Paulo Agostinho Rodrigues Pires
Licenciado em Engenharia Informática

Framework para a construção de "portais de negócio" para gestão de solicitações de consumidores IaaS na HP Cloud

Dissertação para obtenção do Grau de Mestre em Engenharia Informática

Orientador: Prof. Doutor Paulo Orlando Reis Afonso Lopes, Prof. Auxiliar, DI/FCT/UNL

Co-orientador: Engenheiro Informático Rui Miguel Ramalho Ramos, Team Leader de TS Consulting da HP

Júri:

Presidente: Prof. Doutor Fernando Pedro Reino da Silva Birra
Arguente: Doutor Gonçalo Filipe Maurício dos Santos Borges
Vogal: Prof. Doutor Paulo Orlando Reis Afonso Lopes

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Novembro 2013

Paulo Agostinho Rodrigues Pires
Licenciado em Engenharia Informática

Framework para a construção de "portais de negócio" para gestão de solicitações de consumidores IaaS na HP Cloud

Dissertação para obtenção do Grau de Mestre em
Engenharia Informática

Orientador: Prof. Doutor Paulo Orlando Reis Afonso Lopes, Prof. Auxiliar, Departamento de Informática da FCT/UNL

Co-orientador: Engenheiro Informático Rui Miguel Ramalho Ramos, Team Leader de TS Consulting da HP

Novembro 2013

Framework para a construção de "portais de negócio" para gestão de solicitações de consumidores IaaS na HP Cloud

Copyright © Indicação dos direitos de cópia em nome de Paulo Agostinho Rodrigues Pires, aluno de Mestrado em Engenharia Informática da FCT/UNL.

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Agradecimentos

Um profundo agradecimento a todos aqueles que, de diferentes formas, me apoiaram durante estes 5 anos, enquadrando este trabalho como objectivo último de uma licenciatura e mestrado integrado em Engenharia Informática.

Começo assim por agradecer a todos os meus professores da FCT que, ao longo destes 5 anos, contribuíram para a minha formação académica.

Aos meus amigos que no estudo, nas diversões, nos incentivos, conseguiram sempre criar um ambiente ideal e propício à concretização das diferentes etapas deste curso com éxito. Para mim foi um privilégio ter partilhado com eles esta etapa da minha vida.

À HP, pela disponibilização de todas as infra-estruturas, permissões e acessos para a elaboração deste framework. Um abiente acolhedor em pessoas, trabalho, equipamento e apoio.

Ao meu orientador e responsável na HP, Rui Miguel Ramalho Ramos (*TS Consulting Resource Manager*) que, sempre ponto a ajudar, soube resolver atempadamente todas as minhas solicitações. Um muito obrigado pela camaradagem, eficiência e profissionalismo com que encarou toda a implementação.

Um agradecimento especial ao meu orientador da tese, Professor Paulo Orlando Reis Afonso Lopes (Arquitectura de Sistemas Computacionais), pela sua orientação, disponibilidade e sugestões que me ajudaram na conclusão deste trabalho. Um muito obrigado pela sua constante presença e acima de tudo pela confiança que me concedeu durante toda a tese, quer na preparação, quer na elaboração. Uma acessibilidade e presença que fizeram toda a diferença, obrigado professor.

Por último, à minha família pelo encorajamento, pais, irmã e sogros. Um obrigado especial ao meu sogro, Aires Pereira, pelas brincadeiras que proporcionou à minha filha entretendo-a para que eu pudesse estudar e desenvolver esta tese.

À minha esposa, Nadia Pires, e a minha filha, Lara, merecem uma menção especial. Agradeço à minha esposa pelo apoio prestado nesta etapa da minha vida, obrigada pelo amor, pela paciência, pelo otimismo, pela motivação e por me ter acompanhado nas horas mais difíceis desta etapa, estarei eternamente grato. Na minha ausência, soube sempre ser o Pai e a Mãe que uma família precisa.

À minha querida filha, um agradecimento muito especial e um pedido de desculpas em simultâneo. Obrigada pelos sorrisos, pelos abracinhos, pelos beijinhos, enfim… por todos os carinhos que só uma menina doce como ela sabe dar, e que me ajudaram a ultrapassar os momentos mais difíceis desta jornada. Desculpa pela falta de paciência, e por todos os momentos que não podemos estar juntos.

Resumo

O HP CloudSystem Matrix (CSM) faz parte de uma pilha de *software* HP para computação na *cloud* que cobre todos os níveis de serviço considerados relevantes: IaaS (Infra-estrutura como Serviço), PaaS (Plataforma como Serviço) e SaaS (Software como Serviço). Apesar de ser a base desta pilha, i.e., oferecer o nível IaaS, é um produto extremamente complexo pois interage com todas as infra-estruturas: as computacionais (i.e., servidores físicos ou virtuais), as de armazenamento (do disco interno aos discos em servidores de armazenamento), e as de interligação (redes Ethernet e FC).

Apesar de toda a complexidade da infra-estrutura, real e virtual, que gere, o CSM torna conceptualmente simples a entrega aos consumidores de infra-estruturas para suporte a aplicações: 1) o *administrador* define que recursos da infra-estrutura estão disponíveis para integrar a "oferta *cloud*"; 2) o *arquitecto* define *templates* para as arquitecturas que considera adequadas para necessidades dos consumidores (e.g., arquitectura *3-tier* para uma solução ERP - *Enterprise Resource Planning*); e 3) o consumidor escolhe o *template* que melhor se ajusta às suas necessidades e efectua um pedido de aprovisionamento da infra-estrutura.

A interacção entre os diferentes interlocutores (1), (2), (3) e o CSM é fundamentalmente realizada sobre portais; contudo, especialmente no caso do consumidor, o portal disponibilizado pelo produto tem sido considerado como "complexo", por apresentar informação demasiado técnica, "rígido", por não poder ser customizado (por exemplo para suprimir a "informação demasiado técnica"), e "grosseiro" por não permitir a especificação mais fina das características da infra-estrutura que se quer aprovisionar (por exemplo, permite variar o número de CPUs e a quantidade de memória de um servidor, mas não permite escolher a tecnologia dos discos que se pretendem aprovisionar, e.g., SSD em vez de FC, 15K em vez de 10K rpm). Assim, o objectivo final da dissertação é desenvolver um *framework* que permita, com base num conjunto (extensível e configurável) de opções pré-definidas e em *layouts* customizáveis, definir portais que se integram com o HP CloudSystem Matrix e que permitam aos utilizadores (consumidores) uma interacção não só mais simples, mas também mais versátil.

Neste trabalho, são abordados os modelos de serviço e de implantação (*deployment*) de *clouds*; a virtualização (não somente de servidores, mas também de armazenamento e de redes), pedra base de toda a tecnologia *cloud*; e os módulos e APIs disponíveis para interoperar com o CSM, nomeadamente API-MOE e API-VMware. Por fim, é apresentada uma *framework* com uma arquitectura multicamada (N-tier) implementada com tecnologias padrão: TCP/IP para a pilha de comunicações, REST (Representational State Transfer) para regular a interacção e troca de informação cliente/servidor e XML (Extensible Markup Language) e JSON (JavaScript Object Notation) como formatos de dados.

Palavras-chave: Computação na nuvem, Portal de interacção, Tradução Binária, Virtualização de Servidores, Virtualização de Armazenamento e Virtualização da Rede.

Abstract

The HP CloudSystem Matrix (CSM) is a part of the HP software stack for cloud computing that covers all relevant service levels: IaaS (Infrastructure as a Service), PaaS (Platform as a Service) and SaaS (Software as a Service). Although it rests at the base of the stack, i.e., only offers the IaaS level, it is an extremely complex product since it interacts with all infrastructures: computing (i.e., physical and virtualized servers), storage (from individual drives to disk arrays), and networking (Ethernet and FC networks)

Despite all the (real) infrastructure´s complexity that CSM deals with, the delivery (to end users – a.k.a. infrastructure consumers) of infrastructures for application support, becomes a conceptually simple task: (1) the administrator defines which infrastructure resources are available to integrate the cloud offer; (2) the architect defines templates for those architectures he deems appropriate for the end users (e.g., 3-tier architecture for an ERP - Enterprise Resource Planning - solution); (3) the end user chooses the template that better suits its needs, and then submits a request for the delivery of the infrastructure.

The interaction between the different counterparts (1), (2) and (3) and the CSM is fundamentally realised over portals; however, especially in the case of the end user, the portal has been considered "complex", since it presents too much technical information, "rigid", since it is not customizable (e.g., to suppress the "too much technical information"), and "coarse", since it does not allow a finer-graned specification of the infrastructure´s attributes that are required (e.g., it allows to change the number of CPUs and the amount of memory in the server, but it does not allow one to choose the disk technologies, e.g. SSD instead of FC, or 15K instead 10K rpm). Therefore, the final goal of this dissertation is to develop a framework that allows, with a set of pre-defined settings and customizable layouts, to define portals that are integrated with HP CloudSystem Matrix and enables a simpler user interaction.

In this work we studied: service and cloud implementation models; virtualization (not only server virtualization, but also storage and network virtualization), the foundation of all cloud technologies; modules and APIs that interoperate with CSM, namely API-MOE and API-VMware. Finally, we present a framework (prototype) implemented in a multi-layer architecture (N-tier) with standard technologies: TCP/IP for the communication stack, REST (Representational State Transfer) to control the interaction and the client/server information flows and XML (eXtensible Markup Language) and JSON (JavaScript Object Notation) for the data formats.

Keywords: Cloud Computing, Interaction Portal, Binary Translation, Server Virtualization, Storage Virtualization and Network Virtualization.

Glossário e Lista de siglas

API: Application Programming Interface

BIOS: Basic Input/Output System

CC: Computação em Cloud

CIFS: Common Internet File System

CL: Command Line

CMS: Console Monitor System

CP: Control Program

CPU: Central Processing Unit

CSS: Cascading Style Sheets

DAS: Direct-Attached Storage

Dom0: Domínio 0 do XenServer

DomU: Domínio U do XenServer

DVFilter: Distributed Virtual Filter

DynaRecs: Dynamic Recompilation

EIDE: Enhanced Integrated Drive Electronics

FC: Fibre Channel

FCoE: Fibre Channel over Ethernet

FTP: File Transfer Protocol

GUEST: Sistema Virtualizado

HD: Hard Drive

HOST: Servidor Hospedeiro

HP: Hewlett-Packard

HPCMS: HP Console Management System

CSM: HP CloudSystem Matrix (Hewlett-Packard)

IC: Insight Control (Hewlett-Packard)

ID: Insight Dynamics (Hewlett-Packard)

IO: Infrastructure Orchestration (Hewlett-Packard)

SA: Server Automation (Hewlett-Packard)

SS: SiteScope (Hewlett-Packard)

UD: Universal Discovery (Hewlett-Packard)

HPVM: HP Integrity Virtual Machine

VMAN: Virtualization Manager (Hewlett-Packard)

VMMGR: HP Integrity Virtual Machines Manager (Hewlett-Packard)

HTTP: HyperText Transfer Protocol

HVM: Virtualização assistida por hardware

I/O: Input/Output

IaaS: Infrastructure as a Service

IBM: International Business Machines Corporation

IDE: Integrated Development Environment

iSCSI: Internet Small Computer System Interface

J2EE: Java2 Platform Enterprise Edition

JS: JavaScript

JSON: JavaScript Object Notation

JVM: Java Virtual Machine

LAN: Local Area Network

LUN: Logical Unit Number

LVM: Logical Volume Manager

MAC: Apple Macintosh

MOE: Matrix Operating Environment

MSCS: Microsoft Cluster Server

NAS: Network Attached Storage

nbAPI: Northbound API

NetApp: Network Appliance

NFS: Network File System

NIC: Adaptador de rede (Network Interface Controller)

NVRAM: Non-Volatile Random Access Memory

PaaS: Plataform as a Service

PATA: Parallel Advanced Technology Attachment

pNIC: Physical Network Interface Controller

POSIX: Portable Operating System Interface

PV: Para-Virtualização

PVHVM: Para-Virtualização + Virtualização assistida por hardware

QOS: Quality Of Service

RAID: Redundant Array of Independent Disks

RAM: Random Access Memory

REST: REpresentational State Transfer

RPC: Remote Procedure Call

SaaS: Software as a Service

SAN: Storage Area Network

SAP: Sistemas, Aplicações e Produtos

SAS: Serial Attached SCSI

SATA: Serial ATA

SCSI: Small Computer System Interface

SCVMM: System Center Virtual Machine Manager

SF: Sistema de Ficheiros

SLA: Service Level Agreement

SMTP: Simple Mail Transfer Protocol

SO: Sistema Operativo

SOA: Service-Oriented Architecture

SOAP: Simple Object Access Protocol

SSD: Solid-State Drive

TCP/IP: Transmission Control Protocol/Internet Protocol

TI: Tecnologias de Informação

URI: Uniform Resource Identifier

URL: Uniform Resource Locator

UUID: Identificador Único Universal

VDI: Virtual Disk Image

vDS: vNetwork Distributed Switch

VHD: Virtual Hard Disk (Microsoft)

VLAN: Virtual LAN

VM: Virtual Machine

VMBus: Virtual Machine Buss

VMDK: Virtual Machine Disk (VMware)

VMFS: Virtual Machine File System

VMM: hipervisor (também designado Virtual Machine Manager)

vNIC: Virtual Network Interface Controller

VSC: Virtual Service Client

VSP: Virtual Service Provider

vSwitch: Virtual Switch

WOA: Web-Oriented Architecture

WS: Web Service

WWW: World Wide Web

XML: Extensible Markup Language

Índice

1. Introdução 1

1.1 Descrição e contexto 1

1.2 Objectivos 3

1.3 Organização 3

1.4 Contribuições 4

1.5 Resumo 5

2. Virtualização 7

2.1 Evolução histórica 7

2.2 Virtualização de Servidores 8

2.2.1 VMware (VMware, Inc.) 9

2.2.2 Xen (Citrix Systems, Inc.) 10

2.2.3 Integrity VM (HP) 11

2.2.4 Hyper-V (Microsoft) 12

2.3 Virtualização do Armazenamento 13

2.3.1 Armazenamento em ambientes virtualizados 15

2.4 Virtualização da Rede 17

2.5 Resumo 19

3. Computação na *Cloud* 21

3.1 Perímetro de uma *cloud* 21

3.2 Modelos de serviço 22

3.3 Resumo 24

4. HP Cloud Stack 25

4.1 HP CloudSystem Matrix 25

4.2 Modelos de interacção com o HP CSM 27

4.3 Papéis e Portais de interacção 28

4.4 Conceitos fundamentais da plataforma HP CSM 30

4.5 Limitações do HP CloudSystem Matrix 32

4.6 Resumo 33

5. O framework e a sua Realização 35

5.1 Arquitectura 35

5.2 Integração da framework no HP CSM (MOE e VMware) 36

5.3 Um desenho guiado pela Interface Utilizador 37

5.4 Camada de apresentação 41

5.4.1 Do lado do cliente 41
5.4.2 Do lado do servidor 42
5.4.3 Lista de primitivas na óptica dos recursos 43
5.5 Camada Serviços/Lógica 44
5.5.1 Camada de Serviços 45
5.5.2 Camada Lógica 45
5.6 Camada de Dados 71
5.6.1 Dados não persistentes 71
5.6.2 Dados persistentes 72
5.7 Feedback 75
5.8 Questões principais 76
5.9 Visão global 78
5.10 Resumo 79
6. Considerações Finais 81
6.1 Conclusões 82
6.2 Trabalho futuro 83
6.3 Resumo 84
Referências bibliográficas 85

Índice de Figuras

Figura 1 - Ambiente não Virtualizado 1
Figura 2 - Ambiente Virtualizado 1
Figura 3 - Infra-estrutura CSM 2
Figura 4 - Arquitectura VMware ESXi (Tipo-1) 10
Figura 5 - Arquitectura XenServer 11
Figura 6 - Arquitectura Hyper-V 12
Figura 7 - Redundant Array Independent Disks (RAID) 13
Figura 8 - Storage Area Network (SAN) 14
Figura 9 - Network Attached Storage (NAS) 14
Figura 10 - Datastore e VMs 15
Figura 11 - Sistema de Ficheiros VMFS 16
Figura 12 - Network File System (NFS) 16
Figura 13 - Infra-estrutura de rede tradicional 17
Figura 14 - Infra-estrutura de rede virtual 18
Figura 15 - vNetwork Distributed Switch – vDS 18
Figura 16 - Distributed Virtual Filter (DVFilter) 19
Figura 17 - Relação entre serviços e responsabilidades 23
Figura 18 - Relações e papéis dos envolventes nos modelos de Cloud Computing 24
Figura 19 - Contributos CloudStack. 25
Figura 20 - Desenho, publicação e implementação do catálogo de serviços. 26
Figura 21 - Infra-estrutura HP CloudSystem Matrix 27
Figura 22 - Utilizadores de TI em cloud e respectivos papéis (roles) 28
Figura 23 - Portal do Administrador 29
Figura 24 - Portal do Arquitecto 29
Figura 25 - Portal do Utilizador final 30
Figura 26 - Arquitectura n-tier 35
Figura 27 - Relação dos VMIDs (WebService-MOE-vCenter) 36
Figura 28 - Estrutura em árvore do menu lateral esquerdo 38
Figura 29 - Portal 39
Figura 30 - Diagrama de Sequência (Criação de um Serviço) 40
Figura 31 - Obtenção de recursos através da API RESTfull 44
Figura 32 - Execução tab “Template” 48
Figura 33 - Execução tab “Service” 51
Figura 34 - Execução tab “ServerGroups” 54
Figura 35 - Execução tab “Server” 59

Figura 36 - Execução tab “Disks” 63
Figura 37 - Execução tab “Software” 64
Figura 38 - Execução tab “Interface” 65
Figura 39 - Listagem das redes existentes 67
Figura 40 - Execução tab “Snapshot” 68
Figura 41 - Execução tab “Requests” 69
Figura 42 - Execução tab “ServerPools” 71
Figura 43 - Realm JAAS (Java Authentication and Authorization Service) 72
Figura 44 - Modelo ER 74
Figura 45 - Mensagens SOAP na invocação de processos do MOE 78
Figura 46 - Obtenção de recursos através da API RESTfull do WEBSERVICE 79
Figura 47 - VMM ou hypervisor do Tipo-2 93
Figura 48 - VMM ou hypervisor do “híbrido” 94
Figura 49 - VMM ou hypervisor do Tipo-1 94
Figura 50 - Emulação de Hardware 95
Figura 51 - Virtualização Completa 96
Figura 52 - Para-Virtualização 97
Figura 53 - Virtualização assistida por Hardware 97
Figura 54 - LUN num SAN iSCSI ou SAN FC 109
Figura 55 - Network File System (NFS) 110
Figura 56 - Projecto “Dynamic Web Project” no eclipse 113
Figura 57 - Plataforma Java 119
Figura 58 - Comando “java -version” 120
Figura 59 - Instalação da Plataforma Java 121
Figura 60 - Variável de sistema “Path” 121
Figura 61 - Variável de sistema “CLASSPATH” 122
Figura 62 - Variável de sistema “JAVA_HOME” 122
Figura 63 - Directoria de instalação do Servidor Apache Tomcat 123
Figura 64 - Link localhost Apache Tomcat 124
Figura 65 - Add repository 125
Figura 66 - Dynamic Web Project 126
Figura 68 - Propriedades do Dynamic Web Project 127
Figura 69 - Web.xml do Dynamic Web Project 127
Figura 70 - Directoria base do Dynamic Web Project 128
Figura 71 - Criar uma classe 129
Figura 72 - Servidor Apache Tomcat dentro do Eclipse 130
Figura 73 - Execução no Browser 130

Figura 74 - Projecto base MOE....131
Figura 75 - Importação do projecto real MOE....131
Figura 76 - Projecto real MOE....132
Figura 77 - Ficheiro “adminUsersList.txt” 133
Figura 78 - Ficheiro “hp-io-v5.wsdl”....133
Figura 79 - WSDL do HPMOE....133
Figura 80 - Ficheiro “login.config”....134
Figura 81 - Ficheiro “login.config”....134
Figura 82 - Ficheiro “vmwareAcess.txt” 135
Figura 83 - Servidor Apache Tomcat dentro do Eclipse....135
Figura 84 - Compilação do projecto MOE....136
Figura 85 - Login page MOE....136
Figura 86 - Data page MOE....136
Figura 87 - Detalhes gerais do Template “Template_3srv”....137
Figura 88 - Detalhes do ServerGroup “SvrGrp1” do Template “Template_3srv” 138
Figura 89 - Detalhes do Servidor “SvrGrp1-1” 138
Figura 90 - Detalhes do Servidor “SvrGrp1-1” 139
Figura 91 - Detalhes do Disco “Disk1-1” do Servidor “SvrGrp1-1”....139
Figura 92 - Detalhes do Software “Windows 7” do Servidor “SvrGrp1-1” 139
Figura 93 - Detalhes da/s Interfaces “NIC0” e “NIC1” do Servidor “SvrGrp1-1”....140
Figura 94 - Criação do serviço “myService” 140
Figura 95 - Lista de pedidos do utilizador 141
Figura 96 - Eventual cancelamento de um pedido....141
Figura 97 - Serviço “myService” criado....142
Figura 98 - Listagem dos acessos remotos....142
Figura 99 - Acesso remoto na tab “Server” 143
Figura 100 - Acesso remoto por RDP....143
Figura 101 - Acesso remoto por Telnet 143
Figura 102 - Acesso remoto por VMRC....143
Figura 103 - Menu lateral esquerdo “MOE Help”....145
Figura 104 - Menu lateral esquerdo “MOE Help”....145
Figura 105 - “Import xml” help 146
Figura 106 - Template info Tab 147
Figura 107 - Service info Tab 147
Figura 108 - ServerGroup info Tab....148
Figura 109 - Server info Tab....148
Figura 110 - Disk info Tab....149

