com níveis de desempenho aceitáveis pelos utilizadores (76).

3.4 Nuvem Híbrida

A nuvem híbrida visa a gestão dinâmica do ambiente da nuvem, a fim de acelerar a configuração; a flexibilidade e a satisfação rápida das necessidades do negócio.

Na Tabela 5 apresentam-se os aspetos a considerar na utilização de uma nuvem híbrida (69) (73).

Aspeto	Descrição
Segurança	O aumento do número de recursos computacionais, pelo agrupamento de nuvens privadas e públicas, amplia o perímetro para ataques contra a sua integridade.
Gestão do controlo de acesso	A segurança da organização híbrida exige a presença de uma entidade corporativa, cuja responsabilidade consiste no controlo e gestão de acessos à nuvem. A fim de conseguir a uniformidade dos diversos ambientes da nuvem híbrida, esta entidade deve garantir o cumprimento do protocolo <i>Authentication, Authorization and Accounting, (AAA)</i> .
Migração de dados	A existência de uma entidade corporativa permite fazer de forma mais controlada a migração de dados num modelo híbrido. Contudo, ao fazer a migração de dados, as preocupações relativas aos controlos de privacidade e integridade continuam presentes.
Redução de custos	A utilização de recursos de terceiros, tornando desnecessária a aquisição de recursos próprios, permite a redução dos custos. Com o modelo híbrido, as organizações otimizam os custos da infraestrutura nas diferentes fases do seu negócio.
Agilidade de negócio	A utilização de nuvens híbridas conjuga o uso de controlos de segurança, implementados em nuvens privadas, com a escalabilidade existente em nuvens públicas. Em consequência: melhora a agilidade organizacional, pela capacidade de adaptar os recursos às necessidades de negócio; reduz as preocupações inerentes à gestão da infraestrutura, pois os ambientes ficam sob responsabilidade do fornecedor da nuvem híbrida, contrariamente ao que acontece nos ambientes tradicionais.

Tabela 5 - Aspetos a considerar na nuvem híbrida

Este tipo de nuvem acumula as vantagens dos outros dois modelos anteriormente descritos. Possibilita a ampliação dos recursos computacionais da nuvem privada, tal como acontece na nuvem pública, mantendo os níveis de serviço, mesmo nos picos de solicitação de recursos. Consoante a sensibilidade dos dados, o administrador da nuvem

híbrida pode optar pela nuvem privada ou pública, tirando partido das vantagens de uma e de outra (66).

Antes de emigrar os seus dados e aplicações das TI para uma nuvem híbrida, as empresas devem estudar o impacto das regulamentações impostas e avaliar as políticas e procedimentos associados à deslocação. Na análise da mudança, deve-se recorrer a testes, mecanismos de auditoria, e outros métodos, a fim de obter a informação que possibilite a escolha do fornecedor da nuvem híbrida (69).

Em suma, as nuvens híbridas são uma combinação de nuvens privadas ou públicas, que possibilitam a troca de dados e, possivelmente, a compatibilidade e portabilidade de aplicações provenientes de diferentes nuvens. Usualmente, estas nuvens são usadas por empresas que, face à rutura da sua infraestrutura local ou à pretensão de uma implementação mais rápida dos seus processos de negócio, desejam colocar os seus dados em nuvens públicas. Os dados mais sensíveis podem permanecer numa nuvem privada, ou até mesmo no tradicional sistema de ficheiros, garantindo a sua partilha apenas pelos utilizadores autorizados (77).

4 NÍVEIS DE SERVIÇO DA NUVEM

Neste capítulo descrevem-se os três tipos de níveis de serviço da CN indicando o tipo de recursos disponibilizados e as vantagens da sua utilização.

4.1 Introdução

O paradigma CN tem associados três níveis de serviços, designadamente: aplicação como um serviço (*Software as a Service* - **SaaS**), plataforma como um serviço (*Platform as a Service* - **PaaS**) e infraestrutura como um serviço (*Infrastructure as a Service* - **IaaS**). Tipicamente, a disponibilização de serviços é feita a partir de uma interface normalizada, estabelecida numa arquitetura orientada ao serviço, recorrendo a interfaces descritivas (e.g. REST). Em alternativa pode recorrer-se à subscrição do serviço. Neste caso, o modelo de pagamento permite que o utilizador pague apenas pelos recursos contratados durante o tempo estabelecido (23) (48).

A Figura 9 apresenta os referidos serviços e as respetivas camadas, destacando as responsabilidades do cliente e do fornecedor do serviço (50).

CAMADAS	ON PREMISE	IAAS	PAAS	SAAS
Aplicações	Cliente	Cliente	Cliente	Fornecedor
Dados	Cliente	Cliente	Cliente	Fornecedor
Runtime	Cliente	Cliente	Fornecedor	Fornecedor
Middleware	Cliente	Cliente	Fornecedor	Fornecedor
Sistema operativo	Cliente	Cliente	Fornecedor	Fornecedor
Virtualização	Cliente	Fornecedor	Fornecedor	Fornecedor
Servidores	Cliente	Fornecedor	Fornecedor	Fornecedor
Armazenamento	Cliente	Fornecedor	Fornecedor	Fornecedor
Rede	Cliente	Fornecedor	Fornecedor	Fornecedor

Figura 9 - Níveis de serviço da nuvem

Tradicionalmente, o cliente é o responsável por todas as camadas da sua infraestrutura de TI, desde a rede até às aplicações. A contratação de níveis de serviço da nuvem, IaaS, PaaS e SaaS, por esta ordem, implica uma transferência de responsabilidade de gestão do cliente para o fornecedor (68).

Genericamente, o serviço SaaS fornece uma vasta gama de recursos integrados, de um modo extensível e com garantias de segurança dadas pelo fornecedor. O nível PaaS oferece menos recursos integrados, pois visa disponibilizar recursos para desenvolvimento e/ou instalação de aplicações diretamente na plataforma do fornecedor. Por fim, o nível IaaS aposta na extensibilidade de recursos de computação da infraestrutura, permitindo alojamento de sistemas operativos, aplicações e conteúdos geridos pelo cliente do serviço (78) (79).

4.2 *Software as a Service*

O conceito de SaaS representa o culminar de oportunidade da nuvem para os clientes empresariais. As aplicações são disponibilizadas como um serviço no modelo de pagamento PAYG. O consumidor apenas paga pelo que utilizou e/ou pelo tempo de uso dos recursos. Esta modalidade proporciona a redução de complexidade inerente à configuração e manutenção da infraestrutura, (e.g. sistema operativo, *hardware*), pois estas atividades ficam à responsabilidade do fornecedor. O utilizador apenas necessita de se preocupar com o acesso ao serviço, obtido através de um dispositivo com conexão à rede e de uma interface que lhe facilite o acesso (80).

No modelo SaaS são disponibilizados os serviços de mais alto nível alojados na nuvem, aplicações completas, com uma redução de custos conseguida pela dispensa/inclusão da aquisição de licença de *software* ou mesmo pela compra do *hardware* para o executar (81).

Para efetuar a gestão das aplicações da nuvem, o utilizador tem de as controlar remotamente, com funcionalidades como a escalabilidade, a medição dos níveis de desempenho e a integração com serviços provenientes de outros fornecedores (82).

As vantagens do modelo SaaS são a gestão automática do serviço, a compatibilidade, normalmente garantida pela existência de uma única versão para os diversos

utilizadores e a sua característica de *thin client* onde o controlo do serviço e da infraestrutura ficam a cargo do fornecedor. O modelo beneficia também das soluções de negócio, nomeadamente a gestão de relações com o utilizador, CRM, que possibilita a reserva de recursos consoante as necessidades, automatizando as funções de contato com o cliente, bem como as de planeamento de recursos empresariais (*Enterprise Resource Planning* - **ERP**), integrando os dados e processos dos diversos departamentos da organização do cliente num único sistema, de modo a influenciar as tomadas de decisão do negócio e, consequentemente, reduzir o uso de interfaces manuais, os custos, o tempo de resposta no mercado e otimizando o fluxo e a qualidade da informação da organização (83) (84).

Para não comprometer o seu negócio, o cliente deve ter presente um conjunto de aspetos de uma dada solução SaaS, nomeadamente (42) (85):

- a. Realizar regularmente *backups* dos dados de preferência com o formato independente do fornecedor do SaaS;
- b. Compreender quais as ferramentas e/ou tecnologias a serem utilizadas em cada momento do negócio;
- c. Assegurar a gestão eficaz dos antigos e atuais fornecedores de serviços;
- d. Elaborar relatórios de gestão de interfaces e da sua integração entre os diversos ambientes computacionais utilizados;
- e. Garantir a fase de testes, utilizando um protótipo, de modo a avaliar o desenvolvimento das aplicações, a resposta da infraestrutura face à necessidade de escalabilidade, simular casos de falha e analisar as suas políticas de segurança antes da migração do modelo de negócio para a nuvem.

Sendo o modelo SaaS uma nova tecnologia com vantagens sobre outras anteriores, também tem inconvenientes, especificamente, os seguintes (86) (87):

- a. Existência de atualizações pouco frequentes, dado a utilização da aplicação SaaS por diversos utilizadores tornar complexa a sua realização;
- b. A eventual necessidade do utilizador personalizar o serviço acarreta custos adicionais; em consequência, antes da migração, torna-se imprescindível fazer a análise detalhada dos custos, ponderando o tempo necessário para o retorno do investimento;

- c. Na instalação de aplicações, a fase do seu carregamento na nuvem aumenta se elas forem complexas, podendo comprometer o sucesso do negócio;
- d. O acesso às aplicações instaladas na nuvem realiza-se pela Internet, pelo que a dependência da qualidade da conexão constitui um potencial risco.

Essencialmente, o SaaS consiste no alojamento de aplicações na nuvem, acessíveis através da Internet e fornecidas por terceiros, que o utilizador pode usar recorrendo à infraestrutura do serviço nela existente.

4.3 *Platform as a Service*

O conceito de plataforma como um serviço, PaaS, consiste numa camada de *software* programável para o desenvolvimento e instalação (*deploy*) de serviços de mais alto nível SaaS (33).

O objetivo do PaaS consiste em facilitar o desenvolvimento de aplicações, destinadas aos utilizadores da nuvem, através de uma plataforma que agiliza o seu processo de integração, independentemente da localização geográfica, sem preocupações de gestão e de atribuição de recursos. A partir da plataforma são disponibilizados recursos, (e.g. sistemas operativos), para os utilizadores poderem disponibilizar as suas aplicações SaaS (67).

Adicionalmente, o modelo PaaS disponibiliza, de forma transparente, políticas de segurança, gestão e escalabilidade. Face à estruturação e lógica da plataforma, também é evitado o gasto adicional de tempo permitindo colocar mais depressa o produto no mercado, com o conseqüente aumento do retorno de investimento nas aplicações (51) (83).

No que respeita ao controlo e/ou administração da plataforma, salienta-se que o cliente só controla as suas aplicações e, eventualmente, de forma limitada, a infraestrutura de alojamento. O cliente não gere a infraestrutura e os recursos inerentes, nomeadamente, a rede, o *hardware* e o sistema operativo (68) (86).

O modelo PaaS permite que o cliente, usando as tecnologias e ferramentas disponibilizadas, aloje as suas aplicações na infraestrutura do fornecedor que também

tem a possibilidade de enriquecer o conjunto de serviços com o usufruto dos de terceiros, aumentando previsivelmente a qualidade do seu negócio (42).

4.4 *Infrastructure as a Service*

O IaaS é o nível de serviço mais intrínseco de CN e o que suporta os outros, o PaaS e o SaaS. Possibilita, aos interessados, a utilização dos recursos, servidores, rede e dispositivos de armazenamento, entre outros recursos de computação existentes. A infraestrutura é composta por recursos computacionais destinados à sustentação de ambiente escaláveis, com suporte para camadas *middleware* e respectivas aplicações.

Tipicamente, o fornecedor do serviço IaaS também o administra gerindo as interfaces, a fim de permitir a interação com os demais equipamentos e a eventual adição de outros, de um modo transparente, para as camadas de serviço acima. Em consequência, excluindo a gestão da infraestrutura, o cliente pode ter o controlo do *software*, desde o sistema operativo às aplicações, assim como dos dispositivos de armazenamento e possivelmente dos componentes de rede (42) (68) (88).

Como nos níveis de serviço anteriormente referidos, PaaS e SaaS, antes da aquisição do serviço IaaS, vários aspetos devem ser ponderados, nomeadamente (82) (83) (85):

- a. Dento do possível, garantir a interdependência entre a máquina virtual e a aplicação alojada na mesma, de forma a minimizar o nível de esforço na eventual necessidade de mudar de fornecedor;
- b. Regular realização de *backups*, fora do contexto e do formato existentes na máquina virtual;
- c. Antes da migração do processo de negócio para a máquina virtual de destino, por um mecanismo de teste, é conveniente analisar e compreender os requisitos garantidos pela infraestrutura.

O termo IaaS está associado a uma infraestrutura computacional escalável, suportada por metodologias de virtualização de recursos de computação. Resumidamente, o fornecedor gere a camada física, enquanto, a partir do sistema operativo, os clientes interagem com as outras camadas. Tal como nos restantes serviços, a forma de

pagamento obedece à metodologia PAYG, onde o cliente paga consoante o tempo, o tipo e/ou quantidade de recursos utilizados (6) (67).

No presente capítulo faz-se uma descrição e um estudo comparativo dos serviços CN disponíveis no mercado.

5.1 *Google App Engine*

A tecnologia *Google App Engine* (**GAE**) enquadra-se na utilização da infraestrutura do *Google* para a disponibilização de serviços, na forma de SaaS, como é o caso do correio eletrónico *GMAIL*, da conversação *Google Talk* e dos editores, *Google Docs* (13).

Esta tecnologia faculta a replicação dos centros de dados, de modo a contornar os problemas de disponibilidade, em casos de manutenção ou de interrupção de uma determinada instância. A sua infraestrutura escalável permite a adaptação às variações de quantidade do tráfego e de armazenamento de dados, e também a execução de aplicações desenvolvidas em linguagens de programação, tais como: *Python* e *Java*. No entanto, dependendo do tipo de aplicação, a plataforma que a aloja tem um conjunto restrito de interfaces disponíveis, (*Application Programming Interface* – **API**), como a *URLFetch* e a *Datastore*. Por exemplo, uma aplicação alojada na plataforma *App Engine* não pode escrever diretamente no sistema de ficheiros, pois a operação apenas é permitida a partir da *API Datastore* (89).

Os ambientes de execução disponibilizados permitem a abstração do sistema operativo, e também das operações de processamento, encaminhamento e balanceamento de carga (90).

A adição de funcionalidades desenvolvidas por ambientes computacionais corporativos permite, ao nível IaaS, a existência do serviço de armazenamento de dados, facilitador da resolução de problemas das TI, como segurança e disponibilidade. Neste âmbito, salientam-se dois tipos de armazenamento, *blobstore* e *datastore*. O primeiro, onde o armazenamento se faz em objetos designados por *Binary Large Objects* (**BLOB**), está direcionado para dados multimédia, pois os dados armazenados podem ser de dimensão superior à permitida no *datastore*. O armazenamento realiza-se a partir do *upload* de um ficheiro enviado por um pedido HTTP, que tem como resposta uma referência para o

respetivo BLOB. O segundo tipo de armazenamento de dados, o *datastore*, tem um formato par chave-valor, onde as entidades armazenadas não têm uma estrutura pré-definida, *schemaless*. O armazenamento é realizado em *entity groups* com suporte transacional. Na necessidade de utilização de base de dados relacionais, estas são suportadas pelo serviço, denominado por *Hosted SQL*, que está associado ao tipo *datastore* e possibilita a criação de *views* (89).

A plataforma do *Google* suporta a execução assíncrona através de filas de mensagens, identificadas por um *Uniform Resource Locator (URL)*. O despacho é realizado de acordo com a política *First In First Out (FIFO)* e gerido pela entidade denominada por *cron jobs*. Este serviço permite definir tempos, ou intervalos periódicos de execução de tarefas, que são despoletadas pelo serviço *AppEngine Cron Service*, como por exemplo, a atualização periódica da *cache* e a emissão de relatórios. O procedimento do serviço *cron jobs* passa por invocar o URL, incluído no respetivo pedido HTTP. A execução pode demorar algum tempo, estando dependente da existência de outros pedidos em curso (91) (92).

Para facilitar o teste das aplicações, a plataforma permite testar num servidor local incluído no *Software Development Kit, (SDK)* (93). Este servidor simula o tipo de armazenamento *datastore* e outros serviços disponíveis na plataforma. Após a fase de teste, realiza-se o alojamento da aplicação, sem acessos públicos, de modo a testar a sua viabilidade. Posteriormente ao alojamento e à definição de restrições de acesso, a gestão pode ser feita a partir de uma página, onde é verificado o processamento, o volume de dados e o desempenho da aplicação desenvolvida (94).

5.2 Amazon Web Services

A *Amazon* disponibiliza uma plataforma constituída por diversos serviços conhecidos abreviadamente por *AWS* (95). Apesar do fraco acoplamento entre os mesmos é garantida a comunicação independente do local geográfico do alojamento, com variação de custos, dependendo dos serviços e do número de instâncias utilizadas.

As máquinas virtuais disponibilizadas, designadas por *Amazon Machine Instance (AMI)* e a componente de *software* desde aplicações e bibliotecas ficam disponíveis

para os potenciais clientes. Outra opção consiste em utilizar *templates* de máquinas virtuais, igualmente disponibilizadas, que combinam *software* com diferentes tecnologias. Ambas as soluções apresentam como características a baixa latência e níveis de redundância, de modo a dar resposta a eventuais falhas das instâncias disponibilizadas. A título de exemplo, destacam-se os serviços S3, direcionado para o armazenamento de dados e EC2 vocacionado para o aluguer de servidores virtuais escaláveis para a execução de aplicações (67) (96).

Ao nível do armazenamento, a plataforma *Amazon* assegura o formato binário e elementos estruturais de dados. O armazenamento binário é garantido pelos serviços S3, com acessos privados ou públicos, e o EBS semelhante a um sistema de ficheiros sendo interpretado como um disco rígido pelo sistema operativo (17).

Relativamente à comunicação salienta-se o modo assíncrono, disponibilizado pelo serviço *Simple Queue Service*, (**SQS**), que contém uma estrutura de filas ou fluxos de trabalho, que asseguram a comunicação entre aplicações. O serviço SQS possibilita a troca de dados e execução de tarefas, com a garantia de inexistência de perda de mensagens e do seu armazenamento, enquanto aguardam o momento de serem executadas. Quanto à estrutura da mensagem, não existe um formato pré-estabelecido e o tamanho ocupado pelo texto pode ascender até aos 64 KB (23).