Figura 111 - Software info Tab 149
Figura 112 - Interface info Tab 150
Figura 113 - Request info Tab 150
Figura 114 - About info Tab 151
Figura 115 - Team info Tab 151
Figura 116 - Javadoc info Tab 152
Figura 117 - “Maps” da relação Host / VM 153
Figura 118 - “Maps” da relação Host / Network 154
Figura 119 - “Maps” da relação Host / Datastore 155
Figura 120 - “Maps” da relação VM / Network 156
Figura 121 - “Maps” da relação Host / Datastore 157
Figura 122 - Detalhes gerais do Template “Template_3srv” 159
Figura 123 - Detalhes do ServerGroup “SvrGrp1” do Template “Template_3srv” 159
Figura 124 - Detalhes do Servidor “SvrGrp1-1” 160
Figura 125 - Detalhes dos Custos do Servidor “SvrGrp1-1” 160
Figura 126 - Detalhes do Disco “Disk1-1” do Servidor “SvrGrp1-1” 160
Figura 127 - Detalhes do Software “Windows 7” do Servidor “SvrGrp1-1” 160
Figura 128 - Detalhes da/s Interfaces “NIC0” e “NIC1” do Servidor “SvrGrp1-1” 161
Figura 129 - Ficheiro “TemplateList.xml” 161
Figura 130 - Ficheiro “Template_3srv.xml” 162
Figura 131 - Importação de um Template em formato XML 162
Figura 132 - InfoWindow “Delete Template” 162
Figura 133 - InfoWindow com a visualização do Template em XML 163
Figura 134 - Ficheiro “Template_3srv_ServiceList.xml” 163
Figura 135 - Criação de um Serviço com base no Template “Template_3srvl” 164
Figura 136 - Lista de todos os pedidos do utilizador 164
Figura 137 - Eventual cancelamento de um pedido 165
Figura 138 - Lista dos pedidos resultantes da relação Template/utilizador 165
Figura 139 - Detalhes do Serviço “myService” do Template “Template_3srv” 166
Figura 140 - Ficheiro “allServicesList.xml” 166
Figura 141 - Ficheiro “myService.xml” 167
Figura 142 - InfoWindow com a visualização do Serviço em XML 167
Figura 143 - Remoção de um Serviço 167
Figura 144 - Power ON de um Serviço 168
Figura 145 - Power OFF de um Serviço 168
Figura 146 - Power Cycle de um Serviço 168
Figura 147 - Activação de um Serviço 168

Figura 148 - Desactivação de um Serviço 168
Figura 149 - Lista dos Service Actions 169
Figura 150 - Lista dos Acessos Remotos do um Serviço 169
Figura 151 - Acesso remoto por RDP 169
Figura 152 - Acesso remoto por Telnet 169
Figura 153 - Acesso remoto por VMRC 170
Figura 154 - Listagem dos pedidos por Serviço 170
Figura 155 - Detalhes do Grupo de Servidores “SvrGrp1” do Serviço “myService” 171
Figura 156 - Ficheiro “SvrGrp1.xml” 171
Figura 157 - InfoWindow com a visualização do Grupo de Servidores em XML 172
Figura 158 - Adição de Servidores ao Grupo de Servidores 172
Figura 159 - Remoção de Servidores ao Grupo de Servidores 172
Figura 160 - Edição de Servidores ao Grupo de Servidores 173
Figura 161 - Power ON de um ServerGroup 173
Figura 162 - Power OFF de um ServerGroup 173
Figura 163 - Power Cycle de um ServerGroup 173
Figura 164 - Activação de um ServerGroup 174
Figura 165 - Desactivação de um ServerGroup 174
Figura 166 - Listagem de todos os Snapshots existentes no Grupo de Servidores 174
Figura 167 - Acções Snapshot/Servidor 174
Figura 168 - Revert Snapshot 174
Figura 169 - Delete Snapshot 175
Figura 170 - Create Snapshot 175
Figura 171 - Lista de “Stereotype Disks” 175
Figura 172 - Adicionar um disco virtual “Disk62” ao Grupo de Servidores 175
Figura 173 - Adicionar um disco físico “Disk788” ao Grupo de Servidores 176
Figura 174 - Detalhes do Servidor “SvrGrp1-1” do Serviço “myService” 176
Figura 175 - Ficheiro “hostmygostr01_ls.xml” 177
Figura 176 - InfoWindow com a visualização do Servidor em XML 177
Figura 177 - Remoção de um Servidor 178
Figura 178 - Edição de um Servidor 178
Figura 179 - Power ON de um Servidor 178
Figura 180 - Power OFF de um Servidor 178
Figura 181 - Power Cycle de um Servidor 179
Figura 182 - Activação de um Servidor 179
Figura 183 - Desactivação de um Servidor 179
Figura 184 - Create Snapshot 179

Figura 185 - Pools de Servidores 180
Figura 186 - Criar uma Pools de Servidores 180
Figura 187 - Remoção de uma Pools de Servidores 180
Figura 188 - InfoWindo de aviso ao administrador 180
Figura 189 - Informação dos hosts físicos 181
Figura 190 - Sessões Remotas num Servidor 181
Figura 191 - Detalhes do Disco “Disk1-1” do Servidor “SvrGrp1-1” 182
Figura 192 - Criação de um “Stereotype Disks” 182
Figura 193 - Detalhes do Software “Windows 7” do Servidor “SvrGrp1-1” 183
Figura 194 - Lista de Software 183
Figura 195 - Detalhes do Software 184
Figura 196 - Detalhes do Interface NIC0” do Servidor “SvrGrp1-1” 184
Figura 197 - Listagem de todas as redes 185
Figura 198 - Detalhe da rede 185
Figura 199 - Adição de um IP flutuante à Interface “NIC0” 185
Figura 200 - Remoção de um IP flutuante pertencente à Interface “NIC0” 186
Figura 201 - Mover um IP flutuante para a Interface “NIC0” 186

Índice de Tabelas

Tabela 1 - Funções associadas à inicialização do Webservice 46
Tabela 2 - Funções associadas ao Template 50
Tabela 3 - Funções associadas ao Serviço 53
Tabela 4 - Funções associadas ao Grupo de Servidores 57
Tabela 5 - Funções associadas aos Servidores 61
Tabela 6 - Funções associadas aos Discos 63
Tabela 7 - Funções associadas ao Software 65
Tabela 8 - Funções associadas às Interfaces 66
Tabela 9 - Funções associadas às Redes 68
Tabela 10 - Funções associadas aos Snapshots 68
Tabela 11 - Funções associadas aos Requests 70
Tabela 12 - Funções associadas às Pools de Servidores 71
Tabela 13 - Dados gerais VMWARE 99
Tabela 14 - Detalhes VMWARE 99
Tabela 15 - Dados gerais Citrix 101
Tabela 16 - Dados detalhados Citrix 101
Tabela 17 - Dados gerais HP 103
Tabela 18 - Dados detalhados HP 103
Tabela 19 - Dados gerais Microsoft 105
Tabela 20 - Dados detalhados Microsoft 105

Índice de Anexos

Anexo A (Virtualização: Vantagens e inconvenientes) 89
Anexo B (Virtualização: Arquitectura) 93
Anexo C (Virtualização: Tipos de Implementação) 95
Anexo D (Detalhes da plataforma e restrições da VMware) 99
Anexo E (Detalhes da plataforma e restrições da Citrix) 101
Anexo F (Detalhes da plataforma e restrições da HP) 103
Anexo G (Detalhes da plataforma e restrições da Microsoft) 105
Anexo H (Ambiente de armazenamento VMware ESXi e Citrix Xen) 107
Anexo I (Uma exemplificação: “Dynamic Web Project”) 113
Anexo J (Ferramentas de desenvolvimento WebService) 119
Anexo K (Criar e executar um “Dynamic Web Project” no Eclipse) 126
Anexo L (Importar, configurar e executar o RESTWEBSERVICE - MOE) 131
Anexo M (Criação de um Serviço e ligação à/s VM/s) 137
Anexo N (MOE Help) 145
Anexo O (VMware - Virtual Machine Resources) 153
Anexo P (Scenario view) 159

1. Introdução

1.1 Descrição e contexto

A evolução do poder de processamento dos computadores actuais é notória nestes últimos anos. Infelizmente, uma grande parte desse poder não é efectivamente usada no quotidiano, havendo por isso uma crescente preocupação com os desperdícios que resultam dessa subutilização. Tecnologias como a virtualização, em primeiro lugar, e a *Cloud Computing*, como estágio mais avançado desta, são formas de redução de tais desperdícios.

Virtualização

Observando as duas primeiras figuras, rapidamente ficamos com a ideia do que é real, figura 1 e do que é virtual, figura 2. Assim, entenda-se o real como algo com características físicas, concretas, palpáveis e o virtual associado a algo simulado, abstracto, uma imitação do real. É aqui que se enquadra a virtualização, o simular de um ambiente real que suporta todo um ambiente que vai do sistema de operação às aplicações sem necessidade de acesso à máquina real, que neste caso funciona como hospedeira desta tecnologia.

A virtualização encaixa-se num mundo ilusório de várias máquinas virtuais (Virtual Machines - VMs) independentes e isoladas entre si, cada uma com um Sistema Operativo (SO) próprio e potencialmente diferente do SO das demais.

Figura 1 - Ambiente não Virtualizado

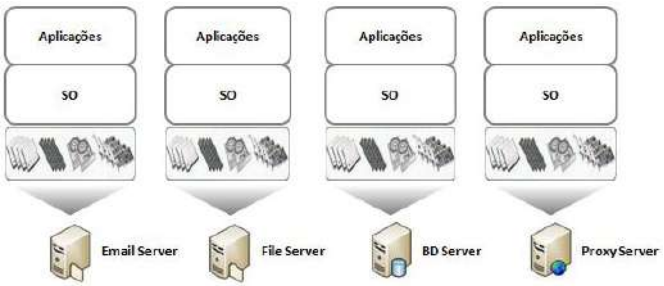
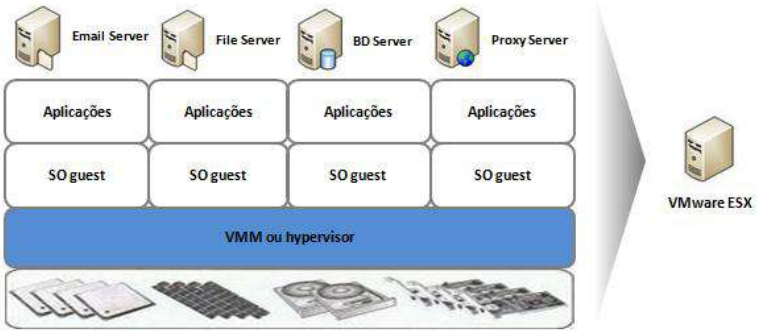


Figura 2 - Ambiente Virtualizado



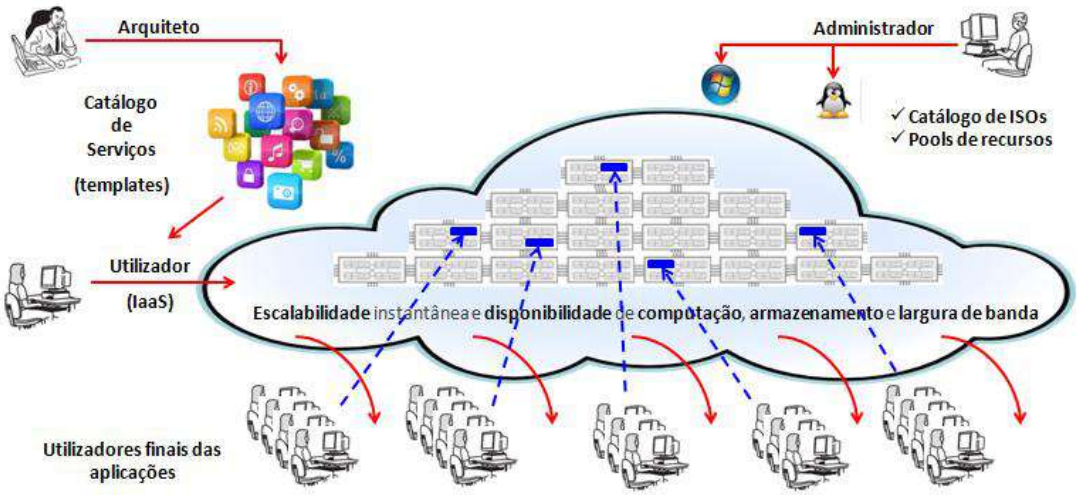
Cloud Computing

As estratégias de consolidação de servidores propiciadas pelo uso de virtualização conduzem, de facto, à redução de custos de *hardware*, *software*, e energia, mas não evitam (e podem até aumentar) o número de servidores, agora virtuais, a gerir, acrescentando o custo humano à equação. A adopção do paradigma *cloud* (a ver na secção 2) é o passo a dar para manter o custo total sob controlo.

O HP CloudSystem Matrix (CSM) [1] faz parte de uma pilha de software HP para computação na *cloud* que cobre todos os níveis de serviço considerados relevantes: IaaS (Infra-estrutura como Serviço), PaaS (Plataforma como Serviço) e SaaS (Software como Serviço). Apesar de ser a base desta pilha, i.e., oferecer o nível IaaS, é um produto extremamente complexo pois interage com todas as infra-estruturas: as computacionais (i.e., servidores físicos ou virtuais), as de armazenamento (do disco interno aos discos em servidores de armazenamento), e as de interligação (redes Ethernet e FC).

Apesar de toda a complexidade da infra-estrutura, real e virtual, que gere, o CSM torna conceptualmente simples a entrega aos consumidores de infra-estruturas para suporte a aplicações: 1) o *administrador* define que recursos da infra-estrutura estão disponíveis para integrar a "oferta *cloud*"; 2) o *arquitecto* define *templates* para as arquitecturas que considera adequadas para necessidades dos utilizadores/ /consumidores (e.g., arquitectura *3-tier* para uma solução ERP[1]); e 3) o consumidor escolhe o *template* que melhor se ajusta às suas necessidades e efectua um pedido de aprovisionamento de uma infra-estrutura.

Figura 3 - Infra-estrutura CSM



A interacção entre os diferentes interlocutores (1), (2), (3) e o CSM é fundamentalmente realizada sobre portais; contudo, especialmente no caso do utilizador-consumidor, o portal disponibilizado pelo produto tem sido considerado

[1] ERP (Enterprise Resource Planning) cobre uma larga fatia das necessidades de uma empresa. Permite uma visão integrada em tempo real dos seus principais processos de negócios.

como “complexo”, por apresentar informação demasiado técnica, “rígido”, por não poder ser customizado (por exemplo para suprimir a “informação demasiado técnica”), e “grosseiro” por não permitir a especificação mais fina das características da infra-estrutura que se quer aprovisionar (por exemplo, permite variar o número de CPUs e a quantidade de memória de um servidor, mas não permite escolher a tecnologia dos discos que se pretendem aprovisionar, e.g., SSD (Solid-State Drive) em vez de FC, 15K em vez de 10K RPM).

1.2 Objectivos

O objectivo desta tese é desenvolver um *framework* que permita, com base num conjunto (extensível/configurável) de opções pré-definidas e em *layouts* customizáveis, definir portais que se integram com o HP CloudSystem Matrix e que permitam aos utilizadores (consumidores IaaS) uma interacção mais simples, prática e organizada, não só na visualização de toda a informação mas também das acções possíveis de serem executadas nos recursos disponibilizados, que são ampliadas face à oferta disponível no portal standard do CSM.

Neste trabalho, foram estudados: modelos de serviço e de implantação (*deployment*) de *clouds*; a virtualização (não somente de servidores, mas também de armazenamento e de redes), pedra base de toda a tecnologia *cloud*; e os módulos e APIs disponíveis para interoperar com o CSM, nomeadamente a API-MOE e a API-VMware para desenvolver um *framework* com uma arquitectura multicamada (N-tier) sobre TCP/IP e usando o protocolo REST (Representational State Transfer) para troca de informação.

1.3 Organização

O presente trabalho encontra-se organizado em 6 secções, que vão desde o estudo prévio das diferentes tecnologias até às Considerações finais tidas como conclusão.

Assim, a secção 1 é a presente introdução, onde se apresenta a virtualização como tecnologia nuclear ao trabalho, enunciam-se os objectivos, a organização de todo o relatório e as suas contribuições.

Na secção 2 é feita uma abordagem à virtualização, desde a sua evolução histórica, culminando com a visão actual das quatro tecnologias mais representativas no mercado: VMware, Citrix, HP e Microsoft. Aborda-se não só a virtualização de servidores, mas também a de armazenamento e de rede.

Na secção 3, apresenta-se uma nova tecnologia, computação na *Cloud*, resultado da união da virtualização e módulos de orquestração. São abordados os seus modelos de implementação e os diferentes níveis de serviço que podem ser oferecidos neste paradigma.

Na secção 4 descreve-se uma nova tecnologia, HP Cloud Stack, como pilha de software e hardware. São abordadas tecnologias como HP CloudSystem Matrix, os

seus modelos de interacção, papeis e portais, conceitos fundamentais para perceber o restante relatório e as limitações desta última tecnologia, razão de ser deste trabalho.

Na secção 5 é apresentada a *framework* (protótipo), desde a arquitectura, implementação dos diferentes módulos, execução nos cenários possíveis e código principal, e finalmente feedback dos orientadores e redefinição de objectivos ou estratégias de realização e, por último, a discussão da relação entre os serviços e funcionalidades do HP MOE e a *framework* implementada.

Na secção 6 são apresentadas as conclusões sobre o trabalho realizado, perspectivas para desenvolvimentos futuros, fazendo-se uma discussão final sobre as dificuldades encontradas na implementação.

1.4 Contribuições

Todo o estudo de conceitos, relações e teorias tem sempre como objectivo último a aquisição de mais conhecimento e acima de tudo, aplicabilidade. Nesse intuito a principal contribuição deste trabalho está directamente relacionada com o principal objectivo, um *framework* modelo, de acesso rápido, intuitivo, organizado, personalizável e acima de tudo prático no acesso a recursos de uma infra-estrutura. Apresentam-se de seguida todas as contribuições identificadas.

Um modelo simples com uma curta curva de aprendizagem e uma flexibilidade com acessos próprios, individuais e personalizados a cada utilizador. O acesso a toda a informação/recursos é feita numa visualização das componentes principais em árvore, permitindo assim uma rápida visualização das configurações de todo o Serviço criado e um acesso rápido e intuitivo aos detalhes de cada componente.

Visualização imediata do Catálogo de Serviços com todos os Templates e a correspondente associação dos serviços por Template. A descrição é a mesma relativamente aos recursos associados a cada Serviço. A informação visualizada é apenas a necessária em cada visualização e instanciação.

Modelo personalizável com acções como ocultar ou visualizar painéis de informação, lista de todos os pedidos, listagem por Template, por Serviço, layout de toda a interface, do menu em árvore, um simples clique e toda o portal fica personalizável a um determinado utilizador.

Disponibilização de todas as acções possíveis a cada recurso na tab respectiva da visualização da informação associada a esse recurso. Acções acessíveis e rápidas de executar, sem menus e submenus com sucessivos cliques que só atrasam todo o processo. A acessibilidade aos recursos é completa com acções de grupo e acção individuais a cada recurso, permitindo assim o verdadeiro aproveitamento de todo o Serviço criado.

Uma informação hierarquizada e organizada em agrupamentos lógicos por recursos é um processo muito fácil de seguir e perceber, associada a esta

disponibilização de informação, acções que possam ser executadas nesse/s recurso/s é o objectivo da simplicidade e performance deste modelo.

1.5 Resumo

Neste capítulo fez-se uma descrição e contexto introdutório de conceitos como Virtualização, *Cloud Computing* e *HP CloudSystem Matrix*, onde se descreveram as mais-valias do uso destas tecnologias, vantagens e inconvenientes e os papéis/funções dos principais interlocutores. Descreveram-se os objectivos desta dissertação, um *framework* que solucione e melhore os problemas existentes em modelos implementados actualmente. Foi igualmente descrita toda a organização das diferentes secções e por ultimo, as contribuições tidas no presente trabalho.

2. Virtualização

2.1 Evolução histórica

Os primeiros passos da virtualização foram dados aproximadamente há meio século, e tiveram origem em preocupações como heterogeneidade, portabilidade, estabilidade e utilização mais eficiente de recursos. Tudo começou na IBM com a sua longa lista de variantes de uma dada arquitectura-base, com as novas gerações incompatíveis com as anteriores, com múltiplos SOs (mono-utilizador interactivo; *batch*; *timesharing*; etc.) criando insatisfação nos clientes. Como resposta, IBM lançou nos anos 60 a arquitectura 360 com suporte para virtualização (ao nível do *hardware*), e com o sistema de operação CP/CMS. O CP (*Control Program*) era usado para gerir as VMs e que executavam o CMS (*Console Monitor System*), um SO interactivo com uma interface "linha de comando".