A interação com uma determinada fila pode ser realizada por múltiplos leitores e/ou escritores, sem a existência de uma coreografia entre eles (97).

A escalabilidade é conseguida através da combinação dos serviços *CloudWatch* (**CW**), e *Elastic Load Balancing* (**ELB**). O serviço CN permite visualizar e analisar métricas, obter estatísticas e também, com base nos dados da métrica, gerar alarmes, a fim de notificar ou realizar alterações consoante as regras definidas pelo utilizador (98). O serviço ELB garante a disponibilidade e escalabilidade de uma aplicação através de metodologias de redundância e de balanceamento de carga entre componentes EC2 (99) (100).

Em suma, através da implementação de serviços e de metodologias de virtualização, a *Amazon* disponibiliza um ambiente otimizado e organizado que, pelo uso de uma máquina virtual privada, a AMI, permite o desenvolvimento e execução de aplicações. A AMI armazena dados e bibliotecas, possibilitando que a administração seja realizada

pelo utilizador, mesmo na escolha do intervalo de endereços IP e na configuração de uma rede virtual privada (**VPN** – *Virtual Private Network*) (13) (101).

5.3 Joyent

A infraestrutura de serviços *Joyent* disponibiliza máquinas virtuais e aplicações na forma de SaaS, com a possibilidade de alojar aplicações de terceiros no sistema operativo *Solaris* (102). Este ambiente apresenta configurações destinadas à Web, com *software* pré-instalado, tal como: *Apache*, *MySQL*, *PHP*, *Ruby* e *Java*. O *Solaris* disponibiliza funcionalidades indispensáveis para sistemas distribuídos, tal como, balanceamento de carga e escalabilidade vertical dos núcleos de CPU, onde se pode escalar o processamento até ao máximo suportado pelo ambiente físico sem interferir com as aplicações alojadas. As interfaces de programação são direccionadas para a Web, incluindo as de administração, com acessos ao servidor virtual a partir de mecanismos de criptografia, como *Secure Shell (SSH)*, possibilita a negociação de parâmetros do nível de serviço (i.e. responsabilidades das partes envolvidas) e naturalmente o pagamento baseado em PAYG (103) (104).

5.4 Eucalyptus

A infraestrutura *Eucalyptus* permite a utilização de aplicações desenvolvidas através de código aberto, direccionadas para o ambiente CN, e disponibiliza uma interface compatível com as plataformas EC2, S3 e EBS, a partir do modelo IaaS. A sua arquitetura é flexível e modular, com uma conceção hierárquica, que facilita a manutenção e o desenvolvimento de aplicações (23).

O sistema permite que os utilizadores realizem a gestão das suas instâncias de máquinas virtuais. Estas são suportadas por um conjunto de recursos físicos necessários para garantir a execução das aplicações alojadas, através de um emulador EC2 SOAP, permitindo, deste modo, a interoperabilidade com a infraestrutura *Amazon* (53).

Os contratos disponibilizam as suas funcionalidades a partir de uma API, cumprindo os requisitos da linguagem *Web Services Description Language (WSDL)*, a partir da qual

se definem as estruturas de dados de entrada e de saída, assim como as operações do serviço (105).

Para a segurança, é utilizado o mecanismo *WSSecurity*, que define um conjunto de políticas para a comunicação segura (71).

5.5 *Windows Azure*

A plataforma WA permite o alojamento de aplicações e de serviços nos centros de dados da *Microsoft* (106). A plataforma subjacente operacionaliza um conjunto de tecnologias NET (C#, VB.Net, ASP.NET) e também *Java* e *Ruby*. Os protocolos SOAP (107) e REST (108) estão incluídos, assim como outras linguagens da plataforma .NET e de terceiros (10).

Para facilitar a integração dos serviços desenvolvidos pelos utilizadores, a plataforma disponibiliza o *IDE Visual Studio* e *SQL Server Management Studio*, juntamente com as tecnologias ASP.NET, *Windows Communication Foundation (WCF)* (para utilização de filas de mensagens) e *Fast Common Gateway Interface (FastCGI)* (em PHP ou Python) (93) (109).

Quanto ao isolamento entre as aplicações, cada uma é executada numa máquina virtual, onde é possível gerir os recursos, o espaço ocupado bem como a zona geográfica para o seu alojamento. Esta última possibilidade permite que o utilizador, de acordo com as políticas de um determinado país, possa optar pelas que mais se adequam aos seus requisitos. O pagamento é realizado de acordo com os recursos utilizados (e.g. componentes de armazenamento e processamento) por unidade de tempo considerando (110).

A gestão dos servidores é realizada pelo serviço *Windows Azure Fabric Controller (WAFC)*, que se responsabiliza pela instalação e atualização de *software* necessários, garantindo o seu funcionamento. Este serviço é igualmente responsável pela escalabilidade, disponibilidade, gestão de recursos de memória e balanceamento de carga. O serviço *.NET Service Bus* regista e interliga diversas aplicações. Por fim, o serviço *.NET Access Control* é responsável por identificar os fornecedores com diretórios corporativos (111).

O armazenamento pode ser realizado de quatro modos distintos, designadamente: a partir do elemento (i) BLOB, destinado a armazenamento de grandes quantidades de dados, o modo (ii) *Table*, preparado para guardar informação estruturada, sem um *schema* pré-definido, a (iii) *queue*, para o armazenamento e entrega de mensagens, favorecendo o fraco acoplamento e o fluxo de trabalho escalável entre os papéis (*roles*) existentes na aplicação, e o modo (iv) *drives*, que consiste em volumes *New Technology File System (NTFS)* a serem utilizados por aplicações WA. A componente de base de dados relacional é igualmente suportada, incluindo mecanismos como a redundância de informação, automatização de responsabilidades administrativas e a possibilidade de utilização de estruturas como índices, vistas, procedimentos armazenados e gatilhos (112).

5.6 Comparação de Fornecedores

Após a análise de alguns fornecedores CN apresenta-se na Tabela 6, uma comparação dos serviços por eles prestados (23) (113) (114) (115).

	GAE	WA	AWS (EC2, S3)	Joyent	Eucalyptus
Autenticação	<i>Google Accounts</i>	<i>App Fabric</i>	<i>AWS Identity e Access Management</i>	<i>Basic Authentication HTTP</i>	Garante o protocolo AAA a partir de um sistema controlo de acesso
Comunicação assíncrona	<i>Task and Scheduled tasks</i>	Fila de mensagens	SQS	-	-
Escalabilidade	Modelo distribuído	Adição dinâmica de instâncias		Vertical	-
Tipos de armazenamento	<i>BlobStore e DataStore</i>	BLOB, <i>Table</i> , <i>Queue</i> , <i>Drives</i> e base de dados SQL	S3, EBS e <i>Amazon Simple DB</i>	<i>Smart-Data-Center</i>	<i>Dynamic block volume</i>
Suporte para SQL	<i>Hosted SQL</i>	<i>Azure SQL</i>	<i>Relation Database Service</i> em EC2	-	Imagem a alojar no servidor <i>MySQL</i>
Tecnologias	API restritas	.NET e de terceiros	Sem restrições	<i>Apache, MySQL, PHP, Ruby e Java</i>	Código aberto e compatível com EC2 e S3
Nível de service	IaaS	PaaS	IaaS	SaaS, PaaS e IaaS	IaaS
Modelo de pagamento	Gratuito e PAYG		PAYG	PAYG	Gratuito
Infraestrutura	Própria	Centro de dados <i>Microsoft</i>	Própria	Própria	Para ser alojada noutra plataforma
Zonas geográficas	-	América do norte, Ásia e Europa	E.U.A	E.U.A	-

Tabela 6 - Comparação de fornecedores CN

6 A PLATAFORMA *WINDOWS AZURE*

A plataforma WA é aberta e flexível, possibilitando que, a partir dos centros de dados da *Microsoft*, os utilizadores possam construir, alojar e gerir as suas aplicações. Estas podem ser desenvolvidas a partir de linguagens, ferramentas e/ou ambientes de desenvolvimento da *Microsoft* ou de terceiros. Estão disponíveis bibliotecas em diversas linguagens, a partir de uma licença de código aberto, alojadas num serviço de *Web Hosting* partilhado destinado a projetos que utilizam o sistema de controlo de versões designado por GIT (116).

No que respeita aos contratos de níveis de serviço, (SLA), a plataforma WA garante quase total disponibilidade, (99,9% dos pedidos realizados aos vários componentes que a constituem), assim como mecanismos de redundância, em caso de falha de alguma instância do serviço (117). A plataforma inclui serviços destinados à resolução de problemas e ao melhoramento da usabilidade e do desempenho. Caso ocorram falhas de *hardware* existem mecanismos de balanceamento de carga, de rede e de recuperação. De modo automático, e em poucos minutos, a plataforma garante a escalabilidade das aplicações. Sendo assim, é possível gerir os recursos de acordo com as necessidades, evitando a existência de recursos subutilizados ou a sua ausência (10) (25).

A existência de centros de dados da *Microsoft* em três continentes, (América, Europa e Ásia), permite a implementação e o alojamento das aplicações mais perto dos clientes, assim como a implementação de mecanismos de redundância de dados em diferentes centros (118).

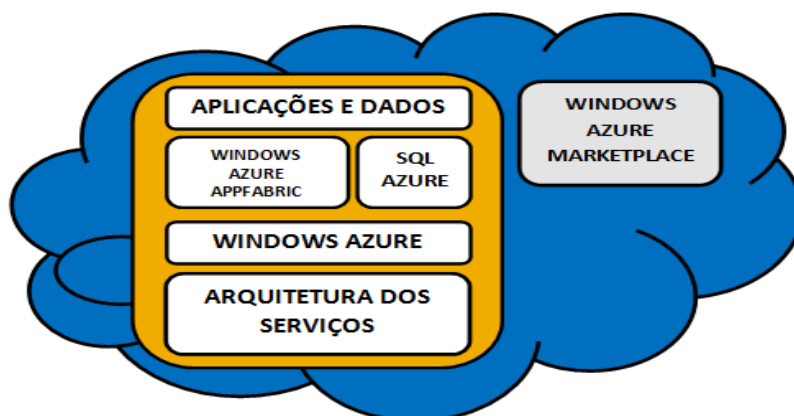


Figura 10 - Componentes da plataforma WA

Na Figura 10 ilustram-se os componentes da plataforma WA que abrange um conjunto de opções direcionadas às necessidades de negócio (110).

Segundo a *Microsoft*, a plataforma WA existentes é um facilitador para a execução, criação, modificação, teste e distribuição de aplicações na Internet com um mínimo de recursos locais, reduzindo custos e esforço de gestão dos recursos de TI (10) (111).

6.1 Arquitetura dos Serviços

O nível de serviço do WA é o PaaS. Em consequência, o utilizador apenas é responsável pelas camadas aplicacionais e de dados, ficando as restantes, nomeadamente, desde a camada de rede até ao *runtime*, inclusive, sob a responsabilidade do fornecedor da plataforma (Figura 11).

A *pool* de recursos, disponibilizados pela plataforma, é constituída por servidores, balanceadores de carga e *hardware* de rede. Para gerir a *pool* existe um controlador, designado por WAFC, que tem como responsabilidade garantir a escalabilidade de recursos e a execução as aplicações (119).

A Figura 11 apresenta a arquitetura dos serviços da plataforma WA (120) (121) (122).

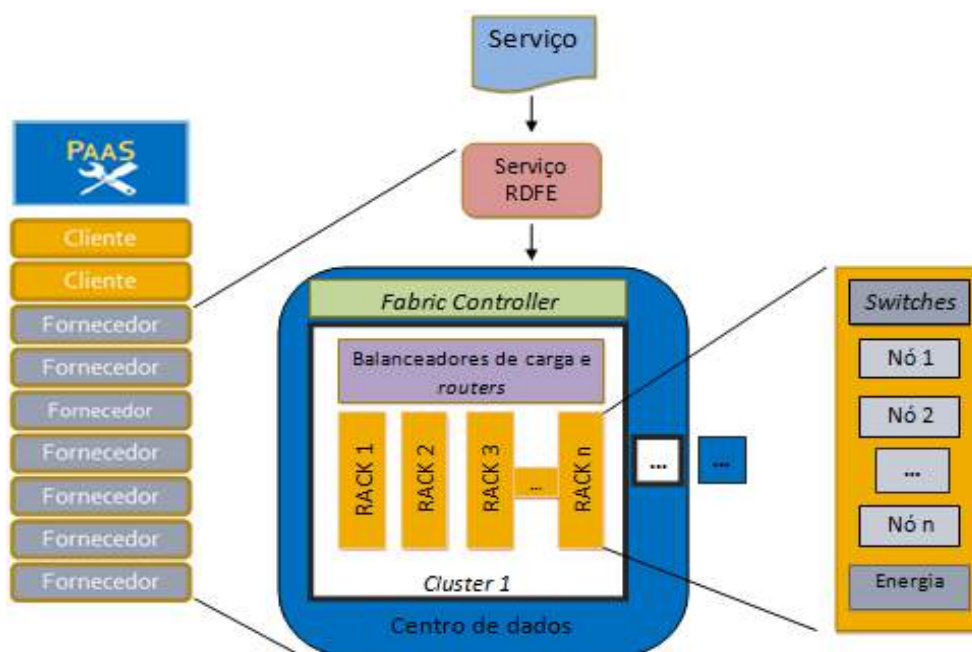


Figura 11 - Arquitetura dos serviços da plataforma WA

Os centros de dados são constituídos por contentores e, por sua vez, por *racks*. Atualmente, os centros de dados são de quarta geração com mecanismo de refrigeração maioritariamente a ar natural. Consequentemente permite a redução de custos tornado mais competitiva a oferta de serviços baseados na nuvem (123) (124).

Para a publicação de um serviço num centro de dados, acede-se ao WA portal, tornado acessível pela criação de uma conta WA (125). Após a fase de autenticação, a informação do serviço é passado ao componente *Red Dog Front End (RD FE)* que, por sua vez, encaminha os ficheiros ao WAFC do centro de dados escolhido. O WAFC analisa as configurações (i.e. descrição do *hardware*, recursos de rede, serviço e do código a ser executado) e provisiona os recursos solicitados.

Cada centro de dados WA, constituído por *clusters*, contentores com aproximadamente 1000 servidores, é gerido por uma instância de WAFC, que tem como função a gestão do *hardware* do centro e dos serviços WA.

As atualizações, o arranque e as respetivas informações, nomeadamente, a descrição do *cluster* físico e dos recursos lógicos, (e.g. endereços IP dos servidores, de rede e unidade de energia), de cada instância de WAFC são feitos pelo controlador *Utility Fabric Controller (UFC)*.

Um *cluster* é constituído por vários *racks*, cada um deles com unidades de potência e comutadores, (e.g. *routers*), para o reencaminhamento de dados entre *racks*. Um *rack* é composto por várias máquinas físicas, designadas por nós (*nodes*). Por sua vez, cada nó pode conter máquinas virtuais. No mínimo, pode conter apenas uma, caso se trate de uma instância de grande dimensão.

No caso de ocorrer uma falha de *hardware* é iniciada uma nova máquina virtual com o sistema operativo WA. O alojamento da mesma pode ser feita no mesmo nó ou noutro já provisionado, desde que exista o espaço necessário.

A comunicação entre o nó e o WAFC do *cluster* é realizada a partir de um componente designado por *Host Agent* existente em cada nó. Por fim, é indicado o endereço da nova máquina ao balanceador de carga responsável por esse serviço (aplicação).

Para garantir a disponibilidade do serviço na plataforma WA, em situações de falha ou de atualizações, é necessária a existência de pelo menos duas máquinas que o alojem. Assim, para a resolução da situação, as máquinas podem ser consecutivamente desligadas, sem deixar de garantir a efetiva execução do serviço.

6.2 Sistema *Windows Azure*

O sistema *Windows Azure* disponibiliza um ambiente de computação e de armazenamento na nuvem. O desenvolvimento das aplicações pode ser feito com outras tecnologias além das da *Microsoft* (23). Salienta-se que a descrição do ambiente de armazenamento está realizada na secção 6.4.

Quanto ao componente WAFC tem a função de gerir as máquinas em execução, originando uma máquina coesa e de forte processamento computacional. Este controlador executa os serviços de computação e de armazenamento (110).

Outro componente, o *Content Delivery Network*, (**CDN**), é responsável pela gestão da *cache* dos dados a que os utilizadores mais acedem, garantindo uma maior velocidade de acesso. O CDN suporta a *cache* em objetos do tipo BLOB (126).

Por fim, o componente *Connect* tem a função de garantir ligações seguras. Fá-lo com a implementação do protocolo *Internet Protocol Security* (**IPSec**), de modo a configurar, ao nível da rede, o tráfego entre o *Windows Azure* e os recursos existentes *on premise* (127).

6.3 *Windows Azure AppFabric*

O WAFC destina-se a executar serviços ao nível da infraestrutura sendo constituído por um componente designado por *Service Bus* (**SB**), implementação do *Enterprise Service Bus* (**ESB**), que permite a comunicação de clientes de diferentes plataformas com os serviços alojados no WA.

O ESB permite que a comunicação atravessasse *firewalls* e dispositivos usando *Network Address Translation (NAT)*. O SB atua como um *router* intermediário entre o cliente e o serviço alojado, não alterando o conteúdo das mensagens (128).

Outro constituinte do WAFC é o *Access Control Service, (ACS)*. Este gere identidades digitais (e.g. *Active Directory, Windows Live ID*), o que facilita a gestão das credenciais e consequentemente do mecanismo de autenticação. O ACS possibilita também a associação de permissões, conforme o papel de cada utilizador na aplicação (129) (130).

No WAFC existe ainda o recurso *Caching* que intervém no armazenamento dos dados mais frequentemente acedidos, reduz o número de acessos à camada de dados, o que garante melhores níveis de desempenho (110).

6.4 SQL Database

O componente *SQL Database* disponibiliza um sistema de gestão de base de dados relacional, juntamente com os serviços de geração de relatórios e de sincronização de dados. Essencialmente o *SQL Database* facilita o armazenamento de dados relacionais, permitindo redução de despesas de capital inicial (custos capex), na aquisição de recursos para o armazenamento de informação.

No componente *SQL Database* tem associado o *SQL Reporting*, que é uma versão do serviço *SQL Server Reporting Services (SSRS)*. Este recurso é responsável pela criação e gestão de relatórios (131).

O último componente do *SQL Database* é o *SQL Data Sync*. Permite a sincronização de dados entre base de dados *Azure* e outras existentes localmente (110).