Por outro lado um outro tipo de virtualização, dito virtualização de aplicações, procura que uma aplicação possa ser executada numa variedade de plataformas distintas que apenas têm em comum o sistema de *run-time* que implementa a VM (não requer que uma VM seja "totalmente" equivalente a uma máquina física, e portanto capaz de executar um SO). Tal é o caso da JVM (*Java Virtual Machine*) capaz de executar aplicações desenvolvidas na linguagem Java (nascida em 1990 de um projecto denominado "*Stealth*" da Sun Microsystems).

A implementação desta tecnologia, tão bem aceite e cada vez mais usada nas organizações, demonstra a mais-valia das propriedades que caracterizam o seu funcionamento:

- ✓ *Isolamento*: Os processos das diferentes máquinas virtuais não podem interferir entre si, ou seja, um processo de uma VM não pode interferir noutra VM, nem sequer no próprio sistema hospedeiro (*host*) que suporta toda a virtualização;
- ✓ *Inspecção*: O gestor de máquinas virtuais (Virtual Machine Manager - VMM) ou hipervisor, deve ter acesso e controlo sobre todas as VMs, e até sobre todos os seus processos;
- ✓ *Interposição*: O VMM deve poder interagir com as VMs, controlar as suas operações, ter acesso ao fluxo de instruções máquina que estas executam e *remover*, adicionar ou alterar essas instruções se for necessário;
- ✓ Eficiência: As instruções não privilegiadas devem poder ser executadas directamente no hardware, sem interposição do VMM;
- ✓ *Gestão*: O VMM deve não só permitir manipular uma VM isoladamente, mas também executar uma operação, de uma só vez, sobre múltiplas VMs;
- ✓ *Compatibilidade de software*: A virtualização simula uma determinada plataforma (real) e, desse modo, todo o software disponível para essa plataforma deve poder ser executado "tal como está (original)" numa VM;

- ✓ *Desempenho*: A sobrecarga (*overhead*) da camada extra de software, pode sacrificar o desempenho do sistema hospedado (*guest*) mas a relação custo/benefícios do uso da virtualização deve ser francamente positiva.

O Anexo A descreve mais detalhadamente as vantagens e inconvenientes da virtualização.

2.2 Virtualização de Servidores

O dinamismo do triângulo existente entre fornecedores de *software*, *hardware* e seus clientes é uma constante. Fornecedores como a VMware e a Microsoft publicitam as suas soluções de virtualização destacando as mais-valias como a utilização eficiente de recursos partilhados, gestão centralizada, e migração dinâmica; os fornecedores de *hardware* publicitam menores consumos de energia, e maior interoperabilidade e suporte; as organizações olham para este mundo com o objectivo último da eficiência, produtividade e custos. A virtualização aparece como uma forma eficaz de fazer mais com menos investimento.

A primeira razão pela qual se torna muito interessante executar múltiplas VMs num único servidor é a mesma pela qual executamos múltiplas aplicações num único servidor: os recursos estariam, se não o fizéssemos, muito subaproveitados. Sendo assim, porque não executar simplesmente múltiplas aplicações no servidor usando um só SO? Podemos avançar desde já com uma razão simples: os fornecedores certificam os seus produtos, aplicações, e serviços apenas para determinados ambientes e pilhas de software; vejamos alguns exemplos: 1) a Microsoft não certifica configurações MSCS (Microsoft Cluster Server) que incluam o serviço Active Domain num dos nós dos cluster; 2) a VMware disponibiliza o seu produto de gestão vCenter Server sobre Unix (SUSE Linux) na forma de uma appliance fornecida pela própria VMware. Assim, se dispusermos de um servidor que tem os recursos apropriados para suportar um nó MSCS e uma instância vCenter teremos de os instalar sob a forma duas VMs a correrem sob um hipervisor instalado nesse servidor. Uma outra situação que "obriga" (do ponto de vista financeiro) à utilização de VMs é a de aplicações *legacy*: uma aplicação antiga (e é demasiado oneroso actualizá-la) está certificada apenas para uma dada versão de SO, também antigo, que não possui *drivers* adequados para os servidores mais actuais, sendo que o servidor antigo já não tem capacidade para executar a aplicação (e é muito oneroso mantê-lo). Uma solução simples é instalar um hipervisor num servidor recente, criar uma VM, e nela instalar a versão antiga do SO e a aplicação.

A arquitectura da virtualização está descrita no Anexo B e os Tipos de Implementação no Anexo C, uma leitura ajuda a perceber todos estes conceitos e tecnologias que se seguem.

Actualmente o número de tecnologias disponíveis para implementação e gestão de máquinas virtuais tem vindo a crescer, destacando-se algumas pela

disponibilização de um conjunto de ferramentas que as tornam vencedoras nos mercados. Seguem-se quatro exemplos das empresas (e suas tecnologias) que mais se destacam hoje em dia:

- ✓ *VmWare, Inc.:* VmWare Player, Workstation, Fusion, ESXi
- ✓ *Citrix Systems, Inc:* XenServer
- ✓ *Hewlett-Packard:* HP Integrity Virtual Machines
- ✓ *Microsoft:* Virtual PC, Hyper-V

2.2.1 VMware (VMware, Inc.)

O VMware apresenta-se como uma plataforma líder (80%) no mercado da virtualização, com uma tecnologia de virtualização do tipo 1 (virtualização completa, também designada nativa ou *bare metal*). Oferece um vasto leque de produtos, desde os hipervisores para *desktops* e *laptops* (VMware Player/Workstation/Fusion) aos para servidores de grande porte (VMware vSphere ESXi) e ainda um vasto conjunto de *software*: de gestão de VMs, orquestração de operações, *Cloud* IaaS, etc..

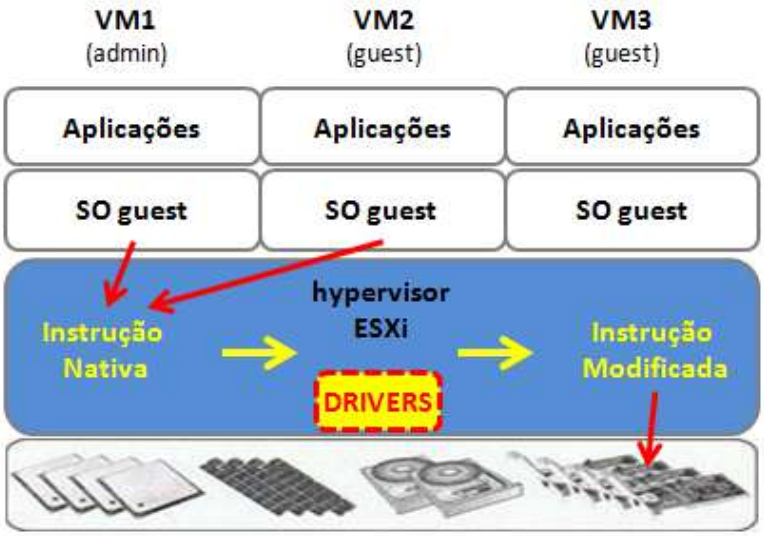
De todos os produtos da VMware, interessa-nos destacar aqui apenas a tecnologia de virtualização ESXi, oferecida para servidores, que descrevemos em seguida muito sucintamente:

- ✓ *VMware vSphere ESXi:* É uma VMM ou hipervisor tipo 1 garantindo um elevado nível de isolamento entre os recursos oferecidos às VMs, sejam estes o processador, memória, discos ou adaptadores de rede. É instalado directamente sobre o *hardware* do servidor, eliminando assim a sobrecarga de ter um SO standard sobre o qual corre um hipervisor; os hipervisores de tipo 1 exibem, portanto, melhor desempenho e aumentam a segurança. O facto de permitir que tudo seja virtualizado torna a VM ainda mais completa. O pacote inclui apenas o hipervisor ESXi e ferramentas básicas de gestão.

A forma como as instruções do sistema hospedado (*guest*) são executadas num ambiente ESXi pode ser descrita com ajuda da figura 4 [8]: 1) as instruções oriundas de aplicações que se executam sobre o SO hospedado são executadas directamente no processador real; 2) as instruções oriundas do próprio SO hospedado são verificadas antes de serem executadas, sendo que as privilegiadas são, traduzidas ou modificadas para comandos ou instruções do próprio hipervisor, uma técnica que é denominada como “Tradução Binária”.

Esta técnica, complementada com outras técnicas de “aceleração” de operações sobre I/O, memória e de gestão de recursos entre VMs fazem com que a VMware consiga atingir desempenhos muito próximos de um ambiente nativo, i.e., não virtualizado.

Figura 4 - Arquitectura VMware ESXi (Tipo-1)



Na versão vSphere para servidores de grande porte estão disponíveis muitas funcionalidades [9], como a migração automática de VMs (balanceamento de carga), a alta disponibilidade para vCenter (suportada pelo vCenter Server Heartbeat), a recuperação de destastres (Site Recovery Manager).

Para maiores detalhes e restrições da VMware, consultar o Anexo D [10,11].

2.2.2 Xen (Citrix Systems, Inc.)

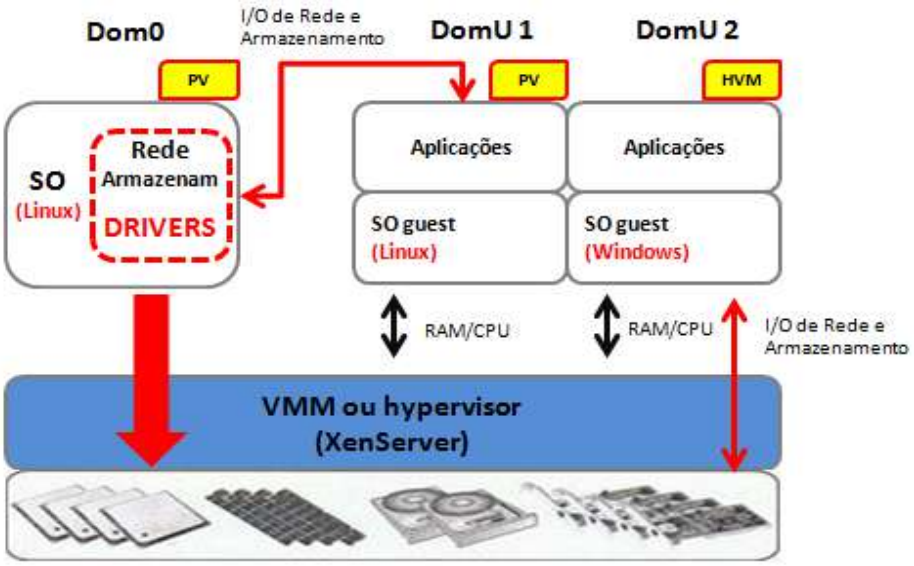
O XenServer é a plataforma de virtualização da Citrix, baseada no hipervisor Xen; a versão base, designada Standard, é gratuita e oferece, para além do hipervisor, ferramentas como o XenCenter, uma consola de gestão para múltiplos servidores, e o XenMotion para migração "in vivo" de VMs.

O hipervisor Xen, na sua mais recente versão, é uma tecnologia de virtualização do tipo 1, instalada num núcleo Linux (x86, x86-64, ou IA-64 SUSE Linux), que permite a criação de várias VMs com recursos e aplicações partilhadas. Contrariamente ao VMware, não usa a tradução binária, mas uma combinação de paravirtualização [12] (eliminando a latência no desempenho introduzida pela tradução binária) e virtualização assistida por *hardware* [13]. Na paravirtualização a máquina abstracta é quase, mas não completamente igual ao *hardware* hospedeiro; por isso, o SO hospedado (*guest*) tem de ser modificado. Como tal só é possível para SOs disponíveis em código aberto, o Xen executa os outros (e.g., Windows) recorrendo a uma combinação da paravirtualização com a virtualização assistida por *hardware* oferecida nos processadores Intel (Intel-VT) e AMD (AMD-V).

Na arquitectura XenServer, figura 5, existe uma VM privilegiada para controlo do hipervisor, denominada "Dom0"; esta VM é parte integrante do ambiente

XenServer, executa uma versão paravirtualizada do Linux, e contém os *drivers* dos dispositivos disponíveis no sistema. As outras VMs são os "DomUs" (domínios do utilizador). Nesta arquitectura, os DomUs comunicam directamente com o hipervisor que controla a memória e o processador do hospedeiro, e, se usam paravirtualização, comunicam com o Dom0 para operações de I/O.

Figura 5 - Arquitectura XenServer



O Anexo E descreve os detalhes da plataforma e restrições da Citrix [10,14].

2.2.3 Integrity VM (HP)

HP Integrity Virtual Machines (Integrity VM ou HPVM) é uma tecnologia de virtualização do tipo 1, instalada num servidor da linha Integrity IA-64 com o sistema de operação HP-UX 11i v3.

A gestão do hipervisor é efectuada com a ferramenta HP Integrity Virtual Machines Manager (HPVMMGR); mas, também é possível gerir os próprios sistemas hospedados desde que nestes sejam instalados os agentes HPVirtualMachine e HPVSwitch. Esta ferramenta apresenta benefícios como flexibilidade e maximização dos recursos do servidor; consolidação de servidores organizacionais; rapidez na implementação e disponibilização de novos ambientes; permite a migração online e offline; isolamento de VMs; oferece alta disponibilidade; monitoriza automaticamente todas as aplicações; e permite a visualização e configuração simplificada com ferramentas como HP Integrity Virtual Machines Manager (HPVMMGR), System Management Home (HPSMH) e HP Virtualization Manager (HPVMAN).

O Anexo F descreve os detalhes e restrições Integrity Virtual Machines da HP [10].

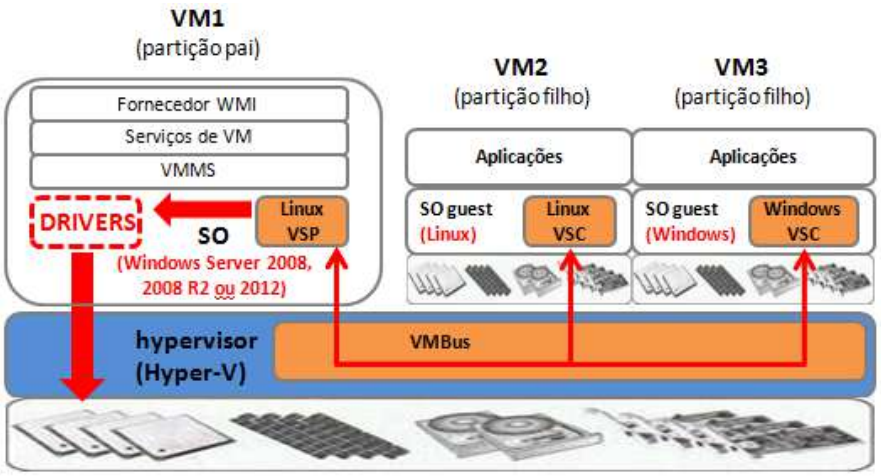
2.2.4 Hyper-V (Microsoft)

O Hyper-V é uma tecnologia de virtualização do tipo 1, da Microsoft, para servidores x86-64 com suporte à virtualização assistida por *hardware*. O produto é disponibilizado de forma isolada (*standalone*), ou integrado em versões Windows Server (2008 R2 ou 2012).

A gestão do ambiente virtual é efectuada com a ferramenta System Center Virtual Machine Manager (SCVMM), que fornece 3 formas distintas de interactuar com o hipervisor: consola de administração, portal self-service (para utilizadores finais) e Windows PowerShell (usando commandlets).

Na arquitectura Hyper-V [15], figura 6, existem dois tipos de partição: 1) a "partição pai", uma VM de administração de todo o sistema, com um Windows Server 2008, 2008 R2 ou 2012, *drivers* de acesso e controlo do *hardware* e ferramentas de gestão; 2) as partições filhas, nas quais são criadas as restantes VMs, completamente isoladas e sem acesso directo ao hardware. No acesso ao *hardware*, há 3 componentes importantes a mencionar: a) o VSP (Virtual Service Provider) responsável por receber as chamadas das partições filhas e aceder aos *drivers* dos periféricos; b) VSC (Virtual Service Client), um módulo instalado no SO hospedado que solicita, via VMBus, o acesso aos periféricos da partição pai; c) o VMBus, canal de comunicação entre VSC e VSP onde se desenrolam as comunicações entre as partições.

Figura 6 - Arquitectura Hyper-V



A Microsoft apresenta-nos, na recente versão Hyper-V 2012, uma panóplia de funcionalidades que a colocam na primeira linha das soluções de virtualização, a par da VMware: migração online e replicação de VMs, migração de armazenamento, e *Clustering*.

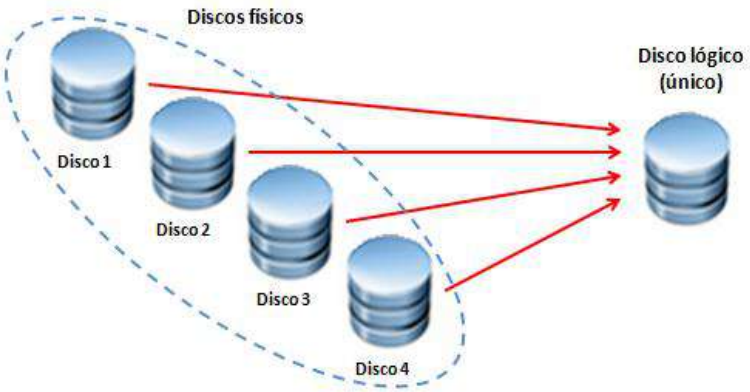
O Anexo D descreve os detalhes as restrições do Hyper-V da Microsoft [10,16,17].

2.3 Virtualização do Armazenamento

Há muito que o armazenamento passou a ser um tópico de "primeira classe" tal como "a rede". Para tal contribuiu significativamente a introdução de tecnologias tais como o RAID (Redundant Array of Independent Disks) [18] que promoveram o desempenho e/ou a tolerância a faltas a (comparativamente) baixo custo – note-se que no artigo original de Patterson o I significava *Inexpensive*. Com a introdução do RAID, rapidamente apareceram os armários de discos (*disk arrays*), externos ao "sistema" e, uma vez que o protocolo SCSI suportava a interligação de discos (*Logical Units*) a múltiplos adaptadores e, daí, a múltiplos sistemas, o conceito de rede de armazenamento aparece naturalmente.

Como se mostra na figura 7, o disco lógico aparece como uma forma elementar de virtualização, já que "aglutina" elementos de diferentes discos físicos mas é apresentado ao sistema como um único disco – ao ponto de ser impossível ao próprio sistema de operação distingui-lo de um disco físico.

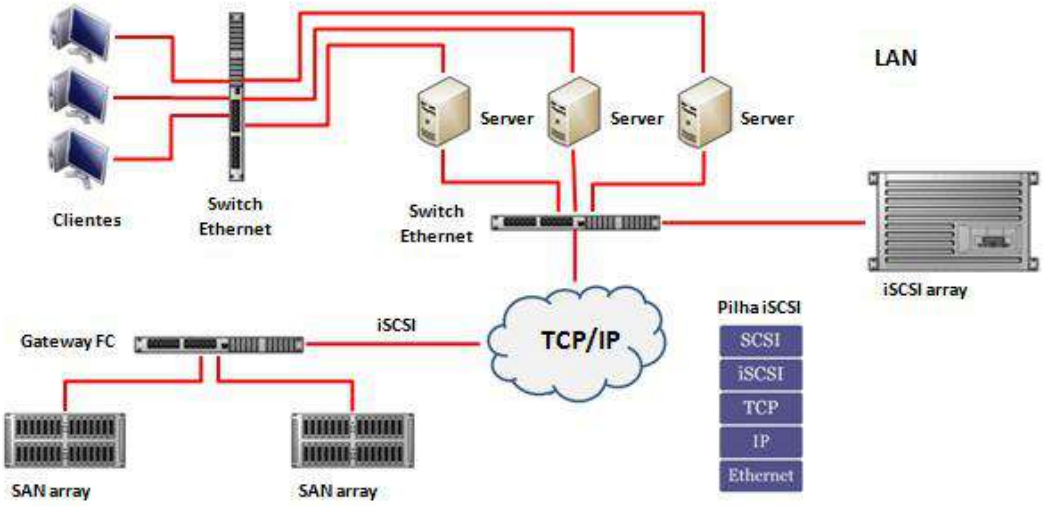
Figura 7 - Redundant Array Independent Disks (RAID)



Numa rede de armazenamento que interliga vários armários de discos e servidores (*hosts*), os primeiros oferecem volumes (ou discos lógicos) acessíveis por identificadores únicos e os segundos tomam posse desses volumes (tipicamente cada *host* tem uso exclusivo de um ou mais volumes) e formatam-nos, neles instalando sistemas de ficheiros (e.g., ext3, XFS, NTFS), que depois usam da forma mais conveniente. Este conceito de rede de armazenamento é conhecido como SAN – Storage Area Network, figura 8.

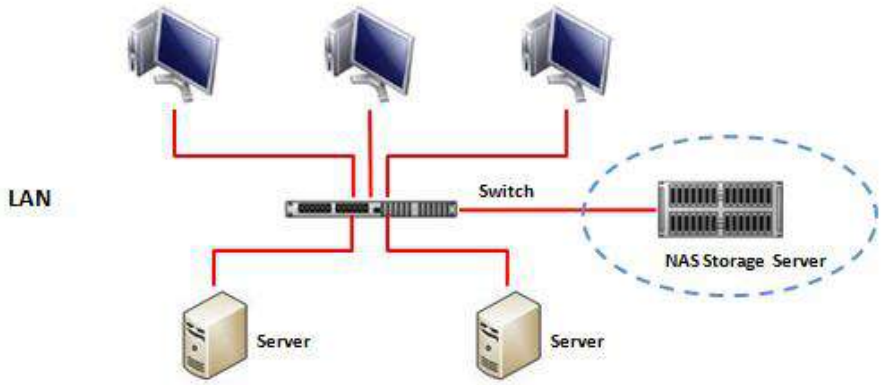
A forma mais comum de SAN usa uma infra-estrutura Fibre Channel (FC), o que significa que tanto *hosts* como *disk arrays* possuem interfaces FC, e na rede existem comutadores (*switches*) FC aos quais *hosts* e *arrays* (e eventualmente outros *switches*) se interligam. Uma outra tecnologia que pode ser usada em SANs é a Ethernet: neste caso *hosts*, *disk arrays* e *switches* têm interfaces Ethernet, e o protocolo de transporte usado é FCoE (Fibre Channel over Ethernet). A figura seguinte mostra-se um ambiente que inclui simultaneamente uma outra tecnologia de "transporte", ISCSI e uma "zona" FC.

Figura 8 - Storage Area Network (SAN)



Se bem que a relação volume/*host* de um-para-um seja de longe a mais frequente, também é possível estabelecer uma relação de um-para-muitos, na qual um volume é detido, de forma partilhada, por múltiplos *hosts*; tal situação implica necessariamente o uso de sistemas de ficheiros (SF) especializados, conhecidos como *shared-disk file systems*. Como exemplo indica-se o GFS [19], GPFS [20], e VMFS [21], este último desenvolvido especificamente para armazenar VMs. Num SF para discos partilhados todos os *hosts* que partilham um dado volume têm uma visão coerente do estado do volume. Em situações de tolerância a faltas esta arquitectura de discos partilhados é muito apetecível pois em caso de falha de um *host* um dos restantes inicia uma recuperação (ao estilo de um sistema transaccional) das últimas operações e rapidamente o SF regressa a um estado coerente.

Figura 9 - Network Attached Storage (NAS)



Uma outra forma, completamente distinta, de armazenar informação é usar um SF distribuído; neste caso, não há partilha de discos físicos, mas sim de ficheiros, sendo que um *host* pode deter apenas parte da árvore do SF (e.g., AFS) ou, nos SFs distribuídos com arquitectura cliente/servidor, um *host*, dito cliente dispõe de um

módulo que lhe permite aceder ao SF remoto de um outro *host*, dito servidor (e.g., NFS e CIFS), como se o SF remoto fosse local.

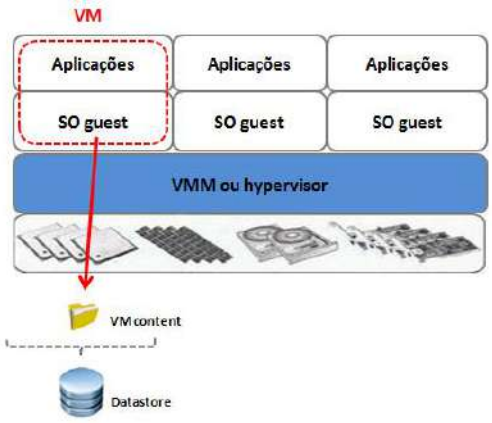
2.3.1 Armazenamento em ambientes virtualizados

Quando se aborda o armazenamento em ambientes virtualizados, são de considerar os seguintes aspectos: a) como representar a arquitectura de uma VM propriamente dita (quanta memória e CPUs tem, qual o *chipset* utilizado, que adaptadores – de LAN, SAN, gráficos – possui, etc.); b) como representar um disco acessível à VM; e, c) como armazenar conteúdos (voláteis ou não) de componentes da VM, tais como a memória RAM e a memória não volátil, NVRAM, que contém o BIOS.

A solução adoptada pelo hipervisor VMware ESXi[2] é a seguinte: a) a representação da arquitectura é efectuada em XML que possui as informações de configuração; b) um disco acessível à VM é representado ou por um ficheiro que virtualiza o próprio disco, ou por um ficheiro que, ao estilo de uma ligação simbólica, descreve o caminho para um disco real; e finalmente, c) tais conteúdos são armazenados em ficheiros. Isto é, tudo é representado por, ou via, ficheiros.

A pergunta que se segue é óbvia: onde armazenar esses ficheiros? E a resposta, evidente: num repositório (*datastore*, na terminologia da VMware) gerido pelo hipervisor, onde uma VM aparece como um pasta, e os seus componentes como ficheiros no interior dessa pasta, como se mostra na figura 10.

Figura 10 - Datastore e VMs



O VMware ESXi suporta armazenamento em sistema de ficheiros VMFS (sobre discos) ou NFS; no caso do VMFS, os discos podem ser internos ao servidor, ou podem estar numa SAN.

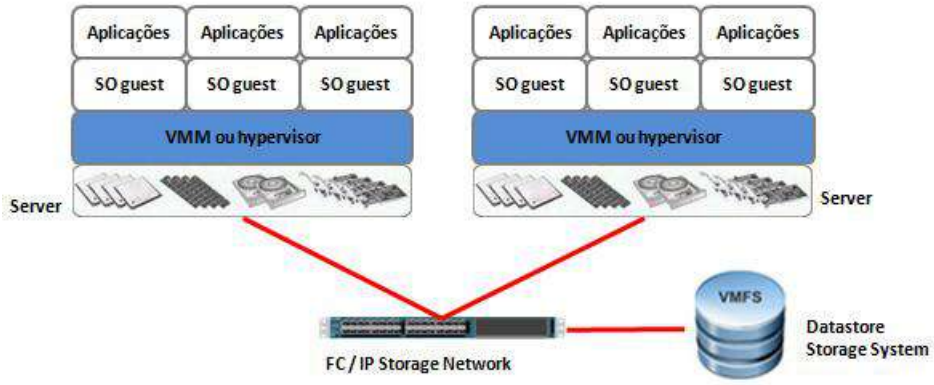
Sistema de Ficheiros VMFS

O VMFS (Virtual Machine File System) é um sistema de ficheiros em *cluster*, e foi optimizado para armazenar grandes ficheiros e realizar operações de entrada e

[2] Para uma comparação entre o VMware ESXi e o Citrix Xen ver Anexo H

saída (I/O) com que movimentam grandes volumes de dados. Os dois servidores, figura 11, têm, portanto, uma visão coerente do *datastore*, e das VMs nele existentes; por isso é, quando comparamos esta arquitectura com uma na qual cada servidor tem a sua datastore, muito mais simples e eficiente mover uma VM activa de um servidor para o outro, já que apenas as páginas modificadas residentes em memória têm de ser copiadas (o resto está no SF partilhado).

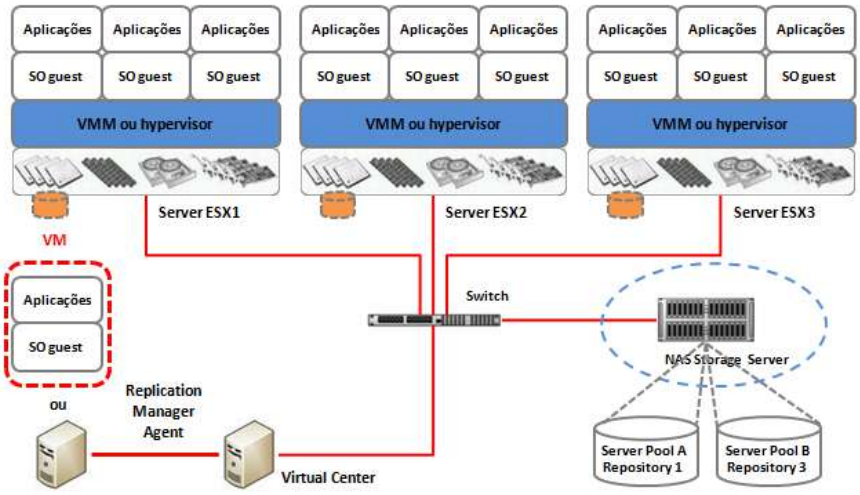
Figura 11 - Sistema de Ficheiros VMFS



Network File System

Nos casos em que os *datastores* residem num sistema de ficheiros NFS [22] os servidores ESXi podem também partilhar o mesmo *datastore*, como se vê na figura 12; assim, a movimentação de VMs entre servidores, por razões de equilíbrio de cargas ou de manutenção de um ou outro servidor, também são muito eficientes.

Figura 12 - Network File System (NFS)



Para concluir, note-se que se abrem, nos ambientes virtualizados, duas oportunidades muito importantes quando um disco virtual é realizado sob a forma de ficheiro: a) a potencial poupança de espaço e b) a possibilidade de implementar com facilidade uma técnica de *snapshots*. No primeiro caso, conhecido como *thin provisioning*, referimo-nos à possibilidade de usar um ficheiro esparso para atribuir um espaço lógico de endereçamento ao disco (ficheiro) muito superior ao espaço de

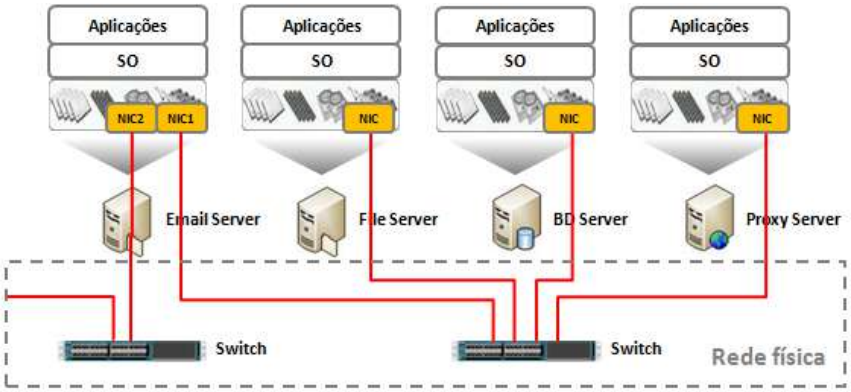
facto "consumido" pelo ficheiro no SF. No segundo, usando um mecanismo de versões baseado em técnicas de *copy-on-write*, armazenar *snapshots* da VM em determinados instantes e para usar como mecanismo de recuperação do estado, em caso de "falha".

2.4 Virtualização da Rede

A virtualização é algo mais que consolidação de servidores - CPU, memória e armazenamento; depois de analisada toda a infra-estrutura de suporte a uma implementação desta natureza, torna-se evidente a ideia simplista desse triângulo. A infra-estrutura de rede virtual funciona como ligação de componentes nessa plataforma: ligações virtuais através dos controladores de rede virtual (Virtual Network Interface Card - vNIC[3]), ligações físicas com os adaptadores físicos (Physical Network Interface Card - pNIC), comutadores virtuais (vSwitch) para essas ligações, etc., uma vasta lista de *software* e *hardware* responsáveis pela comunicação de todos os componentes dos pares VM/VM, VM/servidor e servidor/servidor [23].

Numa infra-estrutura de rede tradicional, figura 13, cada servidor tem uma ou mais placas de rede (NIC) para comunicação com os restantes.

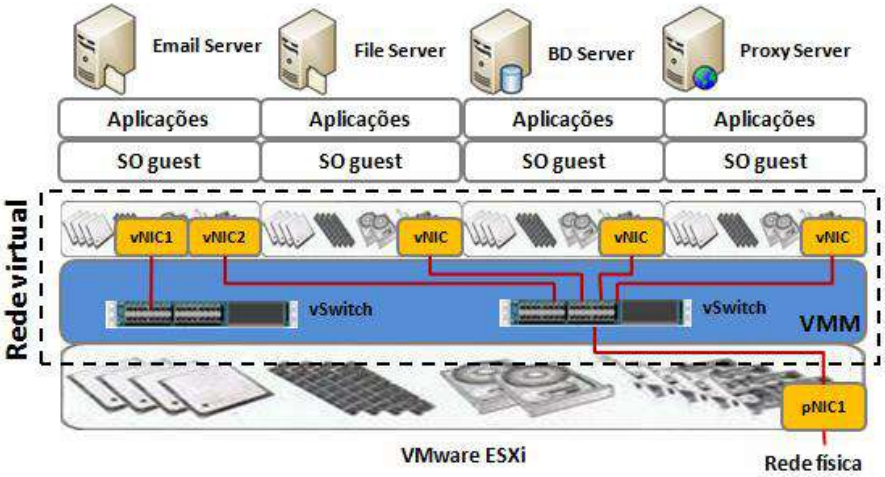
Figura 13 - Infra-estrutura de rede tradicional



Como sabemos, um hipervisor é responsável pela abstracção do *hardware* oferecido às VMs e, na visão de uma VM, este *hardware* (virtual) de que dispõe é seu, e é completo, desconhecendo que esse *hardware* possa estar a ser partilhado com outras VMs ou que, eventualmente, nem exista. A interface virtual de rede (vNIC), figura 14, não possui qualquer componente físico, é apenas *software*; contudo, ela exibe as mesmas propriedades de uma placa física, incluindo um endereço MAC (IEEE 802-2001 Standard). Uma vNIC é ligada a uma outra abstracção, o vSwitch sendo que este pode comutar tráfego que flui apenas no interior do servidor que alberga as VMs, ou estar associado a uma (ou mais, para redundância e/ou aumento da largura de banda) placa de rede física (pNIC).

[3] Mais uma vez seguimos aqui a terminologia da VMware, por simplicidade.

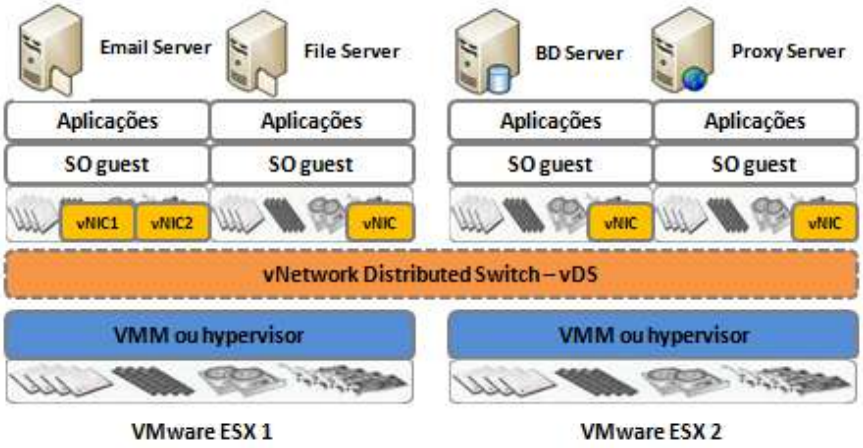
Figura 14 - Infra-estrutura de rede virtual



Nesta infra-estrutura, não existe o limite associado à velocidade da placa de rede, mas sim associado à largura de banda (LB) da memória; facilmente se depreende, também, que toda a rede entre VMs hospedadas num servidor não contribui para o congestionamento da rede física (entre servidores). Contudo, o administrador pode impor quotas na utilização de recursos, por exemplo, na LB da vNIC.

Os vSwitch têm todas as funcionalidades de um *switch* real: suportam agregação de tráfego (IEEE 802ad), VLANs (IEEE 802.1Q), etc. Nesse sentido, a virtualização começa a ir contra-corrente no que deveria ser o simplificar a administração de uma infra-estrutura, pois o administrador de redes tem agora de administrar os *switches* reais e ainda os virtuais.

Figura 15 - vNetwork Distributed Switch – vDS

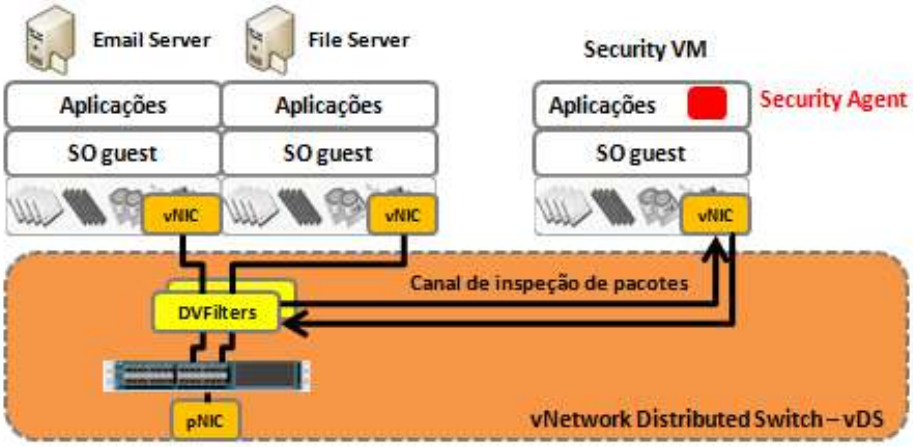


Para resolver este problema, a VMware propõe (figura 15) um *switch* virtual distribuído (vNetwork Distributed Switch – vDS) [23] que representa a agregação de vários *switchs* virtuais de diferentes servidores, ocultando as camadas inferiores, e oferecendo um suporte de rede uniforme, que simplifica, entre outras, a administração da rede, as migrações de VMs, e o desempenho – por exemplo, ao associar um vDS a

múltiplas pNICs, o hipervisor pode, de forma automática, alterar a lógica de encaminhamento de pacotes para redistribuir o tráfego entre as pNICs, aumentado o desempenho e minimizando as falhas.

Na infra-estrutura de rede virtualizada, os aspectos relativos à segurança dessa rede são também contemplados, sendo apresentadas soluções de inspecção de fluxos de pacotes como é o caso dos Distributed Virtual Filter (DVFilter) [23], apresentado na figura 16.

Figura 16 - Distributed Virtual Filter (DVFilter)



2.5 Resumo

Neste capítulo fez-se uma breve descrição histórica para enquadramento do restante capítulo. A virtualização de servidores aparece como núcleo central nas suas quatro plataformas de virtualização que considerei mais importantes: ESXi (VMware, Inc); XenServer (Citrix Systems, Inc); HP Integrity Virtual Machines (Hewlett-Packard) e o Hyper-V (Microsoft). Foram também abordadas tecnologias na área da virtualização de armazenamento, em conceitos como RAID, SAN, NAS e nos ambientes virtualizados, VMFS e NFS. Também a virtualização de rede não foi descurada e nesse intuito, foram abordados conceitos como vNIC, pNIC, VLANs, vNetwork Distributed Switch e Distributed Virtual Filter.

3. Computação na *Cloud*

Como anteriormente referimos, o uso crescente de Tecnologias de Informação (TI) nas organizações leva geralmente a uma proliferação de servidores físicos o que as obriga, mais cedo ou mais tarde, a adoptar estratégias de consolidação de servidores, com a consequente redução de custos de *hardware* e energia. Resolvidas estas questões restam, contudo, outras - o uso de virtualização não reduz, por si só, o esforço de administração das infra-estruturas, nem resolve problemas de disponibilidade, desempenho, segurança, etc., que são questões que nem todas as organizações querem ou estão aptas a resolver, preferindo por vezes que "alguém" assuma esta responsabilidade.

A Computação na *Cloud* (CC), apresenta-se assim como uma sequência de passos nesse sentido, cada um correspondendo a um modelo de serviço (ver em 3.2) e resolvendo mais uma "fatia" dos problemas enunciados. A CC é actualmente associada a palavras chave como padronização, automatização, velocidade, disponibilidade, elasticidade, acesso global, e à ideia de usar um conjunto de recursos partilhados. As organizações podem então preocupar-se essencialmente com as melhorias operacionais e para isso, procuram soluções que possibilitem uma afectação (e, posteriormente, libertação) de recursos de forma rápida e automática.

> "Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."
>
> The NIST Definition of Cloud Computing: [Online] Available: http://www.nist.gov/itl/csd/cloud-102511.cfm

3.1 Perímetro de uma *cloud*

Designamos por perímetro (ou modelo de implantação) de uma *cloud* a forma como caracterizamos o conjunto dos utilizadores com acesso aos serviços dessa *cloud*. Nesta secção apresentamos apenas os três tipos mais comuns, deixando de fora outros, como as *clouds* partilhadas e federadas.

Clouds Privadas

Ambientes construídos exclusivamente para os utilizadores de uma única organização. Nesta tipologia os recursos (infra-estrutura e aplicações) são totalmente detidos pela organização, podendo ser administrados pela própria organização, por terceiros, ou ainda pela combinação das duas entidades. A infra-estrutura está normalmente instalada num *data center* privado, e as maiores preocupações são geralmente a garantia de disponibilidade e o desempenho dos serviços e aplicações.