A plataforma WA, além do componente *SQL Database*, disponibiliza outras abstrações de armazenamento de dados, tais como: objetos *BLOB*, *Table*, *Drive* e *Queue*. Na presente secção, devido às suas características e ao contexto de utilização, apenas serão analisadas a base de dados relacional e as duas primeiras abstrações mencionadas, *BLOB* e *Table*. O tipo *Queue* está analisado na secção 6.7. Para mais detalhes consultar (132) (133).

Um *blob container* contém objetos BLOB. Um objeto *Table* é constituído por entidades com, ou sem, *schemas* iguais. Na composição de objetos de dados BLOB, e de entidades de objetos *Table*, existe um atributo designado por *PartitionKey*. Se, a objetos BLOB do mesmo *blob container*, ou a entidades do mesmo objeto *Table*, for associado o mesmo valor deste atributo, a plataforma garante que os objetos são ordenados e agrupados na mesma partição física do servidor, conseguindo-se assim acessos mais rápidos aos objetos existentes na mesma partição. Ainda no contexto dos objetos *Table*, caso as suas entidades tenham o mesmo valor do atributo *PartitionKey*, é possível realizar transações atómicas, designadas por *Entity Group Transaction*, (EGT) (133).

Cada entidade de uma *Table* ainda inclui na sua composição os atributos *RowKey* e *Timestamp*. Os valores dos atributos *PartitionKey* e *RowKey* definem o índice *cluster* do objeto *Table*, pelo que identificam univocamente as entidades, ordenando-as de um modo ascendente em cada partição. Caso duas entidades tenham o mesmo valor do *PartitionKey*, ambas serão ordenadas de acordo com o valor do *RowKey*.

Cada entidade de um objeto *Table* tem associado o valor do atributo *Timestamp*. A versão da entidade é utilizada no mecanismo de concorrência otimista. O valor do atributo é gerado no lado do servidor e apenas está acessível para leitura. Em acessos concorrentes a uma entidade, esta só é atualizada se o valor do *Timestamp* da versão local coincidir com o valor no servidor. Se tal não acontecer, é lançada uma exceção, pois a entidade foi alterada por outro componente (134).

Caso ocorram catástrofes regionais, a plataforma WA disponibiliza um recurso designado por *Geo Redundant Storage* (GRS) nos objetos de armazenamento de dados. Para garantir maior durabilidade dos dados, as abstrações de dados BLOB e *Table* são replicadas em dois centros de dados do mesmo continente, (e.g. Norte e o Sul dos EUA, Norte e Europa Ocidental, ou Oriente e Sudeste da Ásia) (135).

A Tabela 7 apresenta uma comparação entre a base de dados SQL e as abstrações de dados BLOB e *Table* disponíveis na plataforma WA (136) (137) (138).

	BLOB	Table	SQL Database
Constituição	Um <i>container</i> é constituído por vários BLOB. Para dados de <i>streaming</i> utiliza-se uma lista de <i>Block blob</i> . Para dados aleatórios aconselha-se uma lista de <i>Page blob</i> .	Uma <i>Table</i> é uma coleção de entidades, par nome-valor, com a possibilidade de existirem <i>schemas</i> diferentes.	Compatibilidade com linguagens familiares, como T-SQL e SQL.
Tamanho dos Dados	<i>Block blob</i> até 200GB e <i>Page blob</i> até 1TB.	Cada instância de <i>Table</i> pode ter até 100TB de dados.	Tamanho máximo de 150GB.
Replicação dos dados	É feita a replicação dos dados três vezes, garantindo o balanceamento de pedidos entre as três instâncias. Nos acessos de escrita, a operação tem de ser autorizada por todas.		A replicação dos dados é realizada três vezes. Uma primária e duas secundárias. Os acessos são feitos à replicação primária.
Operações	Inserir, eliminar e reordenar blocos dentro de objetos BLOB.	Suporta as operações CRUD e transações atómicas (EGT).	Suporta operações de filtragem (e.g. junções, vistas).
Desempenho	A partir do serviço CDN disponível na PaaS WA, é possível fazer cache dos objetos BLOB. Estes podem ser divididos por várias partições, de acordo com o valor do atributo <i>PartitionKey</i> .	Recorrendo aos valores dos atributos <i>PartitionKey</i> e <i>RowKey</i> , são realizadas operações como ordenação e partição de entidades.	Cada tabela na base de dados SQL <i>Database</i> tem um índice <i>cluster</i> e pode ter vários índices secundários.
Custos	Depende do espaço ocupado, número de <i>bytes</i> transferidos, a partir do centro de dados, e do número de operações I/O realizadas.		Baseia-se no espaço ocupado e do número de <i>bytes</i> transferidos, a partir do centro de dados.

Tabela 7 - Comparação entre os diferentes tipos de armazenamento na WA

No que diz respeito à arquitetura que sustenta os objetos de dados da plataforma WA, esta é constituída por três camadas: *front-end*, partição e um sistema de ficheiros, distribuído e replicado designado por *Distributed and replicated File System (DFS)* (136).

Ao nível da disponibilidade, cada uma das camadas, além do seu modo de balanceamento automático de carga, tem a capacidade de recuperar e de lidar com falhas. Na camada DFS são feitas réplicas dos dados distribuídos por diversos

servidores. Como esta camada é um sistema de ficheiros distribuído, as réplicas são acessíveis a todos os servidores de partição e de DFS. A consistência dos dados é garantida pela camada de partição, que, utilizando um mecanismo de concorrência otimista, se certifica que cada partição de dados é ordenada e gerida por um único servidor de partição (139).

6.5 Windows Azure Marketplace

O *Windows Azure Marketplace* é uma loja *online*, que disponibiliza dados e aplicações com o objetivo de facilitar a sua procura por parte dos clientes. Este componente é composto pelos recursos *DataMarket* e *AppMarket*. O primeiro destina-se a disponibilizar um conjunto de dados, que podem ser acedidos a partir de tecnologias como RESTful ou a partir do protocolo *Open Data Protocol (OData)*. O segundo recurso, o *AppMarket*, permite que os clientes encontrem e disponibilizem aplicações, facilitando a sua procura de acordo com as necessidades de negócio (110) (118) (140).

6.6 Processadores

No âmbito da WA existem dois tipos de processadores, o *Web Role* e o *Worker Role*.

O processador, designado por *Web Role*, tem como sistema operativo o *Windows Server 2008 R2*, que corre numa máquina virtual e que, por sua vez, está alojado no *Internet Information Services (IIS)*. Este tipo de instâncias é destinada a receber pedidos do utilizador, para a execução de tarefas de curta duração e que mantenham a interação com o utilizador que indicou a tarefa.

Quanto ao segundo processador referido, denominado por *Worker Role*, este é semelhante ao *Web Role*, com a diferença de que não está alojado no IIS destinando-se tipicamente à execução de tarefas de longa duração e não interativas com o utilizador.

6.7 Filas de Mensagens

A WA disponibiliza filas de mensagens, designadamente, *WA Queue*, (**WAQ**), objeto incluído no serviço de armazenamento ou as aprovisionadas pelo componente SB, nomeadamente, *WA Service Bus Queue* (**WASBQ**) e *WA Service Bus Topic*, (**WASBT**) (141).

A fila WAQ possui uma interface REST com algumas operações (i.e. *Get*, *Put* e *Peek*) permitindo manipular mensagens empregues na comunicação entre serviços. Permite a dissociação temporal entre os intervenientes, (*temporal decoupling*), que consiste em emissor e recetor não necessitarem de enviar e receber uma mensagem no mesmo momento temporal, e também a possibilidade do envio e receção de mensagens ocorrerem em taxas diferentes pelos intervenientes (*load leveling*). As mensagens são armazenadas durante um período configurável máximo de sete dias na fila WAQ. Adicionalmente o facto do tamanho da fila se adaptar às mensagens que armazena, permite uma melhor gestão dos recursos (142).

A fila WASBQ, disponibilizada pelo componente SB, garante a ordem temporal em que as mensagens são adicionadas, ou seja, cumpre o protocolo FIFO. Cada mensagem é recebida e processada por um único consumidor. Esta fila apresenta as mesmas vantagens da WAQ e ainda a da utilização do padrão FIFO, garantindo que o consumo de mensagens é realizado pela ordem em que são adicionadas na fila pelos seus produtores (*competing consumer*) (143).

Em oposição à fila WASBQ, a fila WASBT, permite a comunicação entre um produtor e vários consumidores, num padrão *publish/subscribe*. Cada mensagem adicionada fica disponível, na fila de subscrição, para todos os consumidores que nela estão registados, conforme filtros previamente estabelecidos. Como benefícios da sua utilização, além dos indicados para a WASBQ, tem-se o balanceamento de carga pelas várias filas de subscrição que a constituem (*load balancing*) (144).

As filas disponibilizadas pelo componente SB são utilizadas, tipicamente, em cenários onde se pretenda que as mensagens tenham um tempo de vida na fila superior a sete dias, a execução segundo o padrão FIFO, caso seja necessário um mecanismo de *long-polling*, suporte para comunicações *one-to-many* e/ou deteção de duplicação de mensagens. As filas do SB disponibilizam mais características ao nível da

disponibilidade e integração de mensagens do que a fila WAQ, mas têm um custo superior (145).

Resumidamente, a fila de mensagens WAQ e as do SB permitem o armazenamento de mensagens até 64KB e 256KB, respetivamente. Caso haja necessidade de estender esse espaço, pode-se usar outros serviços de armazenamento tais como objetos (e.g. BLOB) ou *cache*. As filas possibilitam também a execução tardia, o processamento de tarefas de longa duração e o desacoplamento entre os intervenientes (146).

6.8 Windows Azure Auto-Scaling Block

Uma característica fundamental de serviços da nuvem é a escalabilidade dos recursos. Para minimizar os custos operacionais (opex), a *Enterprise Library Integration Pack* disponibiliza para a PaaS WA um componente designado por *Windows Azure Autoscaling Application Block (WASABi)*. Este componente permite ajustar o número de instâncias de *roles* às configurações indicadas pelo utilizador. Existem duas possibilidades (147):

- a. O utilizador pode limitar o número mínimo e máximo de instâncias de *roles* a correr na aplicação WA, designadas por *constraints rules*.
- b. O componente WASABi faz o ajuste de instâncias baseado em medidas (*diagnostics*) recolhidas pela plataforma. Esta opção é definida através de *reactive rules*.

Estas métricas ajudam a PaaS WA a ajustar o número de instâncias de *roles* (i.e. componentes de serviços que executam operações), às mudanças existentes na carga de trabalho do negócio e as métricas definidas (148) (149).

7 PROPOSTA

No presente capítulo apresentam-se a arquitetura, os componentes e os protocolos propostos para facilitar a integração de equipamentos de bilhética na CN. A avaliação da arquitetura proposta é efetuada recorrendo à implementação de um demonstrador constituído por um simulador de um equipamento de bilhética e pela implementação da arquitetura proposta (de referência) na plataforma WA.

7.1 Intervenientes

A Figura 12 apresenta os intervenientes e respetivos níveis de serviços disponibilizados e consumidos no contexto da arquitetura proposta.

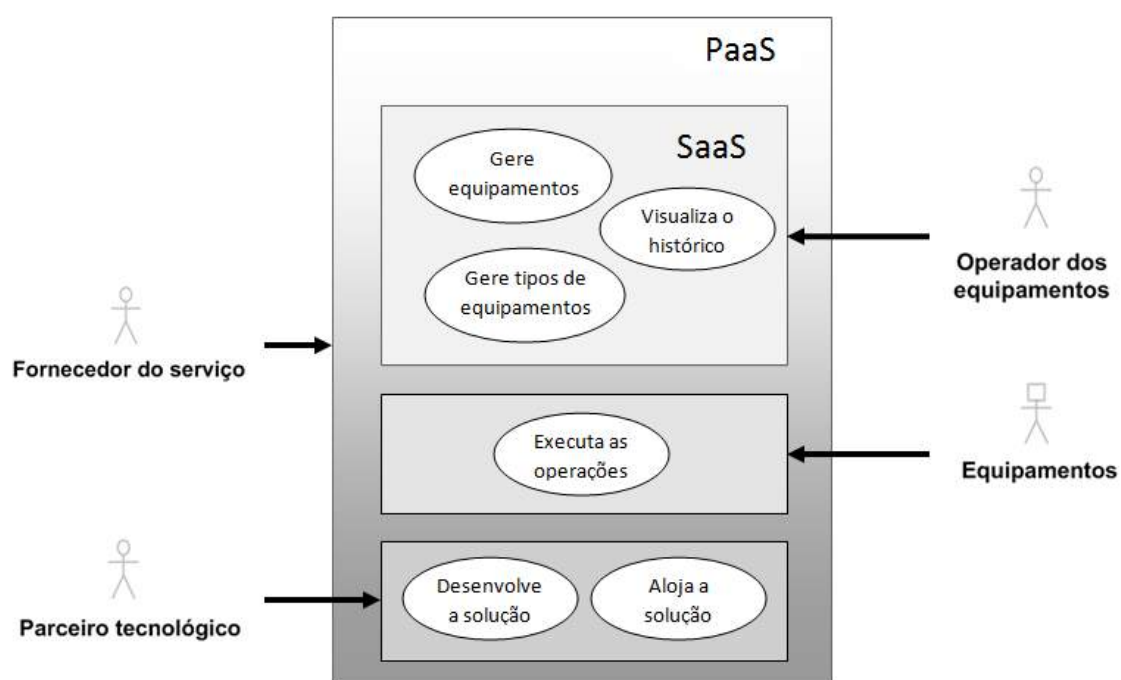


Figura 12 - Níveis de serviços da arquitetura proposta

O fornecedor de serviço disponibiliza o nível de serviço PaaS, mais concretamente recursos e tecnologias à disposição dos parceiros tecnológicos.

Por sua vez, o parceiro tecnológico desenvolve a solução, utilizando os recursos do fornecedor, ao nível do serviço PaaS (e.g. filas de mensagens, processadores). O parceiro tecnológico paga pelos recursos consumidos (PAYG) durante as fases de

desenvolvimento e de alojamento da solução. Após a integração, os equipamentos interagem com a nuvem, enviando pedidos de execução de operações para a nuvem do fornecedor do serviço.

O operador, (e.g. metropolitano de Lisboa), consome um nível de serviço SaaS. O consumo traduz-se na manipulação de configurações e visualização de histórico dos equipamentos integrados, através de uma aplicação sediada na nuvem. O pagamento refere-se ao custo de integração dos equipamentos e dos recursos consumidos pela aplicação referida.

7.2 Arquitetura

A arquitetura proposta para integrar equipamentos numa nuvem computacional é esboçada na Figura 13. A integração de um equipamento pressupõe que o seu comportamento, por exemplo lógica de negócio, é parcialmente suportado na nuvem, considerando, mais do que um tipo de equipamento e recursos computacionais heterogéneos.

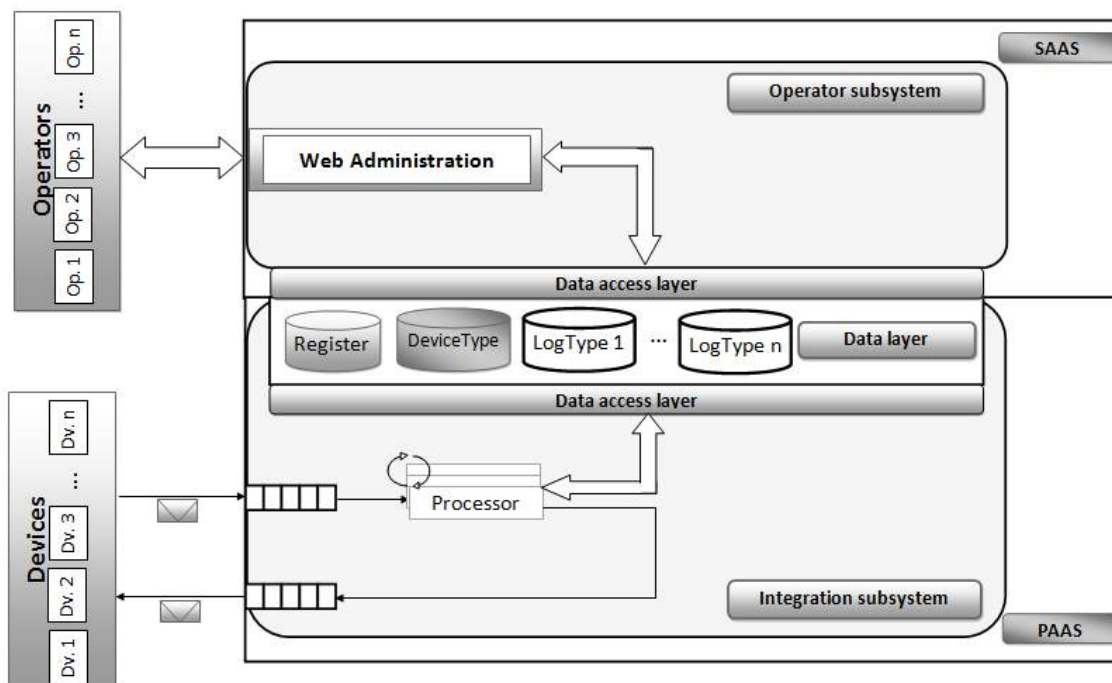


Figura 13 - Arquitetura proposta

Na Figura 13 estão indicados na periferia, dois conjuntos de entidades: os operadores e os equipamentos. O operador manipula as configurações dos equipamentos e acede ao histórico das interações através da aplicação *multi-tenant*, designada por *Web Administration*, que interage com a camada de acesso a dados.

Esta aplicação tem como principais responsabilidades a configuração, acesso ao registo de operação e ativação/desativação dos equipamentos. Estas ações são realizadas pelo respetivo operador (tipicamente proprietário ou concessionário do equipamento).

A configuração de um novo tipo de equipamento pressupõe o procedimento seguinte:

- a. Através da aplicação, o operador regista o tipo de equipamento que fica armazenado de forma persistente no objeto, designado por *DeviceType*.
- b. O comportamento (lógica) do tipo de equipamento é instanciado num componente *Dynamic Link Library (DLL)*, que implementa a interface *IDeviceType*. Posteriormente, a DLL deve ser colocada na pasta do processador (*Processor* - Figura 13). Como os tipos implementam a interface *IDeviceType*, os processadores limitam-se a invocar dinamicamente o seu único método *Process*, consequentemente desencadeia a obtenção do resultado da execução das operações particulares de cada tipo de equipamento.
- c. No marcador *AppSettings* do ficheiro de configuração do componente processador, deve ser adicionada uma entrada com os valores nome do tipo do equipamento e *namespace* da DLL, onde está descrito o tipo, aos atributos *key* e *value* da entrada.

Um protocolo do tipo pedido/resposta suporta a interação dos equipamentos (e.g. validador ou vendedor de bilhetes) com o sistema de informação do operador de transportes (que não faz parte da arquitetura de referência porque pode não estar sediado na nuvem). Este protocolo é suportado por duas filas de mensagens assinaladas com “A”, de entrada, “B”, de saída, respetivamente.

No âmbito do subsistema de integração, as instâncias dos processadores consomem as mensagens da fila “A” e instanciam dinamicamente o processamento, afeto ao tipo de equipamento, que vai alimentar a fila de mensagens “B”.

O contexto de sessão, associada à manutenção de estado do equipamento, varia conforme o seu tipo, por isso, as instâncias de processadores registam-nas em conformidade, o que justifica o armazenamento de objetos com objetivos específicos. Por exemplo, armazenamento de dados de configuração e de histórico de interação com os equipamentos. Para garantir o balanceamento de carga e o isolamento dos dados, o histórico de cada tipo de equipamento está armazenado num objeto, denominado por *LogType n*, onde *n* identifica o tipo do equipamento. As configurações dos equipamentos estão armazenadas num objeto de dados, designado por *Register*. O número de entradas deste objeto coincide com o número de equipamentos integrados na nuvem.

7.2.1 Equipamento

A presente secção tem como objetivo caraterizar o tipo de equipamento suscetível de ser integrado na arquitetura (Figura 13). O equipamento deve ser autónomo com capacidade de execução local (150) (151). Genericamente admite-se que um equipamento está organizado em camadas.

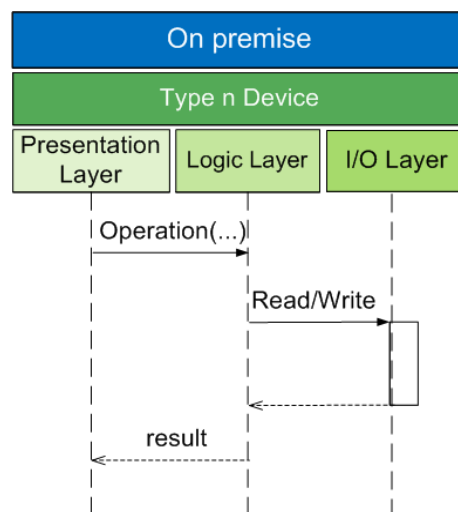


Figura 14 - Diagrama de sequência de um equipamento genérico

Tal como é representado na Figura 14, o equipamento é tipicamente organizado em três camadas, nomeadamente: apresentação (*Presentation Layer*), lógica (*Logic Layer*) e de acesso a recursos (*I/O Layer*). A interação do utilizador com a camada de apresentação desencadeia a execução de operações, no contexto da camada de lógica que, por sua

vez, acedem à camada I/O que manipula recursos (e.g. ficheiros de configuração e de histórico de operações).

7.2.2 Protocolo de Integração

A integração de um equipamento (Figura 14), na arquitetura proposta (Figura 13), implica a realização de adaptações tendo em vista deslocar parcialmente o comportamento do equipamento para a nuvem computacional.

Deste modo, os serviços de bilhética são acedidos através da Internet, usufruindo de recursos elásticos disponibilizados pela nuvem. Os equipamentos tornam-se mais simples porque exigem menos lógica local.

A Figura 15 apresenta o diagrama de classes envolvidas na integração de equipamentos numa CN.

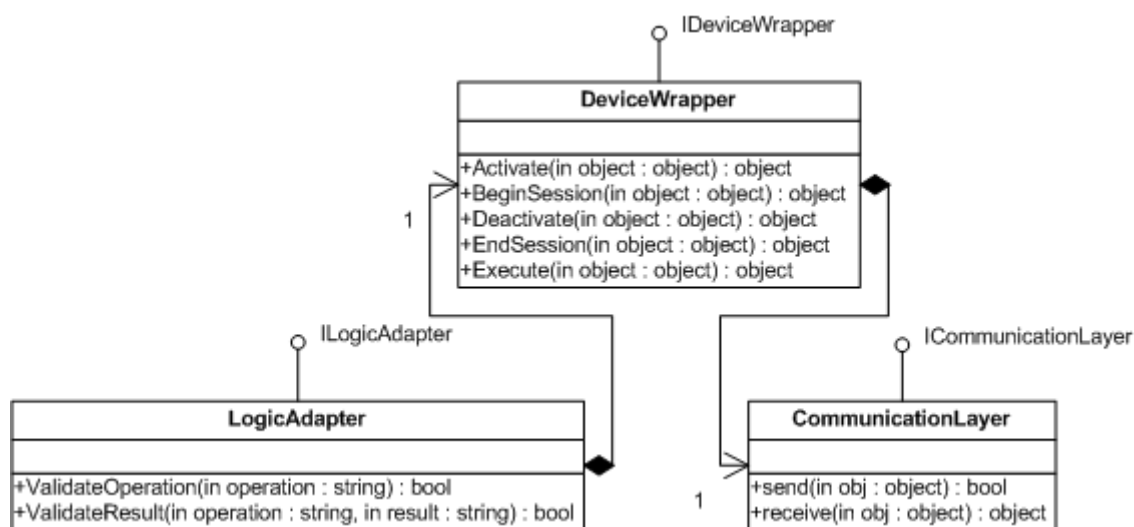


Figura 15 - Diagrama de classes do equipamento

O protocolo de integração proposto admite que a camada de apresentação do equipamento possa ser mantida. A camada de lógica designada por *LogicLayer* (Figura 14) deve ser substituída por uma classe que implemente a interface designada por *ILogicAdapter*. A implementação desta interface é a mesma para todos os equipamentos do mesmo tipo.

Por sua vez, a classe *LogicLayer* realiza os pedidos de execução de operações, interagindo com uma instância da classe do tipo *IDeviceWrapper*. Este contrato disponibiliza um conjunto de operações indispensáveis para completar a lógica do equipamento. A sua implementação é sempre a mesma para todos os equipamentos, independentemente do seu tipo.

Por fim, a classe *DeviceWrapper* facilita o pedido de execução das operações para uma outra classe, que obedece ao contrato *ICommunicationLayer* que expressa a interação com a nuvem. Esta implementação poderá ser comum, caso se pretenda manter o mesmo modelo de comunicação em todos os equipamentos, independentemente do seu tipo.

Na perspetiva da nuvem, o diagrama das classes envolvidas é ilustrado na Figura 16.

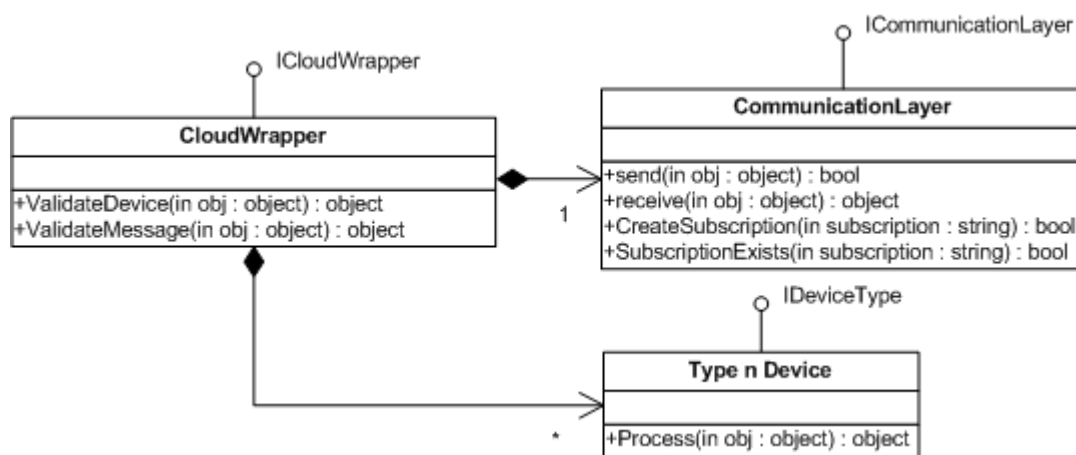


Figura 16 - Diagrama de classes da nuvem

No âmbito da nuvem deve existir um componente que implemente a interface *ICloudWrapper*, de modo a suportar as operações que constituem a lógica alojada na nuvem. Este componente, cuja implementação depende da tecnologia adotada, é responsável por instanciar os tipos de equipamentos que implementam o contrato *IDeviceType*. A devolução do resultado da execução de uma operação é concretizada interagindo com uma instância que implementa a mesma interface que o equipamento, designada por *ICommunicationLayer*. Esta instância acrescenta operações para o estabelecimento da comunicação entre o equipamento e a nuvem (e.g. *CreateSubscription*).

Antes de iniciar as tarefas pretendidas, o operador dos equipamentos está obrigado a um conjunto de ações prévias, ou seja, a integração de um equipamento pressupõe o pré-registo do seu tipo (procedimento indicado na secção 7.2) e adicionalmente:

- a. Através da aplicação de administração (Figura 13) o operador deve fazer o registo do equipamento indicando o seu tipo.
- b. Como resultado do processo de registo a aplicação de administração produz um ficheiro de configuração, que deve ser adicionado ao sistema de ficheiros do equipamento a ser integrado. Após este procedimento, o equipamento fica ativo e em condições de interagir com a nuvem.

O ficheiro de configuração respeita o modelo ilustrado na Figura 17.

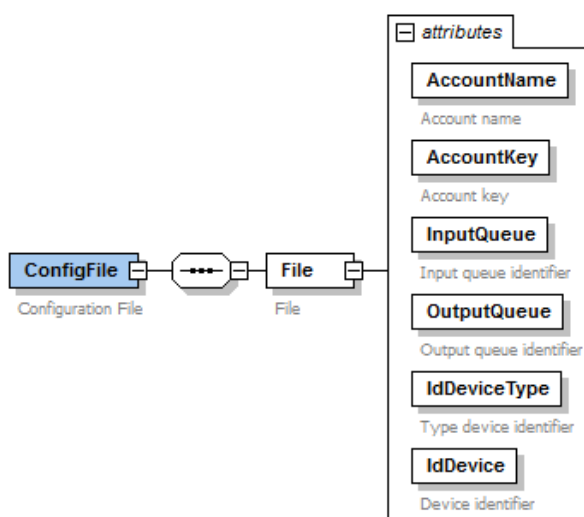


Figura 17- Modelo do ficheiro de configuração do equipamento

Considerando a ordem dos atributos indicados na Figura 17, a informação armazenada são as credenciais de acesso (*AccountName* e *AccountKey*) das filas de mensagens, “A” (*InputQueue*) e “B” (*OutputQueue*), da Figura 13. É igualmente armazenado nos atributos *IdDeviceType* e *IdDevice* respetivamente os identificadores do tipo e do equipamento.

7.2.3 Protocolo de Interação

O protocolo de interação proposto, ilustrado no diagrama de sequência da Figura 18, suporta a interação com os componentes existentes no subsistema de integração da arquitetura (Figura 13).

Após a integração, o equipamento realiza pedidos à nuvem através de uma fila de mensagens. No lado da nuvem (*Cloud*) existem processadores (*Processor*) que recebem as mensagens e que fazem validação do equipamento emissor (*ref Validate device*) e da própria mensagem (*ref Validate message*) confrontando-a com o respectivo XML Schema Definition (*XSD*). Caso se assuma que a mensagem está mal formada, a informação relevante é adicionada no histórico e é terminado o processamento do pedido corrente.

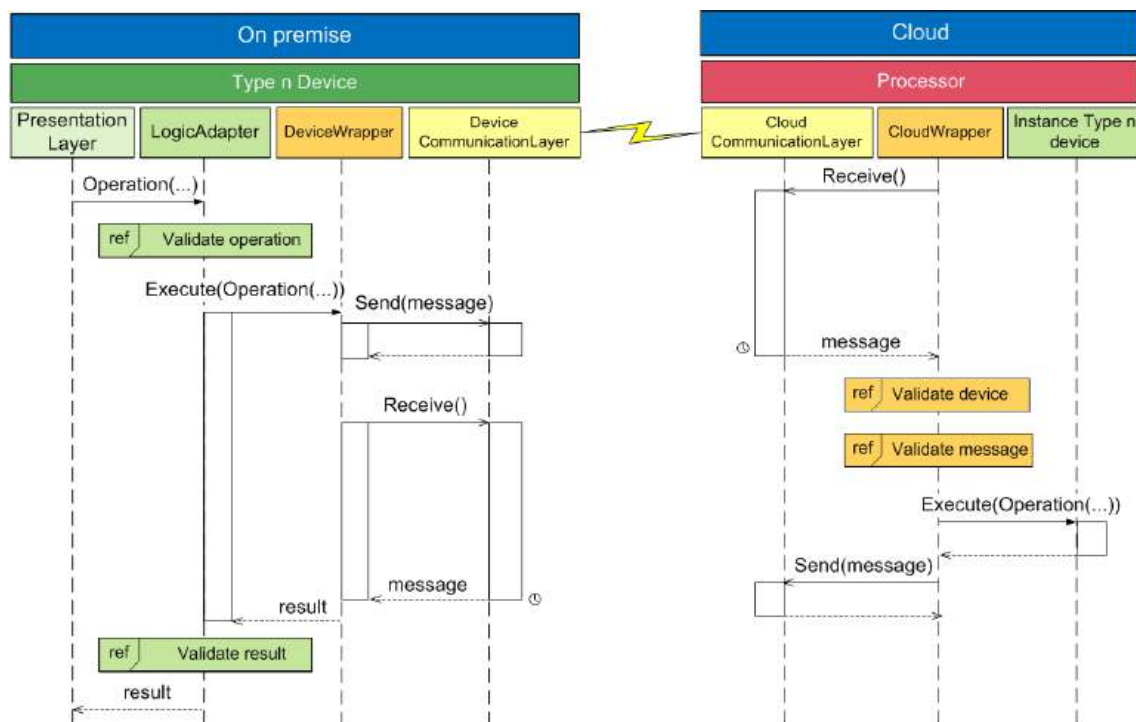


Figura 18 - Diagrama de sequência do protocolo de interação

Caso o equipamento e a mensagem sejam válidos, o processador, face à informação presente no seu ficheiro de configuração e os parâmetros da mensagem, instancia dinamicamente o processamento do seu conteúdo, em conformidade com o identificador do pedido. Após a instanciação e execução da operação o processador regista, no contexto da sessão corrente, o resultado obtido no histórico remetendo para a fila de mensagens de resposta “B”, uma mensagem com o resultado da execução da operação.

Por fim, o equipamento ao receber o resultado, sabe validar a mensagem que o contém e, caso não encontre erros, apresenta-o (se for o caso) ao utilizador, através da sua camada de apresentação.

7.2.4 Mensagens

Na presente secção é apresentado o modelo das mensagens envolvido na interação entre os equipamentos e a nuvem computacional. As mensagens de entrada (fila “A”) e de saída (fila “B”) respeitam o mesmo modelo, de modo a informar os intervenientes, equipamento emissor e o processador consumidor de mensagens da fila “A”, da informação relevante de cada pedido e da respetiva resposta. A Figura 19 apresenta o modelo das mensagens designado por *Message*.

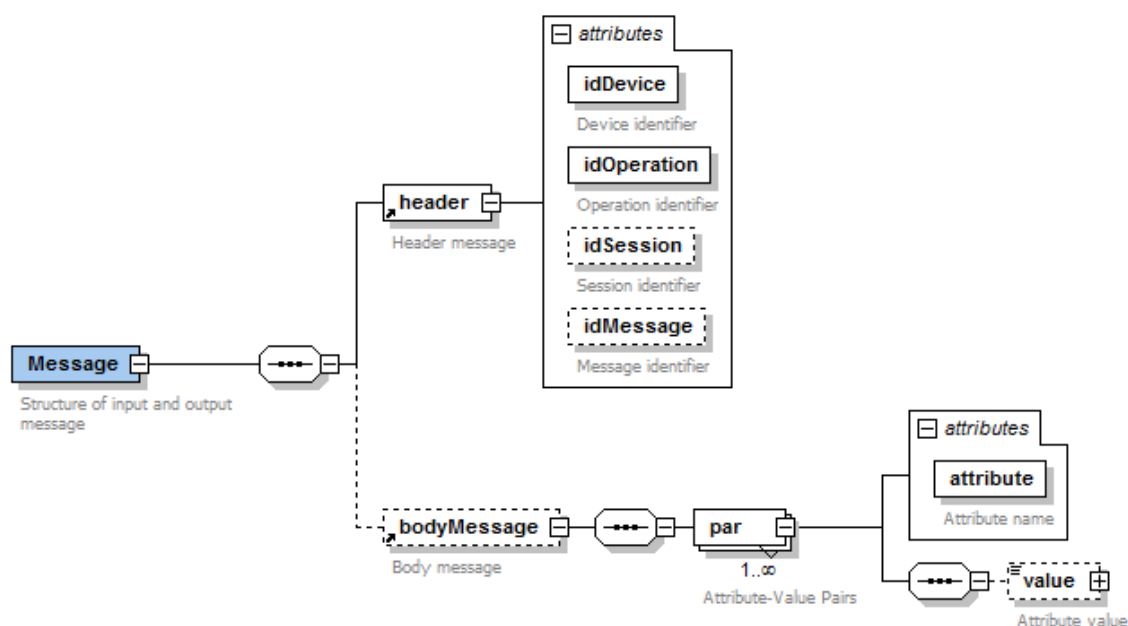


Figura 19 - Modelo das mensagens

Uma mensagem tem como atributos o identificador do equipamento emissor (*idDevice*), o identificador da operação (*idOperation*) e o identificador da sessão (*idSession*), caso já esteja definida, assim como o número de sequência da mensagem no contexto da sessão corrente (*idMessage*). O corpo da mensagem (*bodyMessage*) tem carácter opcional destinando-se a armazenar os parâmetros da operação (*idOperation*), ou o resultado da execução da operação. O corpo da mensagem, caso esteja preenchido, é

constituído por, pelo menos, um elemento “par” com o formato atributo-valor. O valor do atributo deste elemento apresenta uma representação particular de um equipamento.

7.2.5 Modelo dos Objetos de Dados

Nesta secção são indicados os modelos dos objetos existentes na camada de dados da arquitetura (Figura 13).

A Figura 20 apresenta os atributos do objeto de dados, denominado por *Register*, que armazena os dados de configuração dos equipamentos. Estes dados são acedidos pelos processadores para: validação da autenticação, confirmação do estado (ativo ou desativo), determinação do tipo, criação dinâmica de instâncias e gestão das sessões dos equipamentos que emitem os pedidos de execução das operações.