Clouds Públicas

Ambientes construídos para oferecer serviços quer a utilizadores individuais, quer a organizações. Nesta tipologia a infra-estrutura é geralmente detida pela organização que comercializa os referidos serviços, que são tipicamente cobrados na forma "*pay-per-use*". As maiores preocupações da entidade gestora são garantir a privacidade e a segurança (note-se que os "utilizadores" podem, de facto, ser administradores dos servidores que detêm, e podem instalar software), o isolamento entre recursos detidos por utilizadores distintos, e ainda assim garantir bom desempenho dos serviços e aplicações.

Clouds Híbridas

As *clouds* híbridas representam a junção de dois modelos de *cloud*, privada e pública. Este modelo é normalmente implantado em organizações que têm picos de trabalho que, por uma ou outra razão, não podem suportar na sua *cloud* privada e, nessas alturas, a "elasticidade" é usada para aprovisionar recursos de uma *cloud* pública.

3.2 Modelos de serviço

Existem três modelos base de serviços de Cloud Computing que podem ser oferecidos às organizações, e que se diferenciam pelas configurações e responsabilidades do provedor de serviços, figura 17.

IaaS (Infrastructure as a Service)

No modelo "infra-estrutura como serviço", o provedor oferece ao consumidor uma infra-estrutura de processamento e armazenamento com a configuração que se adequa às suas necessidades; o cliente não tem, assim, necessidade de adquirir servidores, racks, software e até mesmo espaço físico. São exemplos de provedores públicos IaaS a *Amazon EC2* [2], e o *GoGrid* [3]; já o *Eucalyptus* [4], o *vCloud* (da VMware) e o HP CloudSystem Matrix são exemplo de produtos software para a criação de *clouds* privadas.

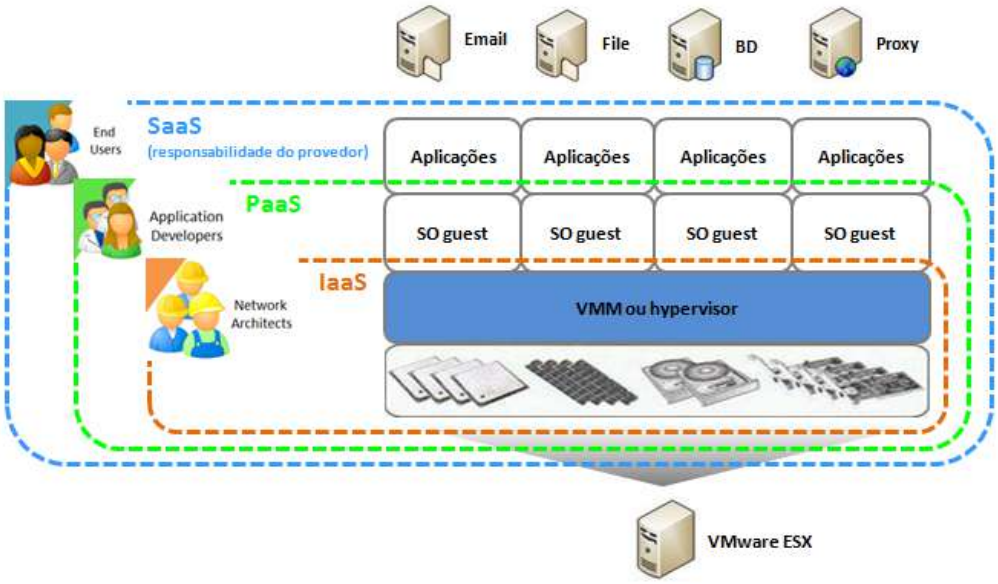
PaaS (Platform as a Service)

No modelo "plataforma como serviço" o provedor disponibiliza plataformas computacionais completas (incluindo, naturalmente, a infra-estrutura computacional) para os utilizadores/consumidores dessas plataformas integrarem as suas aplicações (existentes ou a desenvolver), sendo que estas serão posteriormente disponibilizadas na *cloud*. Como exemplo, indicam-se as plataformas *LAMP* (Linux, Apache, MySQL, PHP), *Google AppEngine* [5] e *Microsoft Azure* [6].

SaaS (Software as a Service)

Neste modelo, o provedor disponibiliza ao cliente final uma aplicação pronta-a-usar; como exemplos mencionam-se as *Google Apps* [7], *Google Docs*, e *Microsoft SharePoint Online*. Este é o paradigma *cloud* por excelência: utilidade, facilidade (interacção via *browser*), disponibilidade e, por vezes, até alguma capacidade de personalização.

Figura 17 - Relação entre serviços e responsabilidades

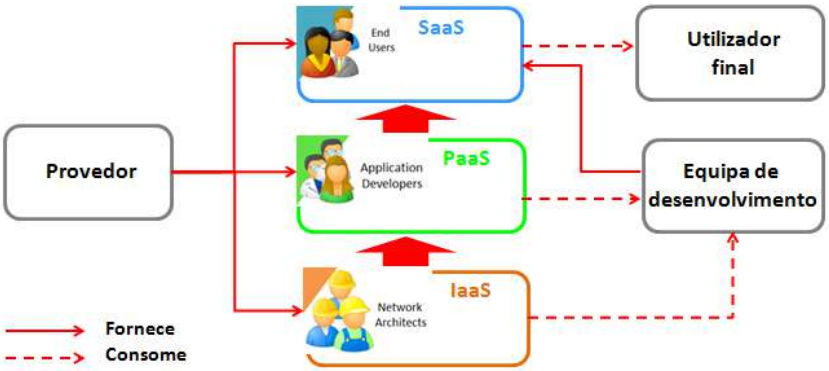


Associados à CC, vários modelos "*As a Service*" têm surgido para responder às necessidades das organizações; contudo, todos acabam por ser enquadrados nos três tipos base (Infra-estrutura, Plataforma, e Software) de acordo com os serviços que disponibilizam.

> "Database As A Service (DBaaS or DaaS), Network As A Service (NaaS), DataCenter As A Service (DCaaS), Storage As A Service (StoraaS), Everything As A Service (EaaS), Data As A Service (DaaS), Security As A Service (SecaaS), Computing As A Service (CaaS), Desktop As A Service (VDI), Voice As A Service (VaaS), Communication As A Service (ComaaS), Testing As A Service (TaaS), Backup As A Service (BaaS), Monitoring As A Service (MaaS), Recovery As A Service (RaaS), Content Delivery Network As a Service (CDNaaS), Application As A Service (AaaS), Business Intelligence As A Service (BIaaS), etc."
>
> Cloud9IDE: [Online] Available: https://c9.io/

As relações entre os modelos IaaS, PaaS e SaaS são hierárquicas, como se pode observar na figura 18, que ainda ilustra os papéis do provedor de serviços, equipas de desenvolvimento e utilizadores finais. Como se observa, o IaaS fornece recursos de *hardware* e *software* ao PaaS e SaaS, o PaaS oferece ferramentas para desenvolvimento ao SaaS e por fim o SaaS executa e disponibiliza aplicações aos utilizadores finais.

Figura 18 - Relações e papéis dos envolventes nos modelos de Cloud Computing



3.3 Resumo

O presente capítulo descreve a tecnologia de Computação na Cloud (*Cloud Computing*), vantagens do seu uso e conceitos tais como perímetro (onde se enquadram designações como Clouds Privadas e Públicas, passando pelas Híbridas); e modelos de Serviço IaaS, PaaS e SaaS, sendo o IaaS o foco desta dissertação.

A virtualização apresenta-se como a tecnologia chave de todo este processo; ela desempenha e continuará a desempenhar um papel indispensável para que os prestadores de serviço continuem a oferecer ambientes de CC que permitem um dimensionamento correcto e economicamente ajustado das infra-estruturas de TI, oferecendo ainda funcionalidades como gestão centralizada, disponibilidade e segurança.

4. HP Cloud Stack

O OpenStack é presentemente: um consórcio que inclui entidades como a AT&T, HP, IBM, Rackspace, Red Hat, Suse, etc, fortemente empenhadas na computação na cloud, e que têm por objectivo colocar no mercado uma implementação ubíqua para clouds tanto públicas como privadas; e uma pilha de software em código aberto, que suporta heterogeneidade (por exemplo, nos hipervisores), e uma enorme riqueza em recursos (de computação, armazenamento e interligação).

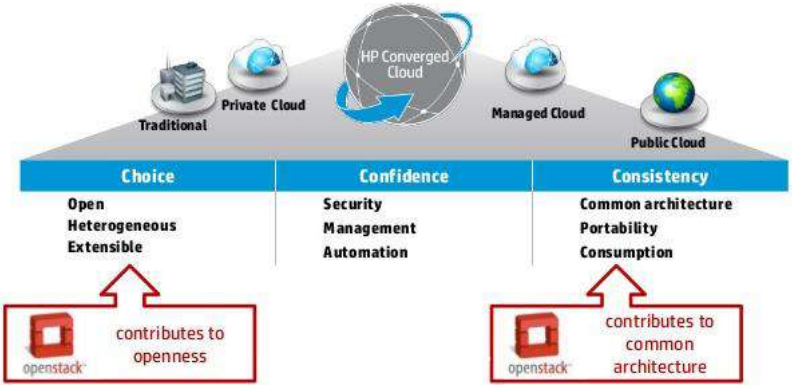
O OpenStack promove uma arquitectura comum, ou seja, um *framework* que é extensível e permite que terceiros (incluindo os parceiros do consórcio) ampliem as suas funcionalidades quer por via de desenvolvimentos adicionais quer por integração de produtos já existentes, embora a pilha-base já permita, por meio de portais, quer a administração das *pools* de computação, armazenamento e rede, quer o provisionamento de recursos.

As plataformas Cloud em Open Source tornam-se atraentes pelo seu baixo custo inicial de implementação e pela perspectiva da portabilidade.

4.1 HP CloudSystem Matrix

A HP CloudSystem Matrix é uma solução proprietária que assenta na plataforma OpenStack para lhe acrescentar módulos de segurança, gestão e automação abrangente a toda a infra-estrutura, como mostra a figura 19. É uma solução IaaS, tanto para *clouds* privadas, como para híbridas.

Figura 19 - Contributos CloudStack.



Norteado por uma visão de "simplificação como chave", o HP Cloud System Matrix permite entregar soluções pré-definidas num catálogo de serviços de TI a consumidores da infra-estrutura (utilizadores) que as requisitaram a partir de um portal self-service.

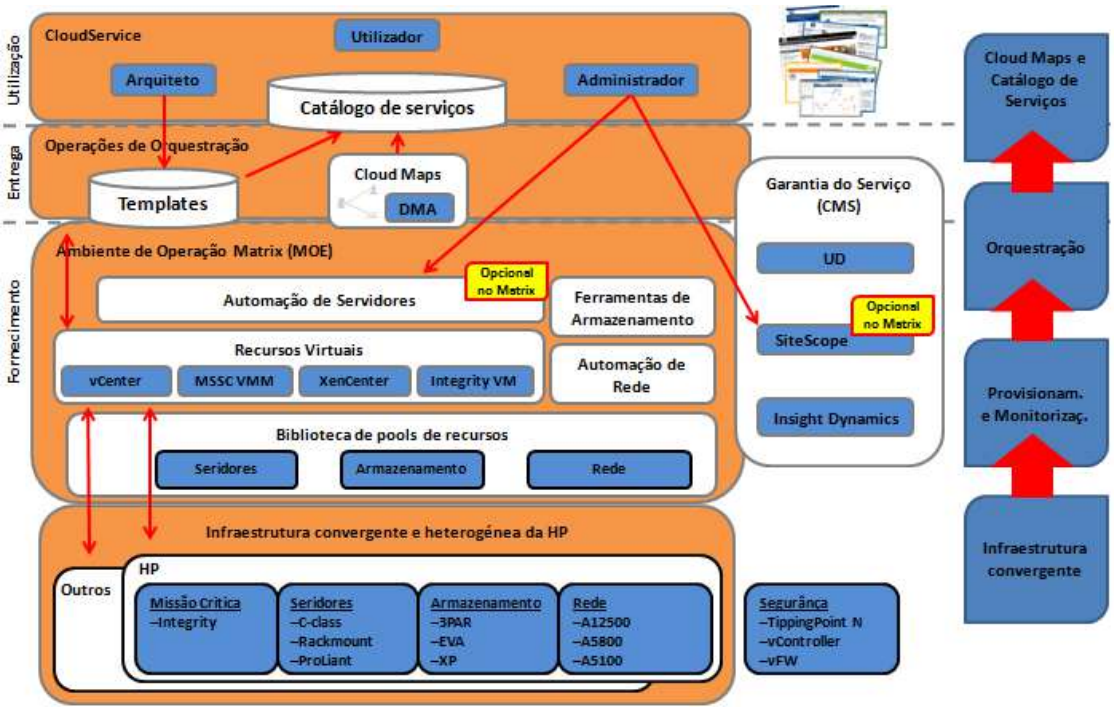
Figura 20 - Desenho, publicação e implementação do catálogo de serviços.

A solução HP CloudSystem Matrix oferece:

- ✓ Um ambiente de trabalho (Matrix Operating Environment - MOE) altamente automatizado, que automaticamente se ajusta a todo o dinamismo de negócio permitindo um rápido desenho, provisionamento e modificação de toda a infra-estrutura, e uma optimização dos ambientes físicos e virtuais a fornecer como serviço.
- ✓ Uma pilha de tecnologias de gestão, figura 21: HP Console Management System (CMS), que inclui ferramentas como HP Universal Discovery (UD) para detectar automaticamente todos os objectos do ambiente; HP SiteScope [25] (opcional), que monitoriza a disponibilidade e desempenho de toda a infra-estrutura - aplicações, servidores, SO, serviços de rede, armazenamento [24] e software de virtualização - evitando e resolvendo alguns problemas de desempenho (estrangulamentos por escassez de recursos).
- ✓ Uma infra-estrutura convergente (HP Converged Infrastructure) apoiada em soluções como o HP Insight Control (IC), para gestão da infra-estrutura, e o HP Insight Dynamics (ID), uma consola de Data Center para uma gestão centralizada de agregados (*pools*) de recursos, e de pacotes pré-configurados de *hardware* e *software*. É aqui que se criam as *pools* de recursos para posterior fornecimento ao MOE.

O MOE assenta nesta pilha como o último nível de software, um ambiente de trabalho altamente automatizado permitindo que seja possível a redução drástica de custos, tempo e esforço na implementação de toda a estrutura necessária, assim como das aplicações, mercê do seu módulo de orquestração da infra-estrutura (Infrastructure Orchestration - IO) e, opcionalmente, do módulo de automação de servidores (Server Automation - SA). Em suma, trata-se de um ambiente optimizado para gestão e aprovisionamento de infra-estruturas híbridas, incluindo servidores físicos (Integrity e x86) e virtuais, rede e armazenamento [23]. O CloudSystem Matrix está igualmente integrado com tecnologias de virtualização como o ESXi da VMware e o Hiper-V da Microsoft.

Figura 21 - Infra-estrutura HP CloudSystem Matrix



4.2 Modelos de interacção com o HP CSM

Definidos os modelos de implementação e serviço, resta definir as formas de comunicação, os modos de interacção entre as infra-estruturas, aplicações e processos. O modelo é transparente para o utilizador final, sendo disponibilizadas três opções:

Web Service API

Este é o método preferido para controlar os processos dentro da TI. Um Web Service (WS) é, de acordo com o W3C[4], "*a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards*". A HP CloudSystem Matrix oferece uma API baseada no protocolo SOAP (Simple Object Access Protocol) para troca de informação estruturada numa plataforma descentralizada e distribuída. É um protocolo baseado numa linguagem de marcação extensível (Extensible Markup Language – XML) para o formato das mensagens enviadas e recebidas e no protocolo de transferência de hipertexto (HyperText Transfer Protocol – HTTP) para negociação e transmissão de mensagens. Como protocolo de transporte, usa igualmente HTTP, garantindo assim uma certa "facilidade" para funcionar em ambientes protegidos por *firewalls*.

[4] http://www.w3.org/TR/ws-arch/#whatis

Command line interface (CLI)

Uma interface de linha de comando que inclui não só todas as opções disponíveis na API WebService, como outras não disponíveis nas outras APIs.

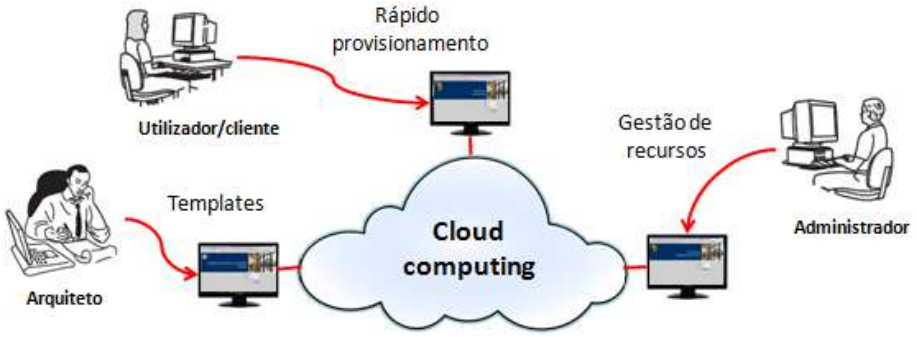
Northbound API (nbAPI)

Uma interface de controlo remoto através de um URL (Uniform Resource Locator), endereço de um determinado recurso disponível numa rede; o acesso pode ser efectuado por HTTP, FTP, etc. (ex. protocolo://máquina/caminho/recurso).

4.3 Papéis e Portais de interacção

Os provedores de serviço *cloud* são responsáveis por fornecer toda a infra-estrutura do serviço contratado. Nestas arquitecturas existem normalmente três tipos de (papéis de) utilizadores da *cloud*, que se associam aos respectivos portais, figura 22: Administrador, Arquitecto e Utilizador final (ou consumidor de serviços *cloud*).

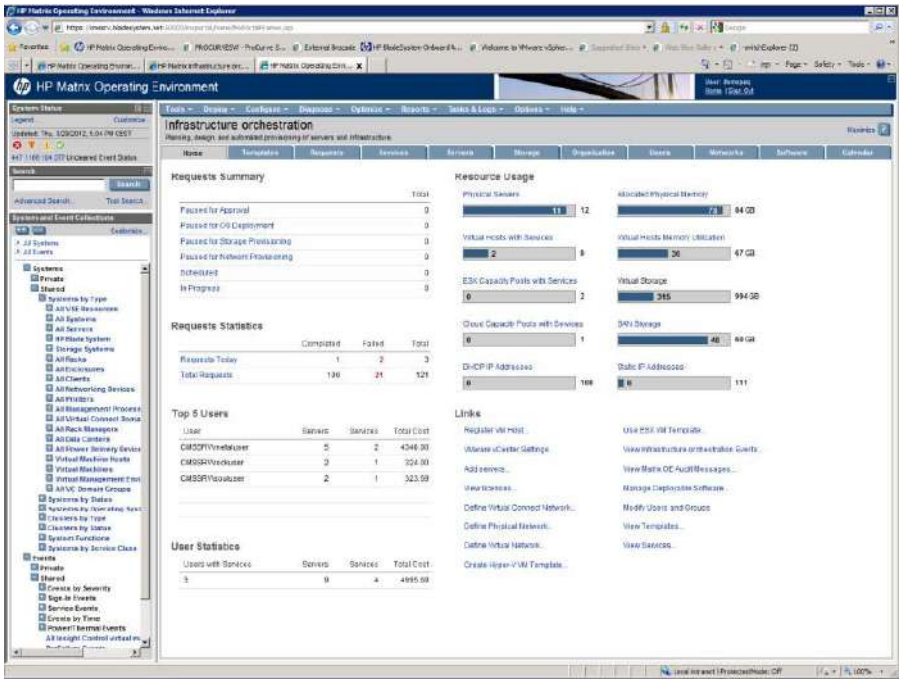
Figura 22 - Utilizadores de TI em cloud e respectivos papéis (roles)



Administrador Cloud

O administrador, usando as ferramentas de ao seu dispor, cria, monitoriza e mantém a pool de recursos (de computação, memória, armazenamento e rede) a disponibilizar na cloud. Normalmente, este papel também envolve a implementação de acordos de nível de serviço (Service Level Agreement - SLA) onde são definidas as políticas entre o fornecedor de serviços cloud e os seus "clientes". O portal associado a este utilizador é o portal do Administrador, figura 23.