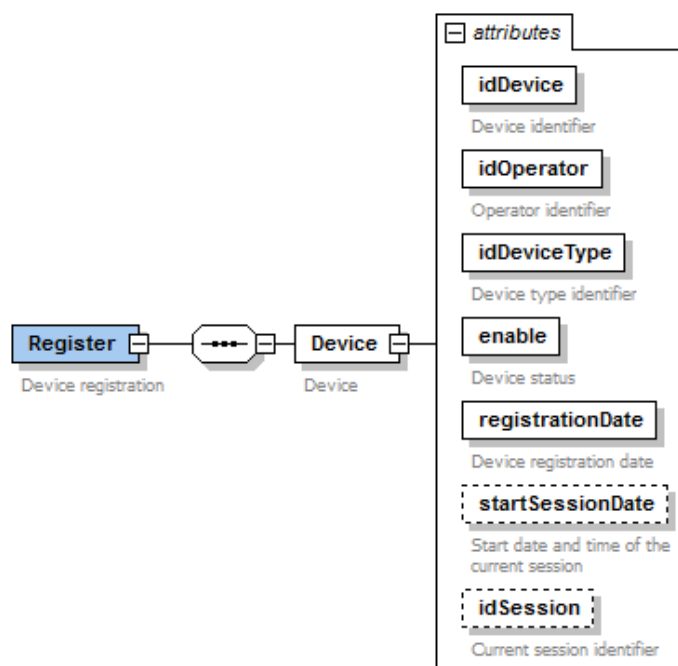


Figura 20 - Modelo do objeto *Register*

Cada entrada do objeto *Register* reúne a informação do identificador do equipamento (*idDevice*), do operador (*idOperator*) e do seu tipo (*idDeviceType*). Adicionalmente, este objeto armazena o estado de ativação (*enable*), a data de registo (*registrationDate*) e a dados sobre a sessão corrente, caso se encontre definida, nomeadamente, a data de início (*startSessionDate*) e o seu identificador (*idSession*).

A Figura 21 apresenta os atributos que constituem o objeto, denominado por *DeviceType* que facilita a gestão dos tipos de equipamentos associados ao respetivos operadores. Sendo assim, a integração de equipamentos por um operador é realizada indicando exclusivamente os tipos que lhe estão associados.

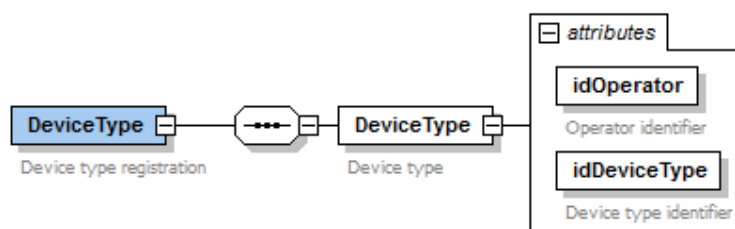


Figura 21 - Modelo do objeto *DeviceType*

Os atributos denominados por *idOperator* e *idDeviceType* são respetivamente os identificadores do operador e do tipo do equipamento.

A Figura 22 ilustra a constituição do objeto, designado por *LogTypeN*, onde n identifica o tipo do equipamento.

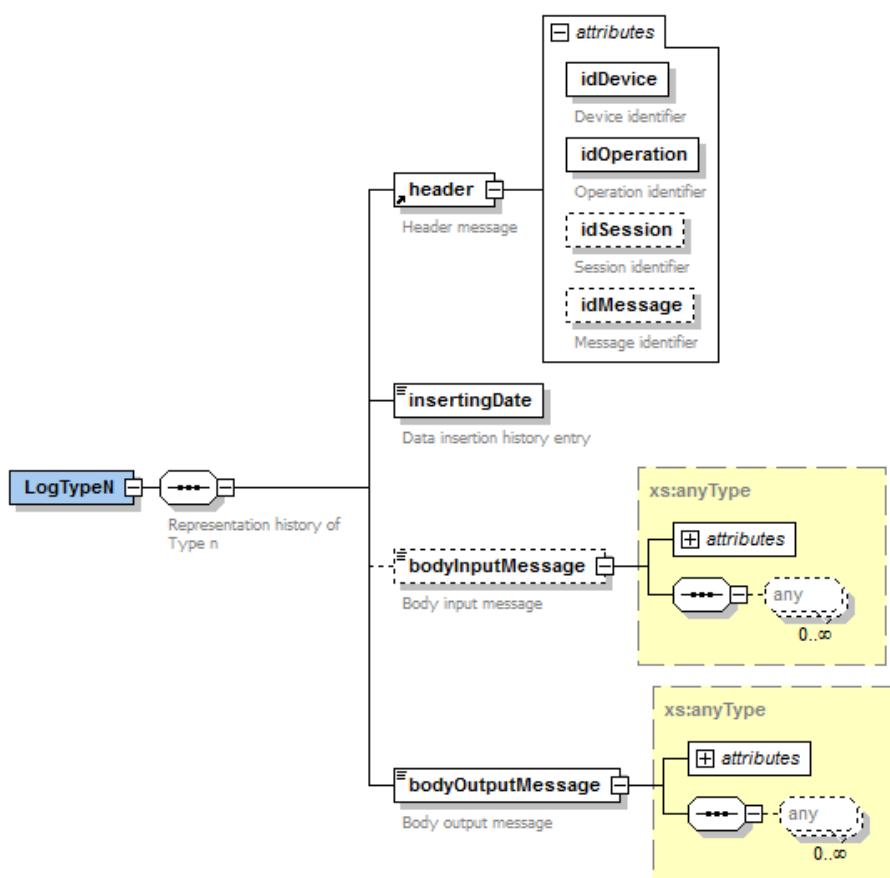


Figura 22 - Modelo do objeto *LogTypeN*

O objeto *LogTypeN* mantém o histórico de operações dos equipamentos do tipo *n*. O histórico fica disponível para consulta do respetivo operador na aplicação de administração (Figura 13).

O modelo da Figura 22 inclui o identificador do equipamento (*idDevice*), o identificador da operação (*idOperation*), o identificador da sessão (*idSession*), caso esteja definida, o número de sequência da mensagem no contexto da sessão corrente (*idMessage*), a data de inserção da entrada no histórico (*insertingDate*) e, por fim, o corpo da mensagem de entrada (*bodyInputMessage*), com carácter opcional e o corpo da mensagem de saída (*bodyOutputMessage*).

Os dois últimos atributos referidos são genéricos e armazenam os parâmetros de entrada, caso existam, e o resultado da operação, respetivamente.

As definições das regras de validação das mensagens e dos objetos de dados são respetivamente indicadas no Anexo A – Regras de Validação da Mensagem e no Anexo B - Regras de Validação dos Objetos de Dados.

7.3 Demonstrador

Nesta secção apresentam-se o simulador do equipamento genérico e a aplicação de administração disponibilizada ao operador, associados à implementação de referência, na plataforma WA, da arquitetura proposta (Figura 13).

7.3.1 Equipamento Genérico

O equipamento genérico é do tipo designado por *ReloadTitlesType* cujo comportamento implementa a simulação da leitura de títulos e possibilita a atualização das respetivas datas de validade.

O *ReloadTitlesType* é um tipo que mantém estado a partir do momento em que é iniciada a sessão. Este estado é mantido até o utilizador indicar que pretende terminar sessão.



Figura 23 - Equipamento genérico

A Figura 23 ilustra a camada de apresentação do equipamento genérico desenvolvido e a respetiva legenda está indicada na Tabela 8.

Item	Significado
1	Apresenta as operações específicas do tipo do equipamento (<i>ReloadTitlesType</i>).
2	Após a seleção da operação que se pretende executar, o utilizador deve pressionar o botão “Executar”.
3	Apresenta o resultado da execução das operações.
4	Indica se a mensagem recebida foi processada com sucesso. Caso contrário, apresenta o erro detetado.
5	Indica o identificador do equipamento. Por se tratar de um simulador, o utilizador pode alterá-lo, simulando um novo equipamento.
6	Apresenta o histórico das operações executadas.

Tabela 8 - Legenda do equipamento genérico

7.3.2 Aplicação de Administração

A interface do operador possibilita a administração dos equipamentos e respetivos tipos, bem como, a visualização do histórico de operações dos mesmos no contexto da nuvem. A Figura 24 ilustra a interface da aplicação de administração do operador.

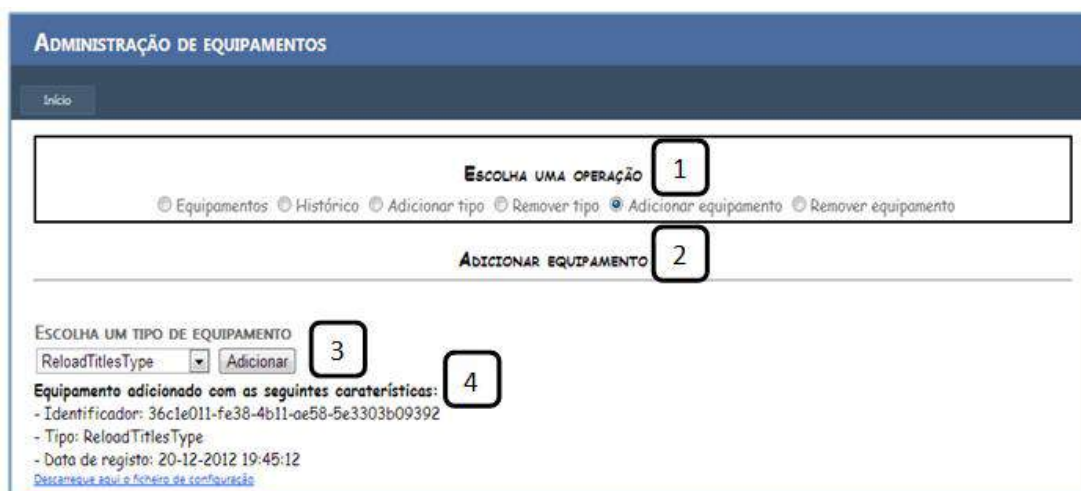


Figura 24 - Aplicação de administração

A legenda da Figura 24 está realizada na Tabela 9. A interface do operador está alojada na plataforma WA (106) e está disponível em <http://operator.cloudapp.net/>.

Item	Significado
1	Apresenta o conjunto de operações disponíveis na aplicação do operador.
2	Indica o nome da operação seleccionada.
3	Indica o nome do tipo do equipamento genérico desenvolvido (Figura 23). O utilizador deve pressionar o botão “Adicionar” de modo a indicar a sua intenção de registar um novo equipamento do tipo <i>ReloadTitlesType</i> .
4	Apresenta a informação de registo do equipamento. A aplicação disponibiliza uma ligação para que o utilizador descarregue o ficheiro de configuração do equipamento.

Tabela 9 - Legenda da interface do operador

7.3.3 Componentes da Plataforma *Windows Azure*

A Figura 25 ilustra a instanciação da arquitetura proposta, Figura 13, assinalando os componentes utilizados da plataforma WA.

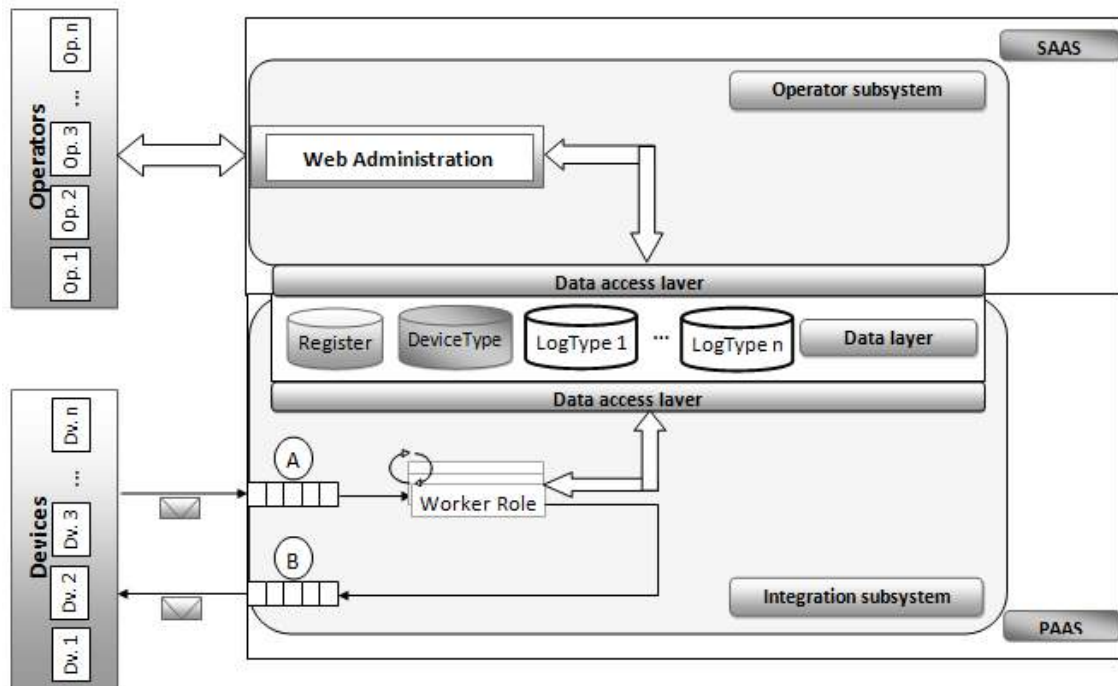


Figura 25 - Arquitetura no contexto da plataforma WA

Para materializar a arquitetura no contexto da WA são necessários os componentes: filas de mensagens (simples e com mecanismos de subscrição), processadores de tarefas assíncronas e objetos para o armazenamento de dados.

7.3.3.1 Tipo de Filas

Na Figura 25 o tipo de fila de mensagens de entrada (i.e. a fila para onde são enviados os pedidos dos equipamentos) é uma WASBQ o que permite a receção de mensagens de um modo assíncrono através de operações de *long-pooling*. Com esta solução, o processador fica bloqueado durante um período de tempo o que, consequentemente reduz o número de transações de interação e os custos associados (144).

Para o mecanismo de notificação dos equipamentos, optou-se pela fila de mensagens do tipo WASBT que disponibiliza o padrão *publish/subscribe* permitindo ao publicador de mensagens enviar para vários consumidores pré-registados (142).

No contexto do demonstrador (Figura 25), o cenário onde se aplica o tipo do processador utilizado é o seguinte: a instância de processador (*Worker Role*) notifica um determinado equipamento, enviando uma mensagem para a respetiva fila de subscrição, com o resultado da execução da operação indicada na mensagem emitida pelo

equipamento. Salienta-se, portanto, a existência de uma fila de subscrição por cada equipamento. Esta solução permite que as mensagens encaminhadas para uma determinada fila sejam recebidas pelo respetivo remetente. Assim, o número de transações de acesso à fila é reduzido, quando comparado com o cenário onde os equipamentos de um determinado tipo estão registados numa fila de subscrição da WASBT.

Para a utilização das filas mencionadas existe a necessidade de referenciar as bibliotecas da plataforma WA, (152) para, posteriormente, poder ser utilizada a API que permite a criação e a interação com filas do tipo SB, WASBT (153) e WASBQ (154).

7.3.3.2 Tipo de Processador

No contexto da arquitetura apresentada na Figura 25, o tipo de processador utilizado para consumir as mensagens da fila de entrada é o *Worker Role*. Justifica-se esta escolha devido a este processador ser destinado à execução de tarefas de longa duração, que não envolvem interação com o utilizador, evitando assim que o emissor fique bloqueado à espera de respostas. A utilização do processador implica a criação de um projeto do tipo *Cloud Services* no ambiente da ferramenta *Microsoft Visual Studio*. Este tipo de projeto está disponível após a instalação do SDK da plataforma WA (152).

7.3.3.3 Tipo de Objetos

Das opções de armazenamento disponíveis na plataforma WA (secção 6.4) optou-se pela *Table*. A *Table* permite acessos rápidos a uma grande quantidade de dados não relacionais no contexto do demonstrador. Esta escolha facilita a existência de um ambiente de dados escalável, com metodologias de acesso direto a dados, evitando a leitura integral dos objetos; abordagens de redundância de dados e balanceamento de carga; disponibilização de funcionalidades que agilizam a implementação (e.g., transações atómicas entre os objetos existentes) a custos mais reduzidos em comparação com a utilização do componente *SQL Database*.

Para o desenho da camada de dados foram considerados aspetos, tais como:

- a. Pertença de todas as entidades a único objeto *Table*, com a definição do mesmo valor do atributo *PartitionKey*. Esta opção facilita o agrupamento dos dados, a ordenação das entidades pelo valor do atributo *RowKey* e a execução de transações atômicas (EGT). Contudo, esta hipótese tem a consequência de limitar a escalabilidade da solução. A atribuição do mesmo valor ao atributo *PartitionKey* origina uma única partição que fica assignada a um só servidor. Este cenário deve ser utilizado, quando há a necessidade de realizar transações entre entidades, em soluções com reduzido número de acessos.
- b. Atribuição de valores de partição distintos a entidades do mesmo objeto *Table*. Esta opção possibilita que o serviço de armazenamento realize o particionamento das entidades por nós do mesmo servidor ou por diferentes servidores, dependendo de fatores inerentes à sua gestão (e.g. necessidade de balancear a carga, escalabilidade da solução). Esta solução é aconselhada para reunir informação do mesmo cliente ou porque os dados a armazenar têm a mesma semântica não necessitando de garantir o seu isolamento.
- c. Dispor as entidades no contexto de objetos *Table* diferentes. Esta circunstância, em comparação com a anterior (b), proporciona o isolamento dos dados dos vários objetos. Tipicamente, este cenário é aproveitado para armazenar dados de diferentes clientes ou porque a informação apresenta semântica distinta entre si. A gestão do alojamento dos objetos pelos servidores é realizada pelo serviço de armazenamento, dependente de fatores como o balanceamento de carga e escalabilidade da solução.

Dos três cenários optou-se pelo que garante uma melhor escalabilidade e balanceamento de carga da aplicação, ou seja, a hipótese c).

O histórico dos tipos de equipamento cliente (e.g. máquina de validação ou venda de bilhetes) consta armazenado num objeto *Table*, designado por *LogType n*. Os atributos *PartitionKey* e *RowKey* têm como valores o identificador do equipamento e o valor resultante da concatenação do identificador da sessão e data de inserção do registo, respetivamente. Com esta decisão, garante-se a existência de registos únicos no mesmo objeto *Table*, agrupando-se os dados pelos identificadores de equipamento e sessão. Esta solução possibilita que numa sessão ocorra mais do que uma operação, mesmo que

repetidas e admite a realização de leituras pelo par, identificadores do equipamento e de sessão, ou exclusivamente pelo identificador do equipamento.

Deste modo, a plataforma garante que os dados do mesmo equipamento estão necessariamente agrupados na mesma partição e ordenados pelo identificador de sessão. Por outro lado, caso a plataforma WA detete que ocorrem acessos para além dos limites de escalabilidade do objeto *Table* referido, a plataforma particiona-o pelo valor do atributo *PartitionKey*, ou seja, pelo identificador do equipamento.

As configurações dos equipamentos estão armazenadas noutra objeto *Table*, designado por *Register*, por se tratar de dados com semântica diferente da do histórico. Os valores dos atributos *PartitionKey* e *RowKey* são respetivamente o identificador do equipamento e do seu tipo. Tendo em consideração o valor do atributo *PartitionKey*, caso exista a necessidade de particionar o objeto *Table* referido para garantir acessos mais rápidos a objetos da mesma partição, a plataforma pode realizá-la, colocando a configuração de cada equipamento num nó de servidor distinto.

O registo dos nomes dos tipos de equipamentos é feito num objeto *Table*, designado na Figura 25 por *DeviceType*. Os atributos *PartitionKey* e *RowKey* têm respetivamente como valores o identificador do operador e do tipo do equipamento. Esta solução permite que os tipos de cada operador fiquem confinados à mesma partição do servidor.

Interessa salientar que para utilizar objetos de dados *Table* num ambiente de desenvolvimento (e.g. *Microsoft Visual Studio*) é necessário referenciar as bibliotecas da plataforma WA (152) e posteriormente utilizar a API que permite a criação e a interação com os objetos de dados alojados na plataforma (135).

Por fim, o armazenamento de ficheiros XSD, que permitem a validação da estrutura das mensagens consumidas pelos processadores *Worker Role*, é feito em objetos BLOB, de modo a que possam ser referenciados nos respetivos processos de validação.

7.4 Considerações Finais

Relativamente às decisões de implementação do demonstrador, pretende-se nesta secção indicar algumas considerações, especificamente:

- a) O limite máximo das mensagens das filas SB (WASBT e WASBQ) é 256 KB. Este valor é considerado suficiente para o envio da informação. Contudo, caso se verifique o oposto, sugere-se a utilização de uma das seguintes soluções: (i) adicionar nas mensagens um número de sequência que indique ao recetor que o conteúdo foi repartido por várias mensagens; (ii) utilizar objetos do tipo BLOB da plataforma WA. Nesta situação, a informação é guardada num ficheiro, armazenado num objeto BLOB, sendo este apenas referenciado nas mensagens.
- b) A opção de uso de objetos *Table*, para a camada de dados do serviço, justifica-se devido aos seus baixos custos (155) quando comparados com o uso do componente *SQL Database*. Considerou-se também que as vantagens superavam as limitações (135) (139) (156).
 - a. Destacam-se as vantagens seguintes:
 - i. Armazenamento de *terabytes* de dados não relacionais, com a capacidade de servir aplicações escaláveis.
 - ii. Possibilidade de desnormalizar estruturas de dados e acessos a partir de índices *cluster*, a fim de permitir acessos mais rápidos a objetos na mesma partição.
 - iii. Cada partição de um objeto *Table* não tem associado um servidor específico. Como os dados estão armazenados na camada DFS, esta característica permite que qualquer servidor possa ter acesso a qualquer partição.
 - b. Estão presentes as limitações seguintes:
 - i. O conteúdo dos objetos *Table* não pode exceder o tamanho máximo de 100TB.
 - ii. Cada objeto *Table* pode servir, incluídas na mesma partição, até 500 entidades por segundo. Todos os pedidos a uma partição são processados por um único servidor de partição.
 - iii. Uma transação atómica apenas é possível entre entidades do mesmo objeto *Table*, que partilhem o valor do atributo *PartitionKey*. Cada transação só pode realizar operações que envolvam no máximo 100 entidades. A atribuição do mesmo valor do referido atributo limita a escalabilidade da solução, pois este está confinado a um único servidor de partição. Este cenário deve

ser utilizado, quando não é esperado demasiado tráfego na mesma partição.

iv. Entidades com elevado número de acessos devem ser distribuídas por diferentes partições, para os pedidos poderem ser atendidos por diferentes servidores de partição.

c) A fila WASBT suporta até 2000 clientes registados e 2000 filtros aplicados. Caso se verifique um número superior, o desenho da arquitetura deve considerar várias instâncias da fila WASBT, com filtros que garantam que os limites não são excedidos (157).

8 EPÍLOGO

Neste capítulo apresenta-se uma avaliação de custos e considerações finais sobre o paradigma CN. Adicionalmente são apresentadas conclusões e perspectivas de desenvolvimentos futuros.

8.1 Avaliação de Custos

Na presente secção pretende-se fazer uma estimativa de custos face aos componentes da plataforma WA utilizados no demonstrador (Figura 13), admitindo como cenário uma empresa de transportes com as caraterísticas seguintes:

- a. Quantidade de passageiros 10.000 por mês;
- b. Cada passageiro usa o seu título de transporte 4 vezes por dia, durante 22 dias num mês.

Para garantir o balanceamento dos pedidos na fila de mensagens “A” (de entrada), admitiram-se duas instâncias de processadores, designados por *Worker Role*, de baixa capacidade de processamento (*Extra Small*).

Pressupondo que o histórico das operações de validação de títulos de transporte é apagado no início de cada mês e considerando as caraterísticas mencionadas, estima-se um total de espaço ocupado de 160 MB por mês, ocupando cada entrada na tabela de histórico 190 bytes. Admite-se a utilização do serviço *Geo Redundant Storage* (GRS), para a redundância dos dados em centros de dados da *Microsoft*. Considerou-se que o espaço ocupado pelas mensagens de ativação dos equipamentos é desprezável face ao espaço ocupado com o seu normal funcionamento.

Por cada pedido, enviado para a fila de mensagens “A”, existem três acessos à camada de dados, um para obter as configurações do equipamento de validação do título e os restantes para a inserção e posterior atualização do histórico do equipamento. Por isso, no final do mês são originadas 2.640.000 transações.

Cada utilização de um título implica o envio e receção de uma mensagem, o que perfaz um total de 1.760.000 mensagens.

Para a estimativa de custos do cenário indicado, consideraram-se os preços indicados em (155), numa política de pagamento baseada em PAYG. A Tabela 10 sistematiza a estimativa descrita.

Componente	Mês	Preço (€)	Obs.
1 <i>Cloud Service</i> (<i>Worker role</i>)	2 instâncias <i>Extra Small</i> (28,80€ /mês)	0,02€/hora /1instância = 14,40€/mês	Instâncias de <i>roles</i> são cobradas após o <i>deployment</i> .
2 <i>Objetos Table</i>	160MB/mês (0,0887€ /mês (GRS))	0,0887€/GB/mês até 1 TB	Com GRS
		0,066€/GB/mês até 1 TB	Sem GRS
	2.640.000 transações/mês (0,18744€ /mês)	0,0071€/ 100.000 transações	-
3 2 filas do SB: WASBQ e WASBT	1.760.000 mensagens/mês/2 filas (1,2496€ de mensagens)	0,0071€/ 10.000 mensagens	-
Total (soma dos valores selecionados a negrito): 30,33€/mês			

Tabela 10 - Avaliação de custos de alojamento na WA

No exemplo de avaliação de custos, apresentado na Tabela 10, foram consideradas instâncias *Cloud Service Extra Small*. Estas instâncias apesar de terem a largura de banda e velocidade de processador inferiores, em relação às restantes, permitem ter em execução seis instâncias *Extra Small* pelo preço de uma *Small*. Neste contexto, consideram-se as características computacionais, das instâncias escolhidas, suficientes para o processamento dos pedidos (155).

Além dos custos indicados na Tabela 10 é necessário ter em consideração os seguintes fatores (155):

- Horas parciais são cobradas como horas completas;
- Após a instalação, é cobrada, pelo menos, uma hora;
- Os objetos *Table*, indicados em 2 na Tabela 10, originam o pagamento apenas da capacidade efetivamente gasta. Ou seja, caso se reserve uma capacidade de 10 GB e apenas a utilize na primeira quinzena do mês, apenas é cobrado uma capacidade de 5 GB no final do mês.

Atendendo a que o protocolo de integração é do tipo pedido/resposta, o tempo expetável das respostas fica dependente das características da rede e da eficiência dos sistemas que executam as tarefas.

8.2 Conclusão

O objetivo de estudar o paradigma CN e de o aplicar na proposta de uma arquitetura para integrar equipamentos de bilhética, de transportes públicos coletivos de passageiros, foi plenamente atingido.

A aquisição de uma infraestrutura própria face à utilização de recursos da nuvem, para a execução de lógica dos equipamentos de bilhética, deve ser ponderada considerando perspectivas de evolução do negócio e das tecnologias subjacentes ao modelo computacional da nuvem.

As considerações sobre o investimento inicial são determinantes. Tipicamente, a utilização de recursos da nuvem, tem custos mais reduzidos (e.g. custo inicial) com uma característica intrinsecamente dinâmica, a elasticidade.

A dependência da rede, o armazenamento dos dados em servidores de terceiros e a velocidade de acesso aos serviços são fatores limitativos. Por isso, a garantia de privacidade e a velocidade de acessos aos dados são aspetos relevantes.

A arquitetura proposta mostra que é possível deslocar a lógica de equipamentos para o contexto de uma nuvem computacional. Porém, antes de se decidir por esta opção, o cliente deve analisar cuidadosamente, na sua perspectiva de negócio, as características, vantagens e limitações dos serviços adquiridos no modelo SaaS.

8.3 Perspetivas Futuras

Como perspetivas futuras para a consolidação da arquitetura proposta (Figura 13) destacam-se:

- a. Identificar possíveis falhas de segurança, por exemplo: falsificação, adulteração, repúdio, divulgação de informações, negação de serviço e/ou elevação de privilégio (*Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege (STRIDE)*) (158);
- b. Estabelecimento de canais seguros com os equipamentos, dependendo das suas limitações face à latência e largura de banda;

- c. Admitir mecanismos de criptografia, de acordo com as limitações de processamento dos equipamentos;
- d. Substituir o acesso à camada de dados por mecanismos de comunicação que façam o desacoplamento entre os dados e o código que os manipula;
- e. Concretizar a arquitetura proposta no âmbito de outras nuvens, por exemplo, de fonte aberta, de modo a comparar o desempenho e custos em diferentes ambientes.

Estes aspetos são fundamentais para melhorar a qualidade da solução proposta tendo em vista facilitar a sua aceitação e adoção pelo mercado dos operadores de transportes.

ANEXO A – REGRAS DE VALIDAÇÃO DA MENSAGEM

Objeto Message
<pre> <xs:element name="Message"> <xs:annotation> <xs:documentation>Structure of input and output message</xs:documentation> </xs:annotation> <xs:complexType> <xs:sequence minOccurs="1" maxOccurs="1"> <xs:element ref="header" minOccurs="1" maxOccurs="1"> <xs:annotation> <xs:documentation>Header message</xs:documentation> </xs:annotation> </xs:element> <xs:element name="bodyMessage"> <xs:annotation> <xs:documentation>Body message</xs:documentation> </xs:annotation> </xs:element> </xs:sequence> </xs:complexType> <xs:sequence minOccurs="1" maxOccurs="1"> <xs:element name="par" maxOccurs="unbounded"> <xs:annotation> <xs:documentation>Attribute-Value Pairs</xs:documentation> </xs:annotation> <xs:complexType> <xs:sequence> <xs:element name="value" type="xs:anyType" minOccurs="0"> <xs:annotation> <xs:documentation>Attribute value</xs:documentation> </xs:annotation> </xs:element> </xs:sequence> <xs:attribute name="attribute" type="xs:string" use="required"> <xs:annotation> <xs:documentation>Attribute name</xs:documentation> </xs:annotation> </xs:attribute> </xs:complexType> </xs:element> </xs:sequence> </xs:complexType> </xs:element> </pre>
<pre> <xs:element name="header"> <xs:annotation> <xs:documentation>Header message</xs:documentation> </xs:annotation> <xs:complexType> <xs:attribute name="idDevice" type="xs:string" use="required"> <xs:annotation> <xs:documentation>Device identifier</xs:documentation> </xs:annotation> </xs:attribute> <xs:attribute name="idOperation" type="xs:integer" use="required"> <xs:annotation> <xs:documentation>Operation identifier</xs:documentation> </xs:annotation> </xs:attribute> </xs:complexType> </xs:element> </pre>

```
</xs:attribute>
<xs:attribute name="idSession" type="xs:string" use="optional">
  <xs:annotation>
    <xs:documentation>Session identifier</xs:documentation>
  </xs:annotation>
</xs:attribute>
<xs:attribute name="idMessage" type="xs:integer" use="optional">
  <xs:annotation>
    <xs:documentation>Message identifier</xs:documentation>
  </xs:annotation>
</xs:attribute>
</xs:complexType>
</xs:element>
```

ANEXO B - REGRAS DE VALIDAÇÃO DOS OBJETOS DE DADOS

Objeto de dados: "Register"
<pre> <xs:element name="Register"> <xs:annotation> <xs:documentation>Device registration</xs:documentation> </xs:annotation> <xs:complexType> <xs:sequence> <xs:element name="Device"> <xs:annotation> <xs:documentation>Device</xs:documentation> </xs:annotation> <xs:complexType> <xs:attribute name="idDevice" type="xs:ID" use="required"> <xs:annotation> <xs:documentation>Device identifier</xs:documentation> </xs:annotation> </xs:attribute> <xs:attribute name="idOperator" use="required"> <xs:annotation> <xs:documentation>Operator identifier</xs:documentation> </xs:annotation> </xs:attribute> <xs:attribute name="idDeviceType" use="required"> <xs:annotation> <xs:documentation>Device type identifier</xs:documentation> </xs:annotation> </xs:attribute> <xs:attribute name="enable" use="required"> <xs:annotation> <xs:documentation>Device status</xs:documentation> </xs:annotation> </xs:attribute> <xs:attribute name="registrationDate" use="required"> <xs:annotation> <xs:documentation>Device registration date</xs:documentation> </xs:annotation> </xs:attribute> <xs:attribute name="startSessionDate"> <xs:annotation> <xs:documentation>Start date and time of the current session</xs:documentation> </xs:annotation> </xs:attribute> <xs:attribute name="idSession"> <xs:annotation> <xs:documentation>Current session identifier</xs:documentation> </xs:annotation> </xs:attribute> </xs:complexType> </xs:element> </xs:sequence> </xs:complexType> </xs:element> </pre>
Objeto de dados: "DeviceType"
<pre> <xs:element name="DeviceType"> </pre>

```

<xs:annotation>
  <xs:documentation>Device type registration</xs:documentation>
</xs:annotation>
<xs:complexType>
  <xs:sequence>
    <xs:element name="DeviceType">
      <xs:annotation>
        <xs:documentation>Device type</xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:attribute name="idOperator" type="xs:ID" use="required">
          <xs:annotation>
            <xs:documentation>Operator identifier</xs:documentation>
          </xs:annotation>
        </xs:attribute>
        <xs:attribute name="idDeviceType" use="required">
          <xs:annotation>
            <xs:documentation>Device type identifier</xs:documentation>
          </xs:annotation>
        </xs:attribute>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
</xs:element>