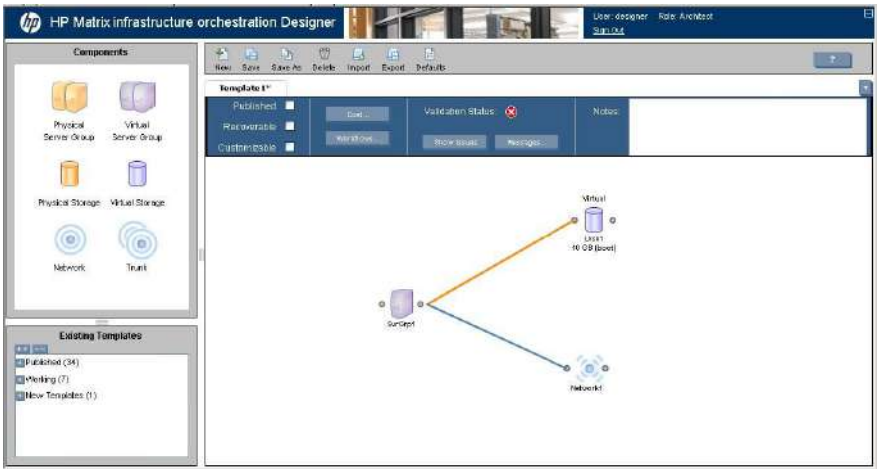
Figura 23 - Portal do Administrador



Arquitecto Cloud

O arquitecto é um utilizador técnico, responsável por projectar/desenhar, montar e tornar disponíveis (publicar) serviços que a própria infra-estrutura pode disponibilizar para os utilizadores, mediante políticas e necessidades dos objectivos estratégicos de uma organização. Esses serviços podem ser personalizáveis e específicos para um determinado utilizador, grupo de utilizadores ou para uso geral. O arquitecto pode ainda criar fluxos de trabalho (*workflows*) que, passo a passo, ajudam na interacção utilizador-ambiente, oferecendo serviços como personalização, gestão, simplificação de processos e a própria integração de TIs. O portal associado a este tipo de utilizador é o *Designer Portal* onde o arquitecto desenha a infra-estrutura e os modelos de serviço (*templates*, de que adiante falaremos) a incluir no catálogo.

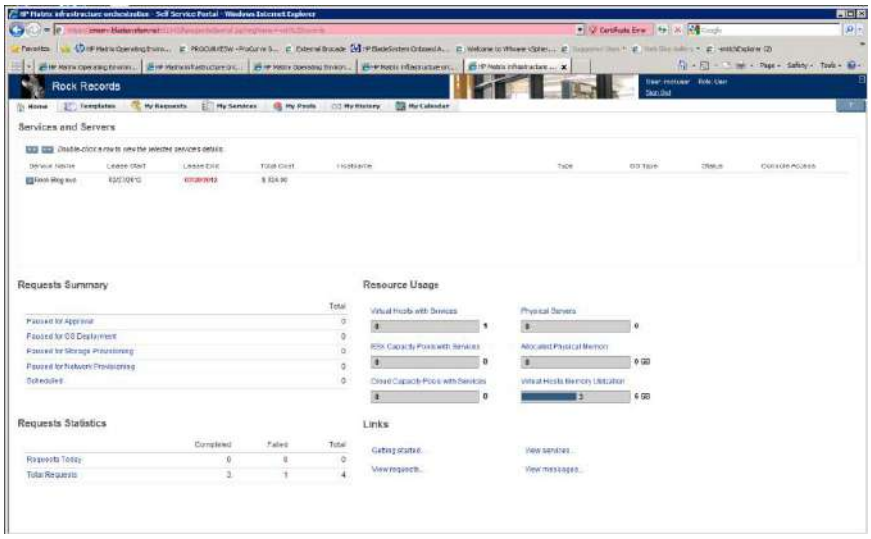
Figura 24 - Portal do Arquitecto



Consumidor IaaS

O portal associado a este tipo de utilizador é o *Self-service Portal* (ou *user portal*) onde o utilizador poderá seleccionar, visualizar e instanciar os modelos de serviço pré-definidos pelo Arquitecto. O utilizador, depois de escolher de entre os que existem no catálogo de serviços, o "modelo de infra-estrutura" que mais se adequa às suas necessidades, solicita a sua instanciação para obter os recursos de computação, memória, armazenamento e rede. Este portal permite ainda ao utilizador efectuar a gestão dos serviços solicitados.

Figura 25 - Portal do Utilizador final



4.4 Conceitos fundamentais da plataforma HP CSM

Apresentamos agora um leque de conceitos fundamentais para compreender o HP CSM e o trabalho que realizámos. Nalguns casos poderemos apresentar exemplos concretos para melhor ilustrar esses conceitos.

Template

Num *template* o Arquitecto da infra-estrutura captura uma configuração típica em TI, especificando os seus componentes-base (hardware e software) e desenha as suas inter-relações: que servidores existem, quais as suas características (CPUs, memória, discos, interfaces), como se interligam os servidores entre si (as conexões de rede) e, se for o caso, como um ou mais desses servidores se ligam ao "exterior" do Centro de Dados. Do template também faz parte o software: haverá servidores com Linux, outros com Windows Server, uns podem ter Apache e outros SQL Server, para dar alguns exemplos. Os templates, após publicação, passam a fazer parte de um catálogo de serviços, para uso dos consumidores de serviços dessa infra-estrutura de cloud.

Como exemplo, vejamos a seguinte situação: em TI, a pilha LAMP (Linux, Apache, MySQL, PHP) é um padrão comum e bastante usado; assim, o arquitecto poderá criar dois templates: *LAMP-base*, com um único servidor e toda a pilha nele instalada, e *LAMP-elástico*, em que há uma *farm* de servidores Apache-PHP que se ligam a um "*failover cluster*" de dois servidores MySQL.

Server Group

Um "Grupo de Servidores" é constituído por um ou mais servidores com as mesmas características e configurações: software, discos, interfaces, etc. O arquitecto usa os Grupos de Servidores para organizar os servidores em torno da função que estes desempenham e/ou das suas características.

Por exemplo, o arquitecto definiria no template *LAMP-base* um único grupo de servidor(es), mas já no caso do template *LAMP-elástico*, definiria o grupo *APfarm* (que incluiria um proto-servidor com determinadas características – 2 interfaces LAN, Linux, Apache e PHP) e o grupo *MySQLfo* (com um proto-servidor também com 2 interfaces LAN, mas com Linux, Linux-HA, e MySQL). Só que o grupo *APfarm* permitiria um mínimo de 1 e um máximo de, digamos, 4 servidores, enquanto o grupo *MySQLfo* seria instanciado sempre com 2 servidores.

Server

É obviamente um servidor (virtual, quase sempre, embora o HP CSM permita "instanciar" – leia-se, instalar o software e configurar – servidores físicos), com atributos (muitos deles especificáveis por um intervalo de valores entre um mínimo, o *default*, e um máximo) como memória (RAM), número de processadores e correspondente velocidade de relógio, discos, lista de software associado, interfaces de rede (LAN e/ou SAN), etc.

Software

O termo "Software" refere-se a um produto software instalável no servidor (pode ser o SO, um SGBD, um qualquer *middleware*, ou até uma aplicação). Há duas formas de efectuar essa instalação: automaticamente no acto de instanciação do servidor, usando um pacote opcional, o HP Server Automation (SA), ou manualmente (a usada neste trabalho, por não dispormos do HP SA) – e neste caso temos de criar uma VM, instalar o software, e depois transformar a VM numa VM-template (o que se faz directamente no software de virtualização, e.g., VMware vCenter Server).

Disk

É um disco, virtual ou físico, que pode ser acedido pelo(s) servidor(es); pode ser *bootable*, shareable (partilhável entre servidores), suportado numa tecnologia magnética ou SSD, acessível por ligação FC ou iSCSI, etc.

Interface

Uma "interface" é um ponto de contacto entre o servidor e as redes (que lhe são) externas; de um ponto de vista pragmático, deixando de lado infra-estruturas

pouco comuns em centros de dados de TI (por contraposição aos CDs HPC), as infra-estruturas relevantes são Fibre Channel (FC) e Ethernet. Assim, as interfaces em questão são HBAs (FC) e NICs (Ethernet).

Service

Um *Service* corresponde a uma instanciação de um *template*, e herda deste todas as configurações base, que poderão ser posteriormente editadas com mais ou menos servidores, processamento, memória, discos, etc. até ao limite máximo definido no Template.

A criação do serviço é efectuada pelo consumidor[5] IaaS, que faz a definição fina das características da infra-estrutura que quer aprovisionar. Continuando com os exemplos baseados na pilha LAMP, um consumidor, do departamento de TI, poderia num dado momento criar um serviço *LAMP-testes* baseado no template *LAMP-base*, e que usaria para fazer testes ao "produto" que está a desenvolver e solicitar ao administrador da cloud o respectivo aprovisionamento, e, mais tarde, criar um serviço *LAMP-produção* baseado no template *LAMP-elástico*, e que usaria para colocar on-line o produto.

4.5 Limitações do HP CloudSystem Matrix

Em resumo, o HP CloudSystem Matrix, é um *software stack* para a implementação de *clouds* privadas que oferecem infra-estruturas como serviço (IaaS), isto é, permite aos utilizadores (consumidores) dos seus serviços aprovisionar de uma forma fácil e rápida (minutos) uma infra-estrutura computacional completa de servidores[6] (CPU, memória e discos) com o SO já instalado, interligados por uma topologia de rede mais ou menos complexa.

Contudo, o supracitado "aprovisionar de uma forma fácil e rápida (minutos) uma infra-estrutura computacional completa de servidores" só é, de facto fácil, se baseada na instanciação de um *template* que serve exactamente as necessidades do utilizador, já que na instanciação apenas se pode modificar, para cada servidor, o número de CPUs e a quantidade de memória, deixando de fora, por exemplo, opções de rede e/ou de armazenamento: discos físicos vs. virtuais, aprovisionamento *thin* vs. *thick* (ver 2.3), escolha da tecnologia da interface (SATA vs. SAS vs. FC) ou do disco (magnético vs. SSD).

Por outro lado, a interacção através do portal *self-service* do utilizador também apresenta algumas desvantagens: a informação apresentada pode ser em muitos casos considerada demasiadamente técnica e/ou desnecessária, sendo que o portal não pode ser configurado para a omitir.

[5] Muitas vezes designado utilizador-consumidor, ou simplesmente utilizador – e nesse caso é importante não o confundir com um mero utilizador final.

[6] Uma das vantagens do HPCSM é permitir aprovisionar não só servidores virtuais, mas também reais (estes últimos têm obrigatoriamente de ser fabricados pela HP)

4.6 Resumo

Este capítulo descreve a *HP CloudSystem Matrix* uma pilha de software que implementa o modelo IaaS para clouds privadas, e faz parte de um leque mais vasto da oferta HP que passa pela CloudSystem Enterprise (clouds privadas e híbridas) e culmina com a CloudSystem System Provider (que acrescenta as clouds públicas).

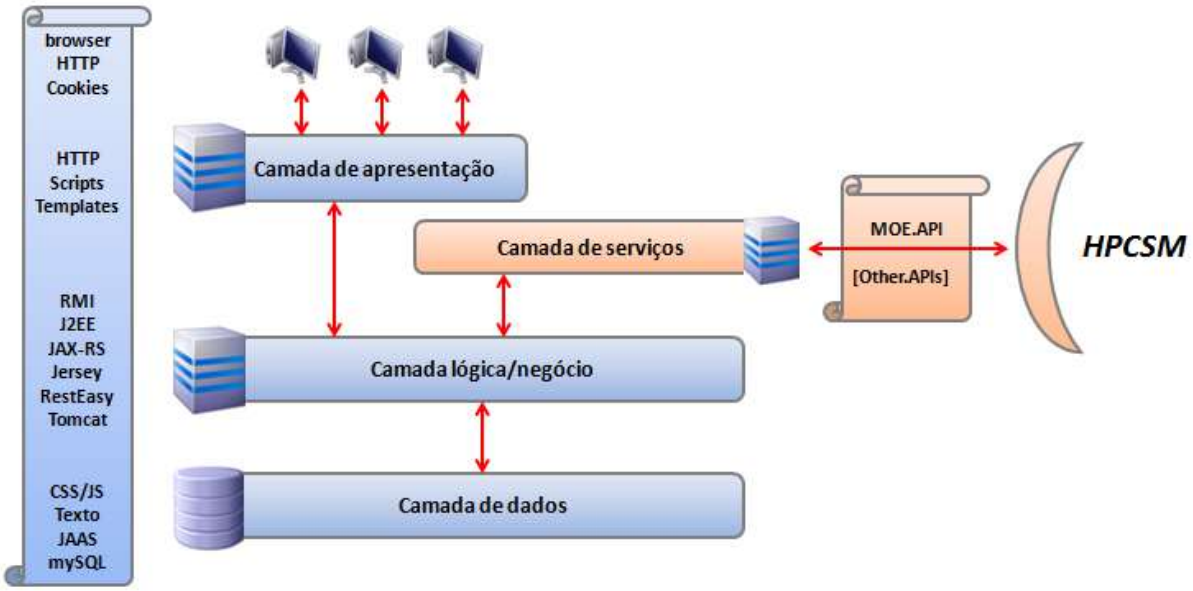
A *HP CloudSystem Matrix* é o núcleo de trabalho deste *framework*, razão de ser deste estudo e deste capítulo. Todo o hardware e pilha de software em que assenta toda a infra-estrutura *HP CloudSystem Matrix* foi descrita, nomeadamente os seus três Modelos de interacção (API, CLI e nbAPI), os papéis e portais de interacção (Administrador, Arquitecto e Utilizador). Concluímos com uma abordagem modular da infra-estrutura com a explicação dos módulos principais e finalmente as limitações desta infra-estrutura, em particular do portal actualmente em uso.

5. O framework e a sua Realização

5.1 Arquitectura

A arquitectura proposta assenta em 3 camadas (3-tier): apresentação, lógica e dados; este tipo de arquitectura é bem conhecida apresentando, como resultado da modularidade, uma série de benefícios que incluem a escalabilidade que resulta da possibilidade de executar uma ou outra camada num servidor distinto, ou até, em certos casos, numa *farm* de servidores. A comunicação entre as camadas apresentação e lógica/serviços assenta num protocolo REST (ou seja, a arquitectura é orientada à Web (WOA) com Axiom0a [28] - Universality2, com um URI por cada recurso importante) e dados em formato XML (visualização) ou JSON (programação) transportados em HTTP sobre TCP/IP. Relativamente à comunicação com o HP CSM, ela processa-se sobre HTTPS, numa arquitectura SOA, com o protocolo SOAP.

Figura 26 - Arquitectura n-tier



Camada de Apresentação

Define e realiza a interface do utilizador, o formato de interacção e visualização, foi optimizada para o navegador Google Chrome mas funciona igualmente noutros browsers. É a camada de mais alto nível e é responsável não só pelo contacto utilizador/aplicação através de um conjunto de interfaces que podem ser apresentadas num browser mas também por lhe fazer chegar a informação das camadas inferiores.

Camada Lógica/negócio

Também conhecida como a camada de negócio, é o núcleo da aplicação e é responsável por controlar as suas funcionalidades; tem igualmente a responsabilidade de realizar a transição de dados entre as camadas de apresentação e dados e ainda todas as transições necessárias entre a própria aplicação e a camada de serviços.

Camada de Serviços

Camada que realiza toda a comunicação com os webservices do CSM, nomeadamente via API do MOE, e outras APIs, como é o caso da API da VMware.

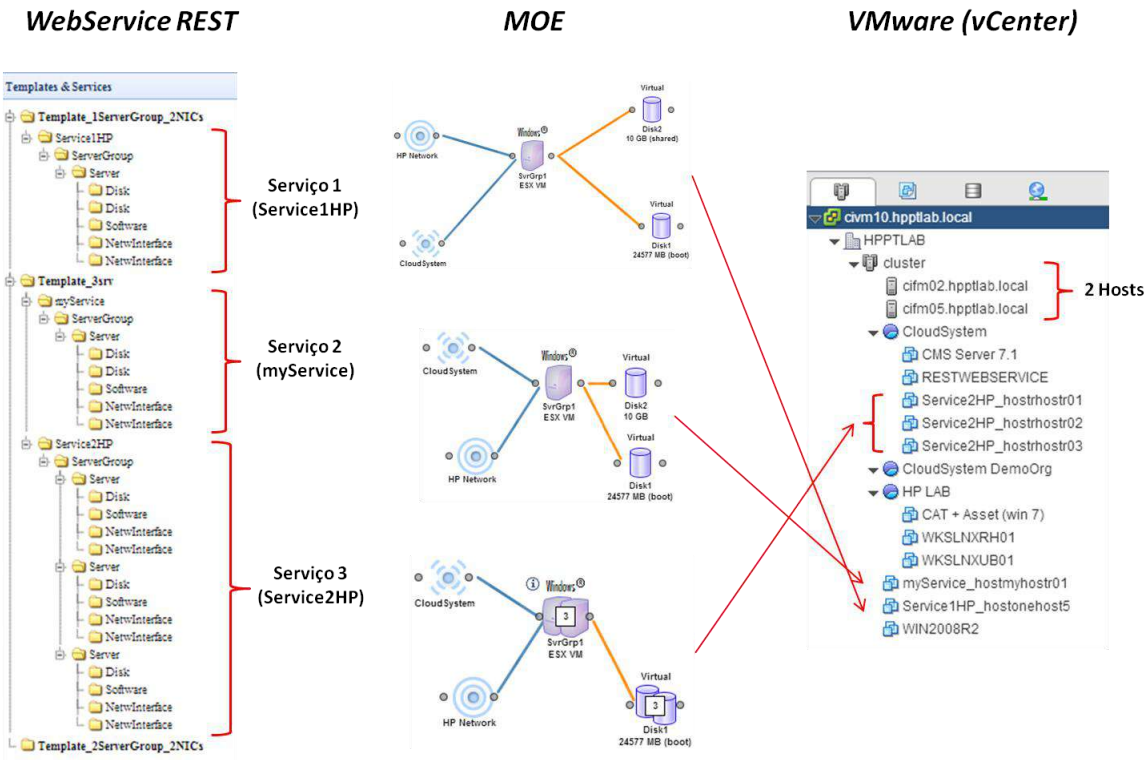
Camada de Dados

Camada responsável por guardar e gerir os dados em texto (ou numa base de dados, opção que activamos quando queremos trabalhar offline – o que neste momento apenas fazemos para testes) e *cache* de leituras (cookies) com a informação sobre o utilizador e outras informações proveniente da camada de lógica/serviços como é o caso da implementação Realm JAAS no módulo de login JAAS.

5.2 Integração da framework no HP CSM (MOE e VMware)

Nesta secção, propositadamente muito simples, mostramos (figura 27) as relações entre a framework, à esquerda, a plataforma HP CSM (MOE) com o qual esta interage, ao centro, e a plataforma VMware vCenter que gere o hipervisor ESXi que, em ultima instância, é quem suporta as VMs que se disponibilizam na cloud.

Figura 27 - Relação dos VMIDs (WebService-MOE-vCenter)



5.3 Um desenho guiado pela Interface Utilizador

"A portal is a window into a much broader set of capabilities in the IT world. It's critical that portals be user-friendly and fairly intuitive to use"

Ken Stephens, vice president of Xerox Corp

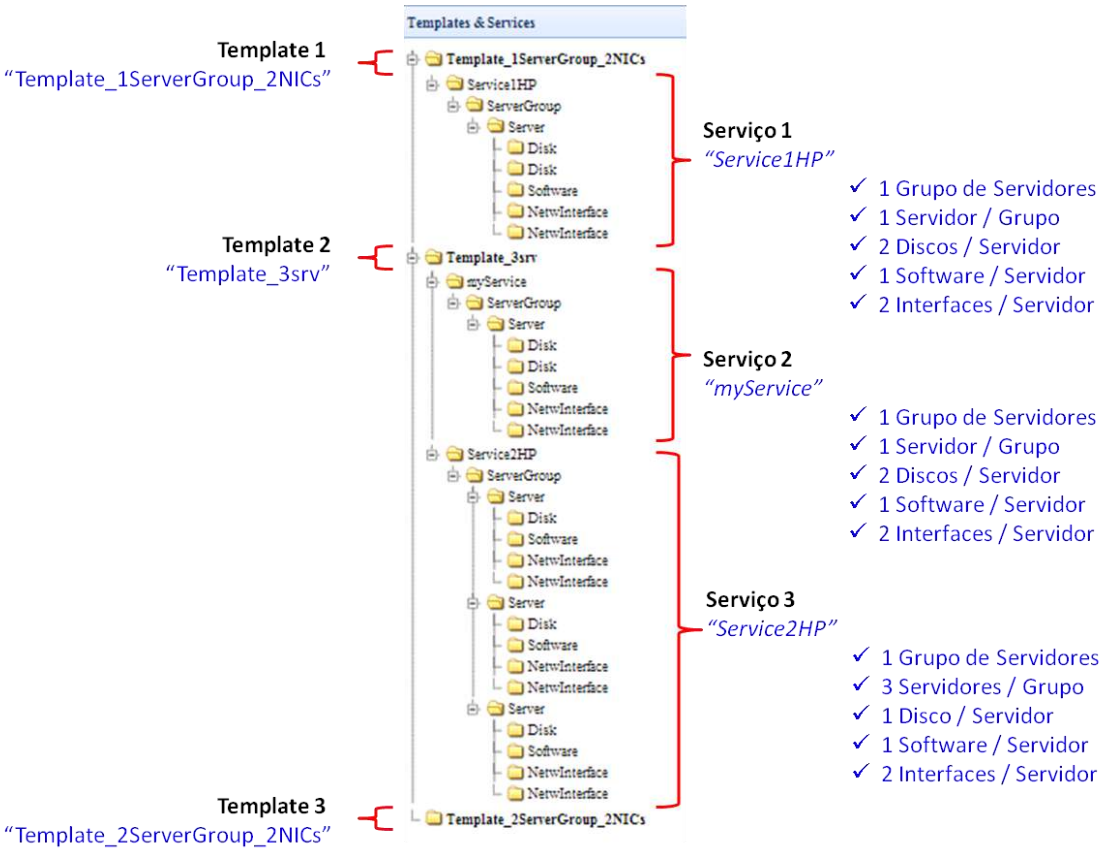
O desenvolvimento deste *framework* teve sempre como objectivo último o utilizador com acessos, consulta de informação e interacções rápidas, intuitivas, fáceis e acima de tudo práticas. Pretendeu-se simplicidade, com uma curta curva de aprendizagem e uma flexibilidade com acessos próprios e individuais a cada utilizador, associados a um *username* e *role* (admin ou user), dispensando a necessidade da se ter de usar dua interfaces distintas, uma para o administrador outra para um utilizador-consumidor de serviços.

"These users want to see performance graphs and have the capability to build complex configurations and templates for their end users"

Lauren Nelson, Forrester Research Inc

A visualização em tempo real de todas as acções do utilizador, com a disponibilização de uma "barra de progresso", a visualização do estado geral do webservice e das suas configurações/recursos, também não foram descuradas. A descrição dos componentes principais em árvore permite a rápida visualização das configurações de todo o Serviço criado, das suas hierarquias e relações, e um acesso rápido e intuitivo aos detalhes de cada componente.

Figura 28 - Estrutura em árvore do menu lateral esquerdo



> *"Developers want a clean, easy-to-use [product] to select prebuilt templates -- but they don't need to see any of the complex performance or payment configurations"*
>
> Lauren Nelson

O Catálogo de Templates publicados pelo Arquitecto é imediatamente visualizado, no menu em árvore, quando o webservice é inicializado, que é imediatamente povoada com os Serviços (e seus recursos) anteriormente criados por esse utilizador.

Para criar um novo Serviço o utilizador seleciona o template que mais se adequa às suas necessidades, e cria o Serviço. Todos os Serviços associados a um determinado Template ficam como ramos na árvore desse Template, o mesmo acontecendo relativamente aos recursos associados a cada Serviço. A informação visualizada é apenas a necessária a cada visualização e instanciação.

A interface é assim rápida, acessível, intuitiva, organizada e personalizável; as acções são acessíveis e rápidas de executar, sem menus e submenus com sucessivos cliques que só atrasam todo o processo.

Figura 29 - Portal

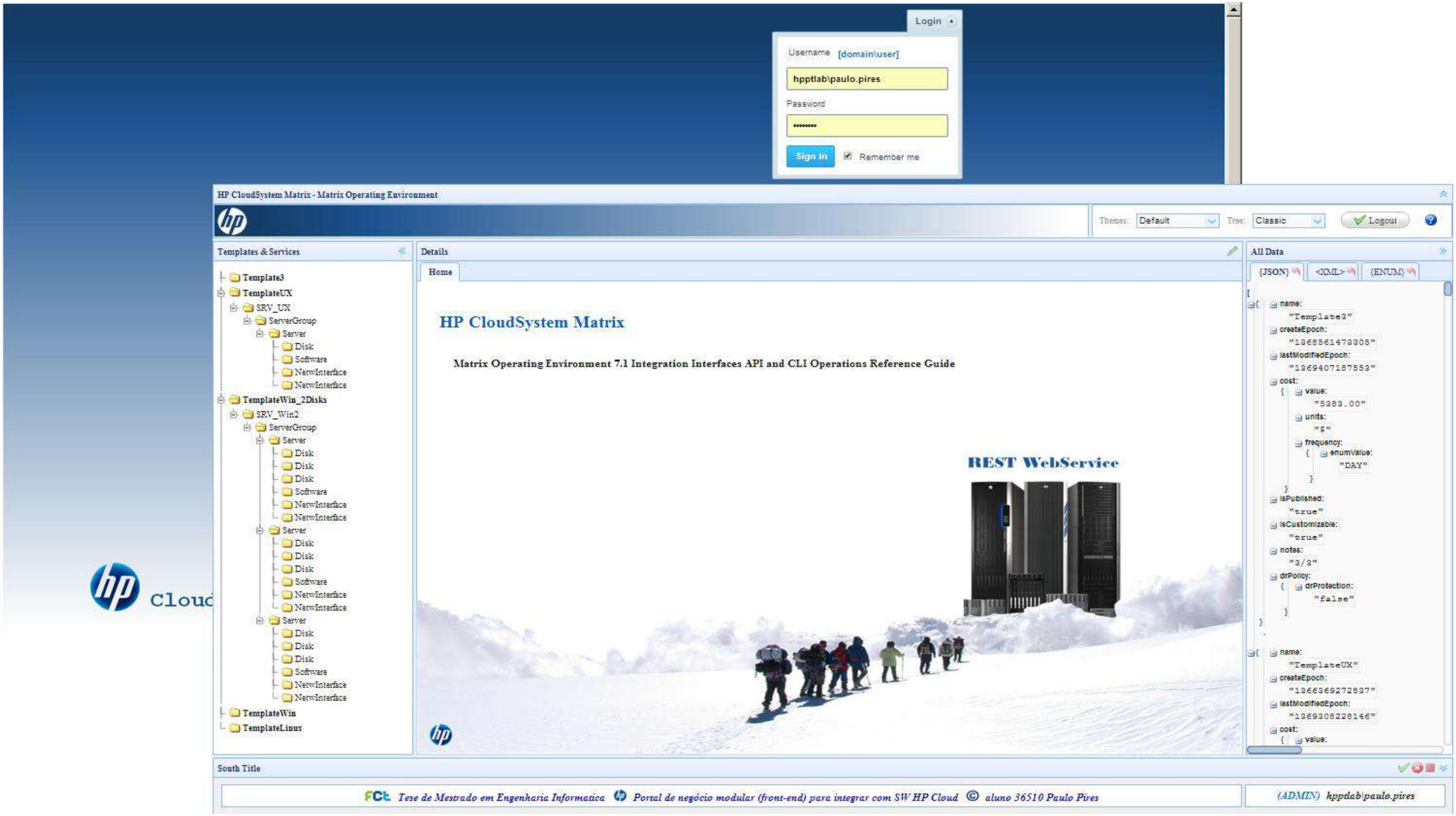
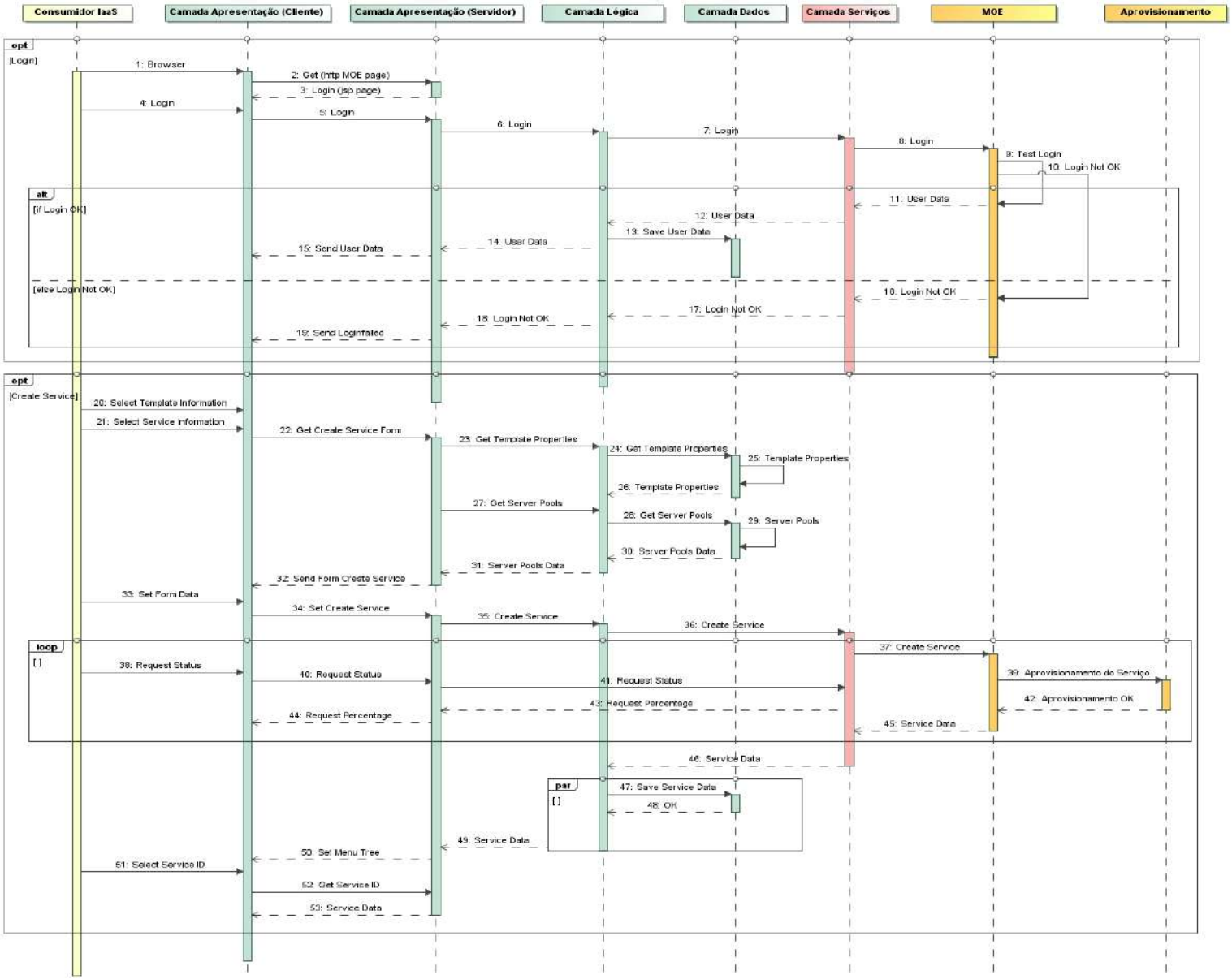


Figura 30 - Diagrama de Sequência (Criação de um Serviço)



"What you want and what you get in most cases depends on the user's persona -- either an IT manager or a developer… By making portals task-based, you're not just seeing a bunch of random menus or links. If you're searching for something, you'll find it quickly"

Bill Forsyth, vice president of portal development at Savvis

Um portal personalizável, em função do utilizador: ocultar ou exibir painéis de informação, a lista de pedidos, limitar a exibição por Template ou por Serviço - um simples clique, e todo o portal é personalizável.

"Paying attention to details and being able to organize information for larger enterprises is important"

Bill Forsyth

A organização de toda a informação a disponibilizar é a chave de todo este processo e diferencia um portal agradável e intuitivo de um portal confuso no seu amontoado de informação e complexidade; quanto maior é a informação a disponibilizar, maior é evidente esta diferença. Uma informação hierarquizada e organizada em agrupamentos lógicos por recursos é um processo muito fácil de seguir e perceber, e associada a esta disponibilização de informação, apenas exibir acções que possam ser executadas sobre esse(s) recurso(s) é o objectivo da simplicidade e desempenho de uma interface. Por último, uma página web de ajuda a toda a interface (Anexo N) para consulta de qualquer dúvida, faz com que tudo seja útil e prático [33].

5.4 Camada de apresentação

A realização da camada de apresentação pode ser apresentada subdividindo-a em duas partes, sendo a primeira correspondente à realização do lado do cliente (*client-side*), e a segunda à realização do lado do servidor (*server-side*).

5.4.1 Do lado do cliente

A visualização de uma página *web* num navegador como Google Chrome, Firefox ou Internet Explorer só é possível se estes "entenderem" a linguagem na qual está construída "a página" e forem capazes de a processar sem necessidade de qualquer espécie de pré-tratamento. Surgiram muitas linguagens conhecidas como linguagens de marcação, um sistema moderno para anotação de um documento de uma forma que "o programa" é sintaticamente distinguível do texto normal.

Do lado do cliente a principal linguagem de marcação é a HTML, à qual se adiciona a JavaScript, na qual os programas acompanham um documento html incluídos em ficheiros com a extensão `.js`, ou são incorporados directamente no

código html entre *tags* <script></script>; o documento html é então carregado pelo navegador do utilizador, e com ele todo o código JavaScript.

A linguagem JavaScript é usada para incutir dinamismo no código html facilitando, por exemplo, a interacção com o utilizador no preenchimento de formulários (aquando da criação de um serviço) ou promovendo o *upload* de um modelo de template em formato XML, ou executando acções ou eventos consoante determinado comportamento do utilizador, etc.. As funções JavaScript e das suas bibliotecas como JQuery são executadas aquando do carregamento da própria página html, ou quando invocadas por eventos como sejam a inicialização da framework, leitura inicial dos dados/recursos relativos ao utilizador, criação de cookies, ou acções do utilizador como o *login* ou o preenchimento de formulários.

Também é usado AJAX (Asynchronous JavaScript and XML) [36] um padrão para o uso de comandos JavaScript para, de uma forma síncrona ou assíncrona, trocar dados/informação com o servidor e actualizar conteúdos de páginas web já exibidas, em que não é necessário carregar toda a página para a actualizar. Um padrão ideal para usar com o protocolo REST (Representational State Transfer) para troca de informação, permite o uso de dados em texto, HTML, XML e JSON (JavaScript Object Notation) como formato da informação para enviar e receber dados, usando HTTP e URI, exactamente como as interacções HTTP foram concebidas: GET, POST, PUT e DELETE. Com AJAX, as operações CRUD (Create, Read, Update e Delete) são muito intuitivas e fáceis de usar.

Finalmente, CSS (Cascading Style Sheets), é a linguagem responsável pela formatação dos dados em termos de visualização para o utilizador que é usada para descrever a apresentação das páginas Web, incluindo cores, layout e fontes.

5.4.2 Do lado do servidor

As linguagens caracterizadas como “do lado do servidor” são necessárias para especificar os programas que têm de ser executados no servidor como “resposta” às acções desencadeadas do lado do browser; quando o resultado tem de ser enviado para o browser, é-o na forma de ficheiro HTML.

Linguagens de conteúdo e comportamento como JSP (JavaServer Pages), uma mistura de HTML com scriptlet, contêm um elemento script (blocos de código em Java), o que as caracteriza como linguagens que correm no servidor. Os scriptlet, como todo o código Java é executado no servidor e o output direcionado para o resto da página JSP e finalmente apresentado ao utilizador. A facilidade de comunicação em variáveis de sessão, torna esta linguagem muito apelativa ao uso dessas variáveis como é o cado da página login.jsp onde se desenrola grande parte da comunicação browser/servidor através da API MOE, logout.jsp e vmrc.jsp para ligação às VMs e comunicação com a API VMware.

5.4.3 Lista de primitivas na óptica dos recursos

Segue-se a lista de primitivas, apresentada numa óptica de recursos que cada uma manipula:

- ✓ *actions.js* - Lista as "actionsGrid" atribuídas aos Serviços, Grupo de Servidores ou Servidores e permite todas as acções associadas nestas listagens.
- ✓ *cookies.js* - Leitura e criação de cookies associados ao utilizador e todo o webservice. Cookies de login com dados como username e role e outros dado associados às acções do utilizador e webservice como layout de visualização da página (globaltheme), menu lateral esquerdo (treetheme), painéis de visualização de informação (northregion, westregion, eastregion, southregion), etc.
- ✓ *disks.js* - Leitura das informações associadas aos discos nas suas três versões "stereotype disks", discos virtuais e físicos. Todas as acções possíveis em discos estão igualmente configuradas neste ficheiro.
- ✓ *geral.js* - Leitura das informações gerais associadas ao webservice. Dados iniciais de informação de todos os Templates do Catálogo de Serviços e dados de Serviços já existentes nos diferentes formatos (JSON e XML), configurações gerais do webservice como tabelas de listagens, conversões de datas ("date2epoch"/"epoch2date"), construção do menu lateral esquerdo em árvore, listeners para acções do utilizador e mensagens de erros e informação do webservice.
- ✓ *help.js* - Configurações de todo o layout das ajudas ao utilizador.
- ✓ *interfaces.js* - Leitura das informações associadas às interfaces, redes e endereços IP. Todas as acções passíveis de serem executadas nas interfaces estão igualmente configuradas neste ficheiro.
- ✓ *remoteSessions.js* - Listagem de todas as Sessões Remotas do Serviço e acções associadas a essas Sessões Remotas por Servidor e suas Interfaces.
- ✓ *requests.js* - Listagem de todos os pedidos, por Template ou Serviço. Todas as acções passíveis de serem executadas relativas aos pedidos do utilizador estão igualmente configuradas neste ficheiro.
- ✓ *serverGroups.js* - Leitura das informações associadas ao Grupo de Servidores quer nos Templaste, quer nos Serviços. Todas as acções relativas ao Grupo de Servidores estão igualmente configuradas neste ficheiro.
- ✓ *serverPools.js* - Listagem de todos as Pools de Servidores com a informação respectiva a cada Pool. Todas as acções passíveis de serem executadas nas Pools de Servidores estão configuradas neste ficheiro.

- ✓ *servers.js* - Leitura das informações associadas a cada Servidor quer nos Templaste, quer nos Serviços. Todas as acções relativas aos Servidores estão configuradas neste ficheiro.
- ✓ *services.js* - Leitura das informações associadas a cada Serviço. Todas as acções passíveis de serem executadas nos Serviços estão configuradas neste ficheiro.
- ✓ *snapshots.js* - Listagem de todos os Snapshots com a informação respectiva ao par Server/Snapshot. Todas as acções passíveis de serem executadas nos Snapshots estão configuradas neste ficheiro.
- ✓ *software.js* - Listagem de todo Software existente no sistema e informação relativa a cada Software em particular.
- ✓ *templates.js* - Listagem de todos os recursos do Template e informação relativa a cada recurso base. Toda a disponibilização de informação e as acções associadas aos Templates estão configuradas neste ficheiro.

5.5 Camada Serviços/Lógica

Como referido (ver 5.1) a comunicação entre a camada de apresentação e a lógica/serviços assenta num protocolo REST e usa XML e JSON como formato para enviar e receber informação, usando HTTP e URI (operações CRUD - Create, Read, Update e Delete) [29].

Figura 31 - Obtenção de recursos através da API RESTfull

Por seu lado, o REST assenta em quatro princípios fundamentais: 1) *Recurso:* cada recurso tem uma identificação única e global associada ao URI: http://exemplo.com/encomenda/2012/10/**88563**; 2) *Acesso:* usando URIs, acede-se a um recurso por um ID único e global, independente da sua localização; 3) *Formatos:* um recurso pode ser representado em diferentes formatos, por exemplo, HTML, XML ou JSON, sendo a escolha ditada pela finalidade específica de utilização do recurso; 4) *Comunicação: stateless* - o cliente pode receber toda a informação necessária num primeiro contacto com o servidor, executar o processamento, e finalmente ligar-se uma segunda vez ao servidor para reportar os resultados.

As APIs usadas e directamente relacionadas com serviços REST foram as seguintes: 1) *JAX-RS (JSR 311 - Java Specification Requests):* Atualmente parte

integrante do Java EE, é uma API Java para serviços Web RESTfull sobre HTTP; JAVAX-RS como especificação desta API [30], fornece anotações que simplificam as configurações; 2) *Jersey:* é uma implementação open-source de referência do JAVAX-RS [31]; 3) *RestEasy:* é uma outra implementação do JAVAX-RS, com opções de suporte para outros protocolos além do HTML e que permite conexões assíncronas, e tem suporte de cache [32].

5.5.1 Camada de Serviços

Como sabemos, o HP CloudSystem Matrix gere recursos de computação, armazenamento, rede e software de uma forma unificada, para oferecer um ambiente virtualizado e altamente automatizado. Contudo, não é um produto fechado, disponibilizando não só um CLI, mas também uma API com uma WSDL (Web Service Description Language) que inclui todas as operações possíveis (https:// <cms-ip-address>: 51443/hpio/controller/soap/v5 wsdl?).

A camada de serviços é um cliente do CSM que funciona segundo o padrão autenticação/perfil-pedido(s)-fecho: na fase de autenticação o utilizador (registado na infra-estrutura MOE) faz o seu *login*; a partir desse momento está criada uma sessão que, por HTTPS, permite o acesso a todos os recursos disponíveis para a combinação utilizador/perfil (*role*).

5.5.2 Camada Lógica

A camada lógica segue o mesmo princípio, está centrada nos recursos; em termos de tecnologias, usa o protocolo REST para a comunicação cliente/servidor, e a linguagem Java para o desenvolvimento.

A realização foi idealizada numa prespectiva hierárquica, o que facilita a compreensão de toda a estrutura. O menu lateral esquerdo é exemplo disso mesmo, a estrutura em árvore espelha esta ideia, simples mas eficaz, os recursos começam por ser apresentados como listas, sobre as quais podemos realizar acções/funções "de grupo", pois são apenas aquelas que são comuns a todos os membros da lista, mas, à medida que nos vamos aproximando das folhas da estrutura em árvore, as acções/funções ficam mais específicas relativamente aos recursos-alvo.

Por exemplo, quando se realiza a acção "Power ON" a um grupo de servidores, todos os servidores do grupo são ligados; quando se efectua essa mesma acção sobre um servidor, apenas o servidor seleccionado é ligado.

A disponibilização de informação relativa a grandes "conjuntos de dados" como é o caso de listagens de pedidos, de acções, de sessões remotas, de snapshots, de *stereotype disks*, de *software* e de *subnets* foi realizada com *datagrids* (tabelas) para melhor visualização.

As acções/funções disponibilizadas estão distribuídas pelas *tabs* respectivas, e executam serviços no nível em que se encontram disponibilizadas. A título de exemplo, as acções que se encontram disponíveis para execução no grupo de servidores apenas podem ser encontradas no tab "Group", e reflectem-se sobre todos os membros do grupo. Uma acção específica de servidor, encontra-se somente disponível na *tab* "Server" e afecta, apenas e só, esse servidor.

Segue-se uma descrição, não exaustiva, da camada lógica, núcleo desta *framework*, desde a sua inicialização até aos recursos e módulos principais.

5.5.2.1 Init

Aquando da inicialização da aplicação e no seguimento do login do utilizador, é desencadeada uma sequencia de pedidos ao servidor por informações relativas aos Templates do Catálogo de Serviços e de todos os Serviços do utilizador. Se as permissões associadas ao utilizador for de simples utilizador do webservice, visualiza apenas as instâncias que criou, já um utilizador com permissões de administração, visualiza os seus serviços e todos os serviços dos restantes utilizadores.

Tabela 1 - Funções associadas à inicialização do Webservice

Função	Descrição
getVersion	Função de teste. Obtém a informação da versão da API.
principal	Função usada aquando do login do utilizador. Obtém os detalhes do utilizador como USERNAME e ROLE.
listAllDataDetailsJson	Função usada após login do utilizador e iniciação da aplicação. Obtém os detalhes dos Serviços associados ao Utilizador. Lista a informação de todos os Templates e Serviços do Utilizador no formato JSON. Função para uso excluivo do webservice.
downloadTabJsonView	Menu lateral direito do webservice. Opção de download da informação relativa aos recursos do Utilizador em formato JSON. É criado um ficheiro com extenção JSON.
downloadTabXmlView	Menu lateral direito do webservice. Opção de download da informação relativa aos recursos do Utilizador em formato XML. É criado um ficheiro com extenção XML.

listEnumsJson	Função usada após login do utilizador e iniciação da aplicação. Obtém os detalhes de todos os ENUMS possíveis no webservice. Lista a informação de todos os ENUMS no formato JSON. Função para uso excluivo do webservice.
listEnumsXml	Menu lateral direito do webservice. Opção de visualização da informação relativa às opções ENUMS possiveis no webservice.
downloadTabEnumsView	Menu lateral direito do webservice. Opção de download da informação relativa às opções ENUMS possiveis no webservice. É criado um ficheiro com extenção XML com a informação visualizada.
listAllDataDetailsXml	Menu lateral direito do webservice. Opção de visualização da informação relativa ao Utilizador em formato XML.

5.5.2.2 Template

Como já referido, o desenho/construção dos Templates, fica a cargo do Arquitecto do MOE que desenha esses mesmos Templates mediante pedido especifico de um determinado utilizador, tendo em vista um grupo de utilizadores, infra-estrutura disponibilizada, ideias pré-definidas, resposta a uma ou mais situações específicas, etc.

Nos detalhes gerais do Template obtemos informações como o nome do Template, a data de criação e modificação, versão, custo geral, se é ou não customizável e o numero de ServerGroups associados, assim como discos e serviços que já foram criados tendo por base este template.

Todos os detalhes do Template ficam disponíveis ao utilizador: 1) Grupo de servidores com informações como nome, quantidade mínima de servidores e que será a base do serviço criado e quantidade máxima de servidores, sinal que o serviço poderá crescer em termos de quantidade de servidores. Tipo de cluster e o número de servidores lógicos existentes no grupo; 2) Servidores, onde o utilizador obtém informações como nome, hostname, tipo de virtualização, memória usada e quantidade máxima de memória, processadores usados e quantidade máxima de processadores, se é ou não virtual e se está ou não em cloud; 3) Custos, em detalhes, ficam disponíveis ao utilizador. Informações sobre custo total, custo de base por cada servidor, custo por processador e memória; 3) Os discos também apresentam informação como nome, tipo de storage, tamanho, raid level, se é ou não bootable, shareable e custo do disco; 4) Software pré-instalado ou instalado posteriormente pelo utilizador. Informações como nome, tipo, OS e virtualização estão disponíveis; 5) Interfaces com informações como nome da interface, se é ou não primária,

redundância, subnet, publica, shareable, tipo de IP, endereço, DHCP activo ou não e custo final.

Segue-se as acções possíveis de execução pelo utilizador aquando da visualização de toda a informação relativa a Templates.

Figura 32 - Execução tab "Template"

✓ *Export All Templates*

Acção disponível apenas ao administrador, permite o download em ficheiro XML dos detalhes de todos os Templates para que o administrador possa consultar essa informação em modo offline se assim o entender. Detalhes que ficam disponíveis para visualização no clique do ícon Template no menu lateral esquerdo.

O ficheiro XML pode depois ser visualizado mesmo no browser ou qualquer programa que suporte essa visualização como é o caso do Notepad++.

✓ *Export Template*

Acção disponível apenas ao administrador, permite o download em ficheiro XML dos detalhes do Template em visualização, igualmente como a acção "Export All Templates", o administrador poderá consultar mais tarde em modo offline se assim o entender. Detalhes que ficam disponíveis para visualização no clique do icon do nome do Template respectivo no menu lateral esquerdo.

✓ *Import Template*

Acção disponível apenas ao administrador, permite que seja possível adicionar mais Templates ao Catálogo de Serviços, ficando assim disponível ao utilizador mais opções com novas configurações. A leitura é feita directamente de um ficheiro XML, não é admissível outro tipo de formato.

✓ *Delete Template*

Acção disponível apenas ao administrador, permite que seja possível remover um Templates do Catálogo de Serviços. Como todos os Serviços tem por base um Template, a remoção de um Template implica a remoção de todos os

Serviços que tem como base esse Template. Tal situação é salvaguardada obrigando o administrador a clicar em OK numa InfoWindow de aviso.

- ✓ *XML View*

Acção disponível apenas ao administrador, permite que seja possível a visualização de todos os detalhes do Templates directamente numa InfoWindow dentro do webservice. A informação em XML torna-se vantajosa ao administrador para a visualização de detalhes mais pormenorizados e organização de toda a infra-estrutura.

- ✓ *Export All Services*

Acção que à semelhança do "Export All Templates", permite o download em ficheiro XML dos detalhes de todos os Serviços associados ao Template em visualização e a um determinado utilizador para que possa consultar em modo offline se assim o entender. Detalhes que ficam disponíveis para visualização no clique do ícon Service no menu lateral esquerdo.

O ficheiro XML pode depois ser visualizado no browser ou qualquer programa que suporte essa visualização como é o caso do Notepad++.

- ✓ *Create Service*

Finalmente a acção/objectivo de todo o webservice, a criação e disponibilização de Serviços. Como já foi referido, um Serviço tem como base um Template por isso na visualização dos detalhes do Template, esta opção está disponível. Aquando da criação de um Serviço, este terá como base o Template em visualização.

Existem campos de preenchimento obrigatórios como nome do serviço, hostname, tempo atribuído ao serviço (1 mês, por defeito) e email do utilizador. As opções "Notes" e "Billing code" são opcionais.

- ✓ *All Requests*

Todas os pedidos realizados no webservice por um determinado utilizador ficam disponíveis para visualização do seu estado e inclusive desencadear acções nesses mesmos pedidos como cancelamento (percentagem inferior a 100%) mediante decisão do utilizador.

A salientar que existem pedidos que carecem da aprovação do administrador por isso o pedido poderá ficar parado numa determinada percentagem (%) a espera dessa aprovação. Se o pedido não necessitar de aprovação, continuará a sua execução até estar completo (100%) e automaticamente será adicionado ao menu lateral esquerdo do utilizador.

- ✓ *Template Requests*

Todas os pedidos associados a um determinado Template por um determinado utilizador ficam igualmente disponíveis para visualização do seu estado.

Tabela 2 - Funções associadas ao Template

Função	Descrição
listTemplates	Função que obtem uma listagem da informação relativa aos Templates do Catálogo de Serviços.
importTemplate	Função que permite o Upload da informação relativa a um Template em formato XML. Operação executada apenas pelo Administrador ou Arquitecto de toda a infra-estrutura de orquestração
exportTemplate	Função que obtém toda a informação relativa a um Template e permite o download dessa informação em formato XML. Operação executada apenas pelo Administrador ou Arquitecto de toda a infra-estrutura de orquestração
exportTemplateList	Função de complemento a toda a informação dos Templates do Catálogo de Serviço. Uso exclusivo do webservice.
deleteTemplate	Função que permite a remoção de um Template do Catálogo de Serviços. Operação executada apenas pelo Administrador ou Arquitecto de toda a infra-estrutura de orquestração.
listServicesByTemplateName	Função que obtém todos os Serviços associados a um determinado Template. Para visualização de informação e opção de download.

5.5.2.3 Serviço

O Serviço tem sempre como base as definições de um Template e nesse sentido, tudo que o utilizador poderá fazer nesse Serviço são acções que estão disponíveis nesses Template como limites mínimos e máximos de servidores, processadores, memória, etc. Outras opções como discos e interfaces ficam disponíveis ao utilizador que poderá adicionar se assim o entender, com os custos inerentes a este upgrade de Serviço.

Um Serviço é composto por dados base relativos a informações gerais desse Serviço, listas de acções associadas e uma lista de grupos de servidores, sendo que cada grupo poderá ser diferente entre si com recursos, funcionalidades e softwares completamente díspares.

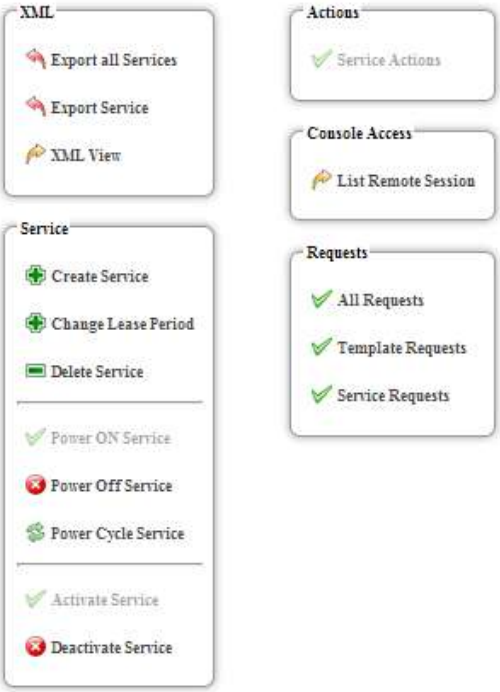
Assim, em termos de informação disponível associada ao serviço, o utilizador poderá obter o nome do serviço, id (apenas acessível ao administrador), notas, status,

o Template base que lhe deu origem, data de criação e ultima data de modificação, o custo geral do serviço, custo fixo, o período de uso do serviço (por defeito 1 mês), nome do utilizador que criou o serviço, organização e tamanho das listas associadas, nomeadamente acções e grupo de servidores.

Os Serviços, assim como os Templates ficam disponíveis em listagem no menu lateral esquerdo num formato em árvore, intuitivo e rápido de selecionar. Cada Serviço fica na cadeia hierárquica do Template que lhe deu origem. Associado a cada Serviço, também poderemos visualizar nesses mesmo menu todos os recursos associados a esse Serviço como listagem dos grupos de servidores, em cada grupo os seu servidores e em cada servidor, os recursos como discos, software e interfaces.

Segue-se as acções possíveis de execução pelo utilizador aquando da visualização de toda a informação relativa ao Serviço.

Figura 33 - Execução tab "Service"



✓ *Export All Services*

Acção que permite o download em ficheiro XML dos detalhes de todos os Serviços do utilizador, permitindo a posterior consulta em modo offline. Detalhes que ficam disponíveis para visualização no clique do ícons Service no menu lateral esquerdo.

✓ *Export Service*

Permite o download em ficheiro XML dos detalhes do Serviço em visualização, igualmente como a acção "Export All Services", o utilizador poderá consultar mais tarde em modo offline os detalhes do serviço. Detalhes que ficam

disponíveis para visualização no clique do icon do nome do Serviço respectivo no menu lateral esquerdo.

- ✓ *XML View*

Permite que seja possível a visualização de todos os detalhes do Serviço directamente numa InfoWindow dentro do webservice. A informação em XML torna-se vantajosa ao utilizador para a visualização de detalhes mais pormenorizados e organização da infra-estrutura.

- ✓ *Create Service*

Mesma acção existente nos Templates. Directamente da visualização de um Serviço, é possível criar um outro Serviço, tendo igualmente por base o mesmo Template. O Serviço criado será igual ao Serviço base em visualização.

- ✓ *Delete Service*

Permite a remoção do Serviço em visualização. Após esta acção, deixam de ser cobrado qualquer custo associado ao Serviço que foi removido.

- ✓ *Power ON Service*

Acção que permite que todos os Servidores de todos os ServerGroups existentes no Serviço sejam ligados e fiquem disponíveis para utilização. Esta acção é dependente do status do Serviço, fica activa quando "serviceStatus=='OFF'"

- ✓ *Power OFF Service*

Acção que permite que todos os Servidores de todos os ServerGroups existentes no Serviço sejam desligados, deixando de estar disponíveis para utilização. Esta acção é dependente do status do Serviço, fica activa quando "serviceStatus==UP"

- ✓ *Power Cycle Service*

Acção que permite que todos os Servidores de todos os ServerGroups existentes no Serviço façam um reboot e continuem disponíveis para utilização. Esta acção é dependente do status do Serviço, fica activa quando "serviceStatus==UP"

- ✓ *Activate Service*

Permite que todos os Servidores de todos os ServerGroups existentes no Serviço sejam activados. No final desta operação, o Serviço fica em modo "serviceStatus==UP" e todos os servidores ficam automaticamente disponíveis para utilização.

✓ *Deactivate Service*

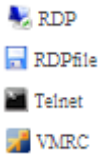
Permite que todos os Servidores de todos os ServerGroups existentes no Serviço sejam desactivados. No final desta operação, o Serviço fica em modo "serviceStatus==DEACTIVATED".

✓ *Service Actions*

Os "triggered actions" associadas a cada Template são parte integrante do Serviço criado. Também esta informação está disponível em forma de listagem. Estas acções automaticamente desencadeadas pelo webservice poderão pertencer ao Serviço propriamente dito ou então existirem apenas num serverGroup ou até mesmo apenas num Server.

✓ *List Remote Session*

No final da criação de um Serviço, o objectivo é a ligação a/s VM/s associadas a esse mesmo Serviço. Nesse intuito, directamente da tab Serviço, através desta acção, ficam disponíveis numa lista todas as VM/s com os acessos remotos permissíveis:



RDP (Remote Desktop Protocol)
Dowload do ficheiro RDP para posterior ligação
Telnet
VMRC (VMware Remote Console)

Todos os pedidos associados a um determinado Serviço por um determinado utilizador ficam igualmente disponíveis para visualização do seu estado. Quando a listagem de pedidos tem grandes dimensões ou se a visualização do pedido diz respeito a este Serviço, fazer apenas a listagem dos pedidos pelo Serviço em causa torna-se mais simples, útil e directa.

Tabela 3 - Funções associadas ao Serviço

Função	Descrição
listServices	Detalhes dos Serviços associados a um determinado Utilizador. Para visualização de informação e opção de download. O Administrador obtem todos os Serviços existentes no webservice.
getService	Informação associada a um determinado Serviço. Para visualização de informação e opção de download.
createService	Criação de um Serviço
deleteService	Remoção de um Serviço associado a um Template
deactivateService	Desactivação de um Serviço

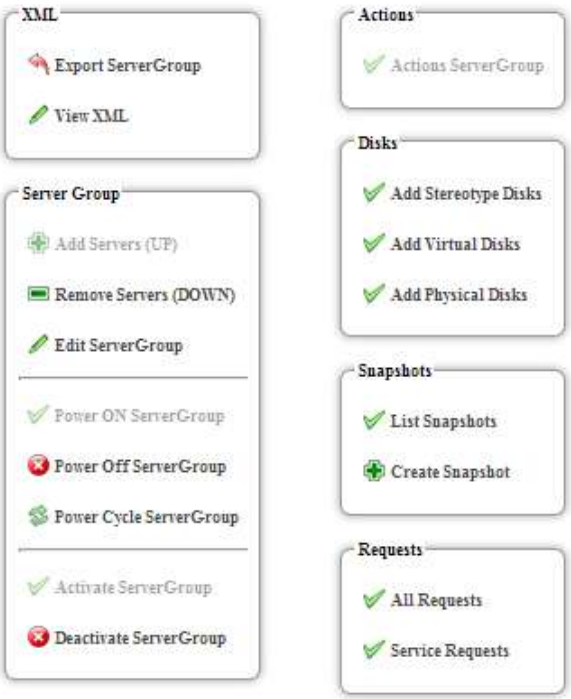
activateService	Activação de um Serviço
changeServiceLease	Edição do "Lease period" asociado a um Serviço
powerOnService	Ligar um Serviço
powerOffService	Desligar um Serviço
powerCycleService	Ligar um Serviço após reboot
getRemoteSessionsList	Listagem de todos os servidores e seus IPs associados para posterior listagem das sessões remotas.

5.5.2.4 Grupo de Servidores

Cada Serviço tem os seus recursos associados, nomeadamente uma lista de Grupo de Servidores onde o utilizador poderá obter informações como o nome do Serviço, nome do Grupo de Servidores, o Status, número mínimo e máximo de Servidores, uma progressbar com indicação do número de servidores disponíveis no momento, se tem uma sequencia Boot, datas de criação e ultima modificação e finalmente, uma lista de Servidores.

Segue-se as acções possíveis de execução pelo utilizador aquando da visualização de toda a informação relativa ao Grupo de Servidores.

Figura 34 - Execução tab "ServerGroups"



- ✓ *Export Server Group*

Permite o download em ficheiro XML dos detalhes do Grupo de Servidores em visualização, o utilizador poderá consultar esses detalhes mais tarde em modo offline. Detalhes que ficam disponíveis para visualização no clique do icon do nome do Grupo de Servidores respectivo no menu lateral esquerdo.

- ✓ *XML View*

Permite que seja possível a visualização de todos os detalhes do Grupo de Servidores directamente numa InfoWindow dentro do webservice. A informação em XML torna-se vantajosa ao utilizador para a visualização de detalhes mais pormenorizados e organização da infra-estrutura.

- ✓ *Add Servers (UP)*

Acções que permite a edição do Grupo de Servidores, aumentando o número de Servidores dentro desse Grupo de Servidores. Esta opção só fica visível e acessível ao utilizador se o limite máximo do número de Servidores for superior ao actualmente existente em utilização. Limite definido nas configurações do Serviço que vieram por arrasto das configurações do Template aquando da criação desse Serviço. Estas alterações implicam o aumento do custo total do Serviço.

- ✓ *Remove Servers (DOWN)*

Acções que permite a edição do Grupo de Servidores, diminuindo o número de Servidores dentro desse Grupo de Servidores, é disponibilizada a lista de Servidores existentes e o utilizador seleciona um ou mais para remoção. Esta opção só fica visível e acessível ao utilizador se o limite mínimo do número de Servidores for inferior ao actualmente existente em utilização. Limite definido nas configurações do Serviço que vieram por arrasto das configurações do Template aquando da criação desse Serviço. Estas alterações implicam a diminuição do custo total do Serviço.

- ✓ *Edit ServerGroup*

Acções que permite a edição do Grupo de Servidores relativamente a memória e processadores. A edição vai ser efectuada em todos os Servidores do Grupo de Servidores. Como nos limites máximos e mínimos de Servidores, também os limites de memória e processadores vieram por arrasto das configurações do Template aquando da criação do Serviço. Estas alterações implicam a diminuição ou aumento do custo total do Serviço.

- ✓ *Power ON ServerGroup*

Acção que permite que todos os Servidores do ServerGroups em visualização sejam ligados e fiquem disponíveis para utilização. Esta acção é dependente do status do Grupo de Servidores, fica activa quando "serverGroupStatus=='OFF'"

- ✓ *Power OFF ServerGroup*

Acção que permite que todos os Servidores do ServerGroups em visualização sejam desligados, deixando de estar disponíveis para utilização. Esta acção é dependente do status do Grupo de Servidores, fica activa quando "serverGroupStatus==UP"

- ✓ *Power Cycle ServerGroup*

Acção que permite que todos os Servidores do ServerGroups em visualização façam um reboot e continuem disponíveis para utilização. Esta acção é dependente do status do Grupo de Servidores, fica activa quando "serverGroupStatus==UP"

- ✓ *Activate ServerGroup*

Permite que todos os Servidores do ServerGroups em visualização sejam activados. No final desta operação, o Grupo de Servidores fica em modo "serverGroupStatus==UP" e todos os servidores ficam automaticamente disponíveis para utilização.

- ✓ *Deactivate ServerGroup*

Permite que todos os Servidores do ServerGroups em visualização sejam desactivados. No final desta operação, o Grupo de Servidores fica em modo "serverGroupStatus==DEACTIVATED".

- ✓ *Actions ServerGroup*

Como no Serviço, os "triggered actions" associadas a cada Template são parte integrante do Serviço criado, inclusive no Grupo de Servidores. Se o Grupo de Servidores em visualização tiver "triggered actions", esta acções estará visível ao utilizador para listagem