```

Objeto de datos: "LogTypeN"

```

<xs:element name="LogTypeN">
  <xs:complexType>
    <xs:sequence>
      <xs:annotation>
        <xs:documentation>Representation history of Type n</xs:documentation>
      </xs:annotation>
      <xs:element ref="header">
        <xs:annotation>
          <xs:documentation>Header message</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="insertingDate" type="xs:string" minOccurs="1">
        <xs:annotation>
          <xs:documentation>Data insertion history entry</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="bodyInputMessage" type="xs:anyType" minOccurs="0"
maxOccurs="1">
        <xs:annotation>
          <xs:documentation>Body input message</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="bodyOutputMessage" type="xs:anyType">
        <xs:annotation>
          <xs:documentation>Body output message</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

1. Link. LINK - Gerimos conhecimento, consigo [Internet]. 2011. 2011 [cited 2012 Jul 12]. Available from: <http://www.link.pt/>
2. Group AF, Paper W. Minimizing Cost, Maximizing Value: [Internet]. Network. 2011 [cited 2012 Mar 19]. Available from: http://resources.idgenterprise.com/original/AST-0042368_Farpoint_Group_2011-123.1_-_WLAN_Management_Moves_to_the_Cloud.pdf
3. Rimal BP, Jukan A, Katsaros D, Goeleven Y. Architectural Requirements for Cloud Computing Systems: An Enterprise Cloud Approach [Internet]. Journal of Grid Computing. 2010 [cited 2012 Mar 16]. p. 3–26. Available from: <http://www.springerlink.com/index/10.1007/s10723-010-9171-y>
4. Jon Oltsik. What's Needed for Cloud Computing? [Internet]. Strategy. 2010 [cited 2011 Dec 8]. Available from: http://resources.idgenterprise.com/original/AST-0002397_AST-0002364_riverbed_esg_whatsneededforcloud.pdf
5. Informática T. Glossário - Termos Técnicos de Informática [Internet]. 2012 [cited 2012 Mar 25]. Available from: <http://www.technologica.inf.br/glossario/exibe.asp>
6. Leymann F. Cloud Computing : The Next Revolution in IT [Internet]. 2009 [cited 2012 Jan 10]. p. 3–12. Available from: <http://www.ifp.uni-stuttgart.de/publications/phowo09/010Leymann.pdf>
7. Oracle. ready or not , here comes the cloud Users forging ahead with private cloud while security and. IDG [Internet]. Available from: <http://cloudtimes.org/2011/09/22/ready-or-not-here-comes-the-cloud/>
8. Report E. The Data Center Build-or-Buy Decision : 6 Key Factors You Should Consider. Executive Report [Internet]. 2011; Available from: http://www.linkedin.com/groups/Data-Center-BuildorBuy-Decision-6-2064074.S.104911799?qid=b08f3f70-5b0f-4440-9016-945468a4b90a&goback=.gna_2064074
9. Report S. Private Cloud [Internet]. InfoWorld. 2011 [cited 2012 May 23]. Available from: <http://www.infoworld.com/d/cloud-computing/download-the-private-cloud-deep-dive-168788>
10. Jennings R. Platform, Windows Azure [Internet]. 29/12/2010 ed. Sons JW&, editor. Managing. 2009. p. 360. Available from:

- http://books.google.pt/books/about/Cloud_Computing_with_the_Windows_Azure_P.html?id=KkwsRbTzkKQC&redir_esc=y
11. NIST. The NIST Definition of Cloud Computing [Internet]. 2011. 2011 [cited 2012 Sep 20]. Available from: <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>
 12. Systems C. Cisco Cloud Computing - Data Center Strategy, Architecture, and Solutions [Internet]. 2009 [cited 2012 Dec 6]. Available from: http://www.cisco.com/web/strategy/docs/gov/CiscoCloudComputing_WP.pdf
 13. Chirigati FS. Computação em Nuvem [Internet]. 2009. 2009 [cited 2011 Oct 10]. Available from: http://www.gta.ufrj.br/ensino/eel879/trabalhos_vf_2009_2/seabra/pesquisa.html
 14. IBM. IBM Perspective on Cloud Computing. IBM Corporation [Internet]. 2008; Available from: <http://www.ibm.com/cloud-computing/us/en/>
 15. Computerweekly. A history of cloud computing [Internet]. 2009. 2009 [cited 2011 Nov 23]. Available from: <http://www.computerweekly.com/feature/A-history-of-cloud-computing>
 16. IBM. Seeding the Clouds : Key Infrastructure Elements for Cloud Computing Seeding the Clouds : Key Infrastructure Elements for Cloud Computing [Internet]. Computing. 2009 [cited 2012 Mar 12]. Available from: http://resources.idgenterprise.com/original/AST-0001412_ibm_seedingthecLOUDS.pdf
 17. Documentation A. Welcome to Amazon S3 [Internet]. 2006 [cited 2012 Jan 11]. Available from: <http://docs.amazonwebservices.com/AmazonS3/latest/dev/Welcome.html?r=3744>
 18. Baca S, Instructor GK. Expert Reference Series of White Papers Cloud Computing : What It Is and What It Can Do for You Cloud Computing : What It Is and. Knowledge Creation Diffusion Utilization [Internet]. 2010;1–6. Available from: http://www.globalknowledge.se/content/files/documents/338677/White_Paper_Cloud_Computing_What_It_Is.pdf
 19. Raghupathi K. 5 Key Events in the history of Cloud Computing [Internet]. 2011 [cited 2012 May 12]. Available from: <http://cloud.dzone.com/news/5-key-events-history-cloud>
 20. Infographic TC. The History of Cloud Computing [Internet]. 2012. 2012 [cited 2012 Jun 11]. Available from: <http://www.thecloudinfographic.com/2012/01/16/history-of-cloud-computing.html>

21. ComputerWorld. Cinco grandes mudanças para a Internet em 2012 [Internet]. 2012. 2012 [cited 2011 Nov 3]. Available from: <http://www.computerworld.com.pt/2012/01/03/cinco-grandes-mudancas-para-a-internet-em-2012/>
22. Networks AB. Building a smarter network with IPAM [Internet]. Time. [cited 2012 May 6]. Available from: http://resources.idgenterprise.com/original/AST-0059678_Whitepaper_Building_a_Smarter_Network_with_IPAM.pdf
23. Castro LAS. Controlo de infra-estruturas de Cloud Computing. Universidade de Aveiro [Internet]. 2009; Available from: <http://ria.ua.pt/handle/10773/2105>
24. Bruno Schulze GC. Digital library [Internet]. Proceedings of the 6th international workshop on Middleware for grid computing. 2008 [cited 2012 Feb 20]. Available from: http://dl.acm.org/citation.cfm?id=1462704&picked=prox&CFID=79055625&CF_TOKEN=62949480
25. Marston S, Li Z, Bandyopadhyay S, Zhang J, Ghalsasi A. Cloud computing — The business perspective [Internet]. Decision Support Systems. Elsevier B.V.; 2011 [cited 2012 Mar 19]. p. 176–89. Available from: <http://linkinghub.elsevier.com/retrieve/pii/S0167923610002393>
26. Report EG. The Future of Cloud Computing [Internet]. Analysis. 2010 [cited 2012 Apr 1]. Available from: <http://cordis.europa.eu/fp7/ict/ssai/docs/cloud-report-final.pdf>
27. Armbrust, M., Fox, A., Griffith, R. J. M. Above the Clouds: A Berkley View of Cloud Computing [Internet]. Armbrust M, Fox A, Griffith R, Joseph AD, Katz RH, Konwinski A, et al., editors. California: Electrical Engineering and Computer Sciences University of California at Berkeley; 2009. Available from: <http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html>
28. Seabra F. Computação em Nuvem [Internet]. 2007. 2007 [cited 2012 Jan 4]. Available from: http://www.gta.ufrj.br/ensino/eel879/trabalhos_vf_2009_2/seabra/componentes.html
29. Workflow W. Principles of Elastic Processes [Internet]. Quality. 2011 [cited 2012 Jun 22]. Available from: <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=06015579>
30. Dr. Fabrizio Gagliardi. Grid and cloud computing for e-Science [Internet]. 2008 [cited 2012 Mar 4]. Available from: http://research.microsoft.com/pubs/78984/TCI_MSR_IGTC2008_Israel_Fab.pdf
31. Metzler BJ. Virtualization : Benefits , Challenges , and Solutions. Technology [Internet]. 2011; Available from: <http://www.cloudcomputinglive.com/online/images/zeuswhitepaper2.pdf>

32. vmware. Real world virtualization for your business [Internet]. Business. [cited 2012 Jan 6]. Available from: http://resources.idgenterprise.com/original/AST-0056716_eBook-Real-World-Virtualization-For-Your-Business.pdf
33. Services R. Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate Online [Internet]. Que. Miller M, editor. Online. 2008. Available from: <http://achariya.org/images/asbt/Engineering/RESOURCEDOWNLOAD/M.TECH/SEM3/COMPUTERSCIENCE/Cloud Computing/Cloud Computing Web-Based Applications That Change The Way You Work And Collaborate.pdf>
34. Calheiros RN. Automated Emulation Of Distributed Systems Through System Management And Virtualization. System [Internet]. 2009; Available from: <http://www.cloudbus.org/students/RodrigoThesis2009.pdf>
35. Ion AT. WHAT HAS I . T . TRIED IN THE PAST ? [Internet]. 2011 [cited 2012 Jan 25]. Available from: http://resources.idgenterprise.com/original/AST-0055046_19619_WhoMovedMyApp_WP.pdf
36. Emerson Network Power. Planning for Cloud Computing: Data Center Infrastructure Considerations [Internet]. 2010 [cited 2012 Mar 12]. Available from: <http://www.emersonnetworkpower.com>
37. IBM. Taking the Enterprise Data Center into the Cloud [Internet]. Computing. 2010 [cited 2012 May 2]. Available from: <http://dcs.asu.edu/faculty/BruceMillard/indexonly/OffTheNet/>
38. Vaquero LM, Rodero-merino L, Caceres J, Lindner M. A Break in the Clouds : Towards a Cloud Definition. Computer Communication Review [Internet]. 2009;39(1):50–5. Available from: <http://dl.acm.org/citation.cfm?id=1496100>
39. Rodrigues AS, Gouvêa TM, Pereira F, Educacional A, Bosco D. Cluster e Grid Computing [Internet]. 2006. 2006 [cited 2012 Mar 16]. Available from: <http://www.aedb.br/seacIV/SI/Poster/grid.pdf>
40. Bookman C. Agrupamento de computadores em linux [Internet]. Moderna C, editor. 2003. p. 240. Available from: http://openlibrary.org/books/OL13413226M/Agrupamento_de_Computadores_em_Linux
41. Buyya R, Yeo CS, Venugopal S, Computing G, Engineering S. Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities. Media [Internet]. 1969; Available from: <http://arxiv.org/ftp/arxiv/papers/0808/0808.3558.pdf>
42. Flávio R. C. Sousa LOM e JCM. Computação em Nuvem: Conceitos, Tecnologias, Aplicações e Desafios 1 2. Universidade Federal do Ceará (UFC) [Internet]. 2010; Available from: http://www.es.ufc.br/~flavio/files/Computacao_Nuvem.pdf

-
43. Jennings R. Utility-based Resource Management for Cluster Computing [Internet]. Department. Yeo CS, editor. Engineering. January 2008; 2008. Available from:
https://www.google.pt/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&ved=0CDcQFjAA&url=http%3A%2F%2Fciteseerx.ist.psu.edu%2Fviewdoc%2Fdownload%3Fdoi%3D10.1.1.144.8256%26rep%3Drep1%26type%3Dpdf&ei=56DcUJGhC8SKhQemuYHwBA&usg=AFQjCNH3eKAIE3OvhMAIcy_S7SI5sI_leg&sig2=Ba_cIeMTAgLVci-WsYkdnQ&bvm=bv.1355534169,d.ZG4
 44. Engineering S. Meta Scheduling for Market-Oriented Grid and Utility Computing Saurabh Kumar Garg Doctor of Philosophy. October [Internet]. 2010;(June). Available from: <http://www.cloudbus.org/students/Saurabh-PhD-Thesis2010.pdf>
 45. Buyya R, Yeo CS, Venugopal S, Broberg J, Brandic I. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility [Internet]. Future Generation Computer Systems. Elsevier B.V.; 2009 [cited 2012 Mar 15]. p. 599–616. Available from:
<http://linkinghub.elsevier.com/retrieve/pii/S0167739X08001957>
 46. Armbrust M. A View of Cloud Computing [Internet]. 2010. 2010 [cited 2011 Dec 13]. Available from: <http://www.cs.berkeley.edu/~rxin/db-papers/cloudcomputing.pdf>
 47. Jatit. Comparison of Grid Computing vs. Cluster Computing [Internet]. [cited 2012 Mar 8]. Available from:
http://www.jatit.org/research/introduction_grid_computing.htm
 48. Buyya R, Ranjan R, Calheiros RN. Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit : Challenges and Opportunities. Department of Computer Science and Software Engineering The University of Melbourne, Australia [Internet]. 2009; Available from:
<http://arxiv.org/ftp/arxiv/papers/0907/0907.4878.pdf>
 49. Leymann F. Cloud Computing - The Next Revolution in IT [Internet]. 2009 [cited 2012 Jan 14]. Available from: <http://www.ifp.uni-stuttgart.de/publications/phowo09/010Leymann.pdf>
 50. Mirashe SP. Cloud Computing. Journal of Computing [Internet]. 2010;2(3):78–82. Available from: <http://arxiv.org/ftp/arxiv/papers/1003/1003.4074.pdf>
 51. Oracle A, Paper W. Database Consolidation onto Private Clouds. Building [Internet]. 2011;(February). Available from:
<http://www.oracle.com/us/products/database/database-private-cloud-wp-360048.pdf>
 52. Ana PV. Cloud Computing : Impasses Legais E Normativos Introdução. 2010;16–25. Available from:
http://www.uniesp.edu.br/guaruja/site/revista/revista20102/PDfs/cloud_computing_2.pdf
-

53. Armbrust M, Fox A, Griffith R, Joseph AD, Katz R, Konwinski A, et al. Above the Clouds : A Berkeley View of Cloud Computing Cloud Computing : An Old Idea Whose Time Has (Finally) Come [Internet]. Computing. 2009 [cited 2012 Apr 20]. p. 7–13. Available from: [http://x-integrate.de/x-in-cms.nsf/id/DE_Von_Regenmachern_und_Wolkenbruechen_-_Impact_2009_Nachlese/\\$file/abovetheclouds.pdf](http://x-integrate.de/x-in-cms.nsf/id/DE_Von_Regenmachern_und_Wolkenbruechen_-_Impact_2009_Nachlese/$file/abovetheclouds.pdf)
54. Armbrust B, Griffith R, Joseph AD, Katz R, Konwinski A, Lee G, et al. A View of Cloud Computing. Ariel [Internet]. 2010; Available from: <http://dl.acm.org/citation.cfm?id=1721672>
55. Christina Lau VB. Best practices to architect applications in the IBM Cloud [Internet]. IBM. 2011 [cited 2012 Jan 18]. Available from: http://resources.idgenterprise.com/original/AST-0057343_ibm_bestpracticestoarchitectapps.pdf
56. Process T, Enterprise E, Think S. for the cloud [Internet]. 2010 [cited 2012 Jun 4]. Available from: <http://www.nerdzwirk.com/clients/resources>
57. Alliance CS. Security Guidance for Critical Areas of Focus in Cloud Computing [Internet]. 2009 [cited 2012 May 22]. Available from: <http://pt.scribd.com/doc/15055401/Security-Guidance-for-Critical-Areas-of-Focus-in-Cloud-Computing>
58. Terry D. ACM Tech Pack on Cloud Computing. Management [Internet]. 2008;7(July):9–11. Available from: http://techpack.acm.org/cloud/cloud_computing.pdf
59. Spring. Open Cloud Manifesto. Open Cloud Manifesto [Internet]. 2009;0–7. Available from: www.opencloudmanifesto.org
60. IBM. Enterprise infrastructure as a service [Internet]. 2011 [cited 2012 Jun 9]. Available from: http://resources.idgenterprise.com/original/AST-0057341_ibm_smartcloudenterprise.pdf
61. Schmidt E. Computação na Nuvem [Internet]. 2006. 2006 [cited 2012 Jan 4]. p. 18–24. Available from: http://www.maxwell.lambda.ele.puc-rio.br/19632/19632_3.PDF
62. Go J. Hosting the Scaling Engine [Internet]. Windows Azure. 2011 [cited 2012 Mar 18]. Available from: <http://msdn.microsoft.com/en-us/library/windowsazure/hh508979.aspx>
63. Banerjee N. Introduction to Cloud Computing [Internet]. 2011. Harvard University Extension School; 2011 [cited 2011 Oct 16]. Available from: http://www.isical.ac.in/~ansuman/dist_sys/CloudComputingIntro.pdf
64. Azure M. HPC and Azure [Internet]. 2012. 2012 [cited 2012 Jan 9]. Available from: <http://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=9&ved=0>

- CG4QFjAI&url=http%253A%252F%252Fflwaugbe.blob.core.windows.net%252Ftalks%252FHPC%252520and%252520Azure%252520Feb%25252012.pptx&ei=WQjqT7ufKsexhAf7jN0W&usg=AFQjCNF9wzVWPFqrpCB8v0g3gbsFksAeCA&sig2=dumNAppqh6xC20eM0oCy7qg
65. Bogdan A. Windows Azure EBC TDM Overview [Internet]. Microsoft; [cited 2012 Apr 20]. Available from:
<http://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=13&ved=0CGIQFjACOAo&url=http%3A%2F%2Fthor.info.uaic.ro%2F~adria%2Fteach%2Fcourses%2Fpcd%2Fresources%2FMicrosoft%2FWindows%2520Azure%2520EBC%2520TDM%2520Overview.pptx&ei=kgjqT-3lF4y7hAfr6NUz&usg=AFQjCNHsb39esN827OJzJKWPRjyUXUCIHA&sig2=0D1CPbYNzUozQo5CZH63LQ>
 66. Herbert L, Erickson J, Jones D, Stanton R. The ROI Of Cloud Apps. Reproduction [Internet]. 2011; Available from:
http://resources.idgenterprise.com/original/AST-0042573_Forrester_Report_The_ROI_of_Cloud_Apps.pdf
 67. William Voorsluys, James Broberg Buyya R. Introduction To Cloud Computing. Foundations [Internet]. 2011; Available from:
<http://www.thecloudforge.org/pubs/cloud-computing>
 68. Computing C, Case U, Group D, Amrhein CD, Anderson P, Andrade AD, et al. Cloud Computing Use Cases A white paper produced by the. October [Internet]. 2009; Available from:
https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&ved=0CDQQFjAA&url=http%3A%2F%2Fcloudusecases.org%2FCloud_Computing_Use_Cases_Whitepaper-4_0.odt&ei=pKvcUPPqM4imhAeZ7oCYBQ&usg=AFQjCNGjAUZ2GH6Z6VuG5qxU1CLcLSa_ug&sig2=FMiU4OLp7-OFJcRX3cl-ig&bvm=bv.1355534169,d.ZG4
 69. Subramanian K. Hybrid Clouds [Internet]. Analyst & Researcher Krishworld.com. 2011 [cited 2012 May 12]. Available from:
<http://www.vmware.com/files/uk/pdf/IDCWP32S-web.pdf>
 70. InfoWorld. Cloud Computing. InfoWorld [Internet]. 20

- http://resources.idgenterprise.com/original/AST-0056071_Accellion_Secure_File_Sharing_Cloud_Whitepaper.pdf
73. Networking C. The 2011 Cloud Networking Report [Internet]. Computing. 2011 [cited 2012 May 23]. Available from: http://resources.idgenterprise.com/original/AST-0054943_2011CNR-Part1Dell.pdf
74. Revival R, Paradigm N. Cloud Computing : Retro Revival or the New Paradigm ? Wiley Periodicals, Inc. [Internet]. 2010; Available from: <http://onlinelibrary.wiley.com/doi/10.1002/jcaf.20605/abstract>
75. Webroot. Cloud Vs. Cloud: Which Side Are You On? [Internet]. 2004. 2004 [cited 2012 May 21]. Available from: <http://itpro.webroot.com/articles/cloud-vs-cloud-which-side-are-you-on.html>
76. Engineering S. Authorisation and Accounting Services for the World Wide Grid. Direct [Internet]. 2004;(June). Available from: <http://www.cloudbus.org/students/AlexMastersThesis.pdf>
77. Antedomenico N. Optimizing Security Of Cloud Computing Within The Dod [Internet]. Security. NAVAL POSTGRADUATE SCHOOL; 2010. Available from: <http://www.amazon.com/Optimizing-Security-Computing-within-ebook/dp/B0075R6KKA>
78. Lau C. Best practices to architect applications in the IBM Cloud [Internet]. 2011. 2011 [cited 2012 Mar 3]. p. 1–15. Available from: http://resources.idgenterprise.com/original/AST-0057343_ibm_bestpracticestoarchitectapps.pdf
79. Force.com. Force . com : A Comprehensive Look at the World ’ s Premier Cloud-Computing Platform. Salesforce.com cloud plataform [Internet]. 2009; Available from: http://www.developerforce.com/media/Forcedotcom_Whitepaper/WP_Forcedotcom-InDepth_040709_web.pdf
80. Paper ANW. Understanding Security in Cloud Storage [Internet]. 2010. 2010 [cited 2012 Apr 15]. Available from: http://resources.idgenterprise.com/original/AST-0054142_Understanding_Security_in_Cloud_Storage.pdf
81. Clarity EM, Deployments PC. Fair to Partly Cloudy Fair to Partly Cloudy [Internet]. NimSoft. 2011 [cited 2011 Nov 10]. Available from: http://resources.idgenterprise.com/original/AST-0053500_Ensuring_Monitoring_Clarify_in_Public_Cloud_Deployments--FusionStorm.pdf
82. Alliance CS. Security Guidance Critical Areas of Focus for [Internet]. Security. 2009 [cited 2012 May 2]. p. 1–76. Available from: <https://cloudsecurityalliance.org/csaguide.pdf>

83. Buyya R, Pandey S, Vecchiola C. Cloudbus Toolkit for Market-Oriented Cloud Computing. Computing [Internet]. 2009;24–44. Available from: https://www.google.pt/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CEEQFjAA&url=http%3A%2F%2Fciteseerx.ist.psu.edu%2Fviewdoc%2Fdownload%3Fdoi%3D10.1.1.148.3039%26rep%3Drep1%26type%3Dpdf&ei=DZ_cULa-IsiThgfQ1YGoBQ&usg=AFQjCNHIJ9pDnQsMKG0TtVd3yJbgX3DIFQ&sig2=3XtPEYFqucjx3MFbufH2hw&bvm=bv.1355534169,d.ZG4&cad=rja
84. Matos MMD. Metodologia De Análise De Impactos após A Implementação De Sistemas De Gestão Empresarial [Internet]. 2004 [cited 2012 Apr 19]. Available from: <http://pt.scribd.com/doc/4672891/Tese-Impactos-apos-a-implementacao-de-um-ERP>
85. Pattabhiram BC, Anna JD. Integration : The Critical Path to Cloud Computing New Capabilities , New Questions. Computing [Internet]. 2012;(May 2010). Available from: <http://www.csitd.co.uk/PDFS/ISD/integration - critical path.pdf>
86. Cuts HS, Cost H, By I. White Paper Dollars & Sense of Identity Management Big Savings From Cloud-Based Identity Beyond the Bottom Line [Internet]. Identity. 2011 [cited 2012 Feb 13]. Available from: http://resources.idgenterprise.com/original/AST-0054318_TCO_White_Paper.10.2011-IDG.pdf
87. Eid T. Cloud Single Sign-On for SaaS Providers SaaS Success Criteria and the Role of SSO [Internet]. 2012. 2012 [cited 2012 Jun 19]. Available from: http://resources.idgenterprise.com/original/AST-0059750_Montclair_SaaS_White_Paper_3-1_v2_finalmarch2012.pdf
88. Fransham K, Agarwal A, Armstrong P, Bishop A, Charbonneau A, Desmarais R, et al. Research computing in a distributed cloud environment [Internet]. Journal of Physics: Conference Series. 2010 [cited 2012 Apr 2]. Available from: <http://stacks.iop.org/1742-6596/256/i=1/a=012003?key=crossref.b609a7d4760a2255b9659c4d95f15f7d>
89. Engine GA. Using the Datastore [Internet]. 2012. 2012 [cited 2012 Jun 25]. Available from: <https://developers.google.com/appengine/docs/python/gettingstarted/usingdatastore>
90. Chohan N, Bunch C, Pang S. AppScale : Scalable and Open AppEngine Application Development and Deployment. 2009; Available from:

- [Internet]. [cited 2012 Feb 3]. Available from:
<https://developers.google.com/appengine/docs/python/config/cron>
93. Azure W. Developer Center [Internet]. 2012. 2012 [cited 2012 Sep 26]. Available from: <https://www.windowsazure.com/en-us/develop/overview/>
 94. Engine GA. Local Unit Testing for Java [Internet]. [cited 2012 Jun 13]. Available from: <https://developers.google.com/appengine/docs/java/tools/localunittesting>
 95. Amazon. Amazon Web Services [Internet]. 2012. [cited 2012 Dec 18]. Available from: <http://aws.amazon.com/pt/>
 96. Blog AWS. New EC2 Features: Static IP Addresses, Availability Zones, and User Selectable Kernels [Internet]. 2008 [cited 2011 Nov 23]. Available from: <http://aws.typepad.com/aws/2008/03/new-ec2-feature.html>
 97. Documentation A. What is Amazon Simple Queue Service? [Internet]. Amazon. 2011 [cited 2012 Jul 20]. Available from: <http://docs.amazonwebservices.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/Welcome.html?r=5162>
 98. Documentation A. Introduction to Amazon CloudWatch [Internet]. 2010 [cited 2012 Mar 7]. Available from: http://docs.amazonwebservices.com/AmazonCloudWatch/latest/DeveloperGuide/CloudWatch_Introduction.html
 99. Leng D. Scalable Execution of LIGO Astrophysics Application in a Cloud Computing Environment [Internet]. The Department of Computer Science and Software Engineering The University of Melbourne, Australia; 2012. p. 46. Available from: <http://www.cloudbus.org/students/DougLeng2011.pdf>
 100. Documentation A. Elastic Load Balancing Developer Guide [Internet]. 2011. 2011 [cited 2012 Apr 20]. Available from: <http://docs.amazonwebservices.com/ElasticLoadBalancing/latest/DeveloperGuide/Welcome.html?r=7735>
 101. Guide GS. Amazon CloudWatch. 2010; Available from: <http://aws.amazon.com/pt/cloudwatch/>
 102. Oracle. Oracle Solaris 11 [Internet]. 2012. [cited 2012 Dec 17]. Available from: <http://hub.opensolaris.org/bin/view/Main/>
 103. Computing J smart. White Papers [Internet]. 2012. 2012 [cited 2011 Nov 23]. Available from: <http://www.joyent.com/resources/white-papers/>
 104. technizine. Joyent tem um produto novo e novo CEO [Internet]. 2012. [cited 2012 Dec 18]. Available from: <http://pt.technizine.com/page/joyent-has-a-new-product-and-new-ceo.html>

-
105. Nurmi D, Wolski R, Grzegorzczak C, Obertelli G, Soman S, Youseff L, et al. The Eucalyptus Open-source Cloud-computing System [Internet]. Computer Science Department University of California. [cited 2012 Dec 27]. Available from: <http://open.eucalyptus.com/documents/ccgrid2009.pdf>
 106. Microsoft. Windows Azure - A rock-solid cloud platform for blue-sky thinking [Internet]. 2012. [cited 2012 Dec 17]. Available from: <https://www.windowsazure.com/en-us/>
 107. Microsoft MSDN. SOAP, REST e muito mais [Internet]. 2012. 2012 [cited 2012 May 22]. Available from: <http://msdn.microsoft.com/pt-br/magazine/dd942839.aspx>
 108. Tarandeep Singh. REST vs. SOAP – The Right WebService [Internet]. 2009. 2009 [cited 2012 Apr 2]. Available from: <http://geeknizer.com/rest-vs-soap-using-http-choosing-the-right-webservice-protocol/>
 109. Laing B. Summary of Windows Azure Service Disruption [Internet]. 2009. 2009 [cited 2012 Jun 28]. Available from: <http://blogs.msdn.com/b/windowsazure/archive/2012/03/09/summary-of-windows-azure-service-disruption-on-feb-29th-2012.aspx>
 110. Chappell D. Introducing The Windows Azure Platform. October [Internet]. 2010;(October). Available from: http://www.davidchappell.com/writing/white_papers/Introducing_the_Windows_Azure_Platform,_v1.4--Chappell.pdf
 111. Chambers D. Columbus State University D . Abbott Turner College of Business and Computer Science The Graduate Program in Applied Computer Science Windows Azure : Using Windows Azure ' s Service Bus to Solve Data Security Issues Applied Computer Science Submitted in P. 2010;(July). Available from: http://archive.org/stream/windowsazureusin00cham/windowsazureusin00cham_djvu.txt
 112. Azure W. Armazenamento [Internet]. 2012. 2012 [cited 2011 Nov 6]. Available from: <http://www.windowsazure.com/pt-br/home/features/storage/>
 113. Joyeur. A Bit More About the New Joyent Cloud API [Internet]. 2012 [cited 2012 Mar 4]. Available from: <http://joyeur.com/2012/01/03/a-bit-more-about-the-new-joyent-cloud-api/>
 114. Intel. Joyent SmartDataCenter* Reference Architecture [Internet]. 2010 [cited 2012 Jan 1]. Available from: <http://www.intel.com.br/content/www/br/pt/cloud-computing/cloud-computing-joyent-smartdatacenter-architecture.html>
 115. Systems E. 3.0.1 Usage Guide 2012. Image (Rochester, N.Y.) [Internet]. 2012;1–62. Available from: [ftp://ftp.pku.edu.cn/open/cloud/Eucalyptus/Usage Guide.pdf](ftp://ftp.pku.edu.cn/open/cloud/Eucalyptus/Usage%20Guide.pdf)

116. Microsoft. Windows Azure - Unlimited Possibilities [Internet]. 2012 [cited 2012 Jul 18]. Available from: <https://www.windowsazure.com/en-us/home/features/what-is-windows-azure/>
117. Microsoft. Contratos de Nível de Serviço [Internet]. 2012. 2012 [cited 2012 May 6]. Available from: <https://www.windowsazure.com/pt-br/support/legal/sla/>
118. Microsoft. Learn About Windows Azure Marketplace [Internet]. 2012. 2012 [cited 2012 Jan 18]. Available from: <https://datamarket.azure.com/about>
119. Russinovich M. Inside Windows Azure: the cloud operating system [Internet]. 2011. 2011 [cited 2012 Aug 22]. Available from: <http://channel9.msdn.com/Events/build/build2011/sac-853t>
120. Microsoft. White Papers on Windows Azure [Internet]. 2012. 2012 [cited 2012 Aug 22]. Available from: <https://www.windowsazure.com/en-us/develop/net/other-resources/white-papers/>
121. Shankarnarayan Chandrika. Monitoring and Managing Your Windows Azure Applications and Services [Internet]. 2012. 2012 [cited 2012 Aug 22]. Available from: <http://channel9.msdn.com/Events/TechEd/Europe/2012/AZR319>
122. Russinovich M. Introduction to Windows Azure: the cloud operating system [Internet]. 2011. 2011 [cited 2012 Aug 22]. Available from: <http://channel9.msdn.com/Events/build/build2011/sac-853t>
123. Janous B. Microsoft's Energy and Efficiency Evolution to Power the Cloud [Internet]. 2012. 2012 [cited 2012 Sep 27]. Available from: <http://www.globalfoundationservices.com/posts/2012/september/12/microsofts-energy-and-efficiency-evolution-to-power-the-cloud.aspx>
124. Gauthier D. Modular Data Centers [Internet]. 2012. 2012 [cited 2012 Sep 27]. Available from: <http://www.globalfoundationservices.com/modular-data-centers.aspx>
125. Microsoft. Sign in to your Microsoft account [Internet]. 2012. 2012 [cited 2012 Dec 17]. Available from: <https://login.live.com/login.srf?cbcxt=azu&vv=910&lc=1033&wa=wsignin1.0&wtrealm=urn:federation:MicrosoftOnline&wctx=wa=wsignin1.0&rpsnv=2&ct=1355748278&rver=6.1.6206.0&wp=SAPI&wreply=https:%2F%2Faccount.windowsazure.com%2FSubscriptions&lc=1033&id=500867&cbcxt=azu>
126. Microsoft. Using CDN for Windows Azure [Internet]. 2012 [cited 2012 Jun 15]. Available from: <https://www.windowsazure.com/en-us/develop/net/common-tasks/cdn/>
127. Sanchez F. Windows Azure – Recursos de Rede [Internet]. 2012 [cited 2012 May 25]. Available from: <http://fabriciosanchez.com.br/2/windows-azure-recursos-de-rede-azure-connect-traffic-manager/>

128. Microsoft. Workflow Services in Azure AppFabric Overview [Internet]. 2012 [cited 2012 May 23]. Available from: <http://msdn.microsoft.com/en-us/library/windowsazure/hh239639.aspx>
129. Microsoft. Windows Identity Foundation Simplifies User Access for Developers [Internet]. 2012. 2012 [cited 2012 Feb 4]. Available from: <http://msdn.microsoft.com/en-us/security/aa570351.aspx>
130. Microsoft. Windows CardSpace [Internet]. 2012 [cited 2012 Jun 22]. Available from: <http://windows.microsoft.com/en-us/windows-vista/Windows-CardSpace>
131. Microsoft. Windows Azure SQL Database [Internet]. 2012 [cited 2012 Jun 24]. Available from: <http://msdn.microsoft.com/en-us/library/windowsazure/ee336279.aspx>
132. Microsoft. Understanding Cloud Storage [Internet]. 2012. 2012 [cited 2012 Aug 28]. Available from: <https://www.windowsazure.com/en-us/develop/net/fundamentals/cloud-storage/#drives>
133. Microsoft MSDN. Understanding Cloud Storage [Internet]. 2012. 2012 [cited 2012 Aug 25]. Available from: <https://www.windowsazure.com/en-us/develop/net/fundamentals/cloud-storage/#sql>
134. Sabnis M. Managing Structured Data using Windows Azure Table Service [Internet]. 2012. 2012 [cited 2012 Aug 29]. Available from: <http://www.dotnetcurry.com/ShowArticle.aspx?ID=809>
135. Cremers R. Everything you need to know about Windows Azure Table Storage to use a scalable non-relational structured data store [Internet]. 2012. 2012 [cited 2012 Aug 29]. Available from: <http://robbincremers.me/2012/03/01/everything-you-need-to-know-about-windows-azure-table-storage-to-use-a-scalable-non-relational-structured-data-store/>
136. Cremers R. Everything you need to know about Windows Azure Blob Storage including permissions, signatures, concurrency [Internet]. 2012. 2012 [cited 2012 Aug 25]. Available from: <http://robbincremers.me/2012/02/27/everything-you-need-to-know-about-windows-azure-blob-storage/>
137. Storage WA. Windows Azure Storage Abstractions and their Scalability Targets [Internet]. 2010. 2010 [cited 2012 Aug 29]. Available from: <http://blogs.msdn.com/b/windowsazurestorage/archive/2010/05/10/windows-azure-storage-abstractions-and-their-scalability-targets.aspx>
138. Microsoft MSDN. Optimistic Concurrency [Internet]. 2012. 2012 [cited 2012 Dec 27]. Available from: [http://msdn.microsoft.com/pt-br/library/aa0416cz\(v=vs.71\).aspx](http://msdn.microsoft.com/pt-br/library/aa0416cz(v=vs.71).aspx)
139. Storage WA. Windows Azure Storage Architecture Overview [Internet]. 2010. 2010 [cited 2012 Aug 29]. Available from:

- <http://blogs.msdn.com/b/windowsazurestorage/archive/2010/12/30/windows-azure-storage-architecture-overview.aspx>
140. Microsoft. Windows Azure Marketplace [Internet]. 2012. 2012 [cited 2012 May 12]. Available from: <https://www.windowsazure.com/en-us/home/features/marketplace/>
 141. Microsoft MSDN. Overview of Building an Application that Runs in a Hosted Service [Internet]. 2011. 2011 [cited 2011 Sep 10]. Available from: <http://msdn.microsoft.com/en-us/library/windowsazure/gg433030.aspx>
 142. Microsoft MSDN. Queues, Topics, and Subscriptions [Internet]. 2012. 2012 [cited 2012 Sep 16]. Available from: <http://msdn.microsoft.com/en-us/library/windowsazure/hh367516.aspx>
 143. Valery Mizonov SM. Windows Azure Queues and Windows Azure Service Bus Queues - Compared and Contrasted [Internet]. 2011. 2011 [cited 2012 Sep 16]. Available from: <http://msdn.microsoft.com/en-us/library/windowsazure/hh767287.aspx>
 144. Valery Mizonov SM. Windows Azure Queues and Windows Azure Service Bus Queues - Compared and Contrasted [Internet]. 2011. 2011 [cited 2012 Dec 27]. Available from: <http://msdn.microsoft.com/en-us/library/windowsazure/hh767287.aspx>
 145. Cremers R. Everything you need to know about windows azure queue storage to build disconnected and reliable systems [Internet]. 2012. 2012 [cited 2012 Sep 18]. Available from: <http://robbincremers.me/2012/03/07/everything-you-need-to-know-about-windows-azure-queue-storage-to-build-disconnected-and-reliable-systems-2/>
 146. Haridas J. Windows Azure Queues: Improved Leases, Progress Tracking, and Scheduling of Future Work [Internet]. 2011. 2011 [cited 2012 Sep 18]. Available from: <http://blogs.msdn.com/269837/ProfileUrlRedirect.ashx>
 147. Microsoft MSDN. Autoscaling and Windows Azure [Internet]. 2012. 2012 [cited 2012 Sep 17]. Available from: [http://msdn.microsoft.com/en-us/library/hh680945\(v=pandp.50\).aspx](http://msdn.microsoft.com/en-us/library/hh680945(v=pandp.50).aspx)
 148. Azure W. How to Use the Autoscaling Application Block [Internet]. 2012. 2012 [cited 2012 Sep 10]. Available from: <https://www.windowsazure.com/en-us/develop/net/how-to-guides/autoscaling/#CreateHost>
 149. Melnik G. Autoscaling Windows Azure applications [Internet]. 2011. 2011 [cited 2012 Sep 10]. Available from: <http://channel9.msdn.com/posts/Autoscaling-Windows-Azure-applications>
 150. Acer. Acer e-Ticketing Solution [Internet]. [cited 2012 Dec 2]. Available from: http://www.acer.net/e-ticketing/en_index.jsp

151. Link. SmartCITIES [Internet]. 2010. 2010 [cited 2011 Dec 12]. Available from: <http://www.link.pt/upl/%7B15b23eb4-bec1-4966-a29a-ff7c7fa67644%7D.pdf>
152. Microsoft Azure. .NET Developer Center [Internet]. 2012. 2012 [cited 2012 Sep 29]. Available from: <https://www.windowsazure.com/en-us/develop/net/>
153. Azure M. How to Use Service Bus Topics/Subscriptions [Internet]. 2012. 2012 [cited 2012 Sep 18]. Available from: <https://www.windowsazure.com/en-us/develop/net/how-to-guides/service-bus-topics/#header-8>
154. Cremers R. Everything you need to know about Windows Azure queue storage to build disconnected and reliable systems [Internet]. 2012. 2012 [cited 2012 Sep 30]. Available from: <http://robbincremers.me/2012/03/07/everything-you-need-to-know-about-windows-azure-queue-storage-to-build-disconnected-and-reliable-systems-2/>
155. Microsoft. Pricing Details [Internet]. 2012. 2012 [cited 2012 Sep 17]. Available from: <https://www.windowsazure.com/en-us/pricing/details/>
156. Azure W. How to Use the Table Storage Service [Internet]. 2012. 2012 [cited 2012 Sep 3]. Available from: <https://www.windowsazure.com/en-us/develop/net/how-to-guides/table-services/>
157. Azure W. Windows Azure Service Bus Quotas [Internet]. 2012. 2012 [cited 2012 Jun 19]. Available from: <http://msdn.microsoft.com/en-us/library/windowsazure/ee732538.aspx>
158. Shawn Hernan, Scott Lambert, Tomasz Ostwald AS. Uncover Security Design Flaws Using The STRIDE Approach [Internet]. 2006. 2006 [cited 2012 Sep 25]. Available from: <http://msdn.microsoft.com/en-us/magazine/cc163519.aspx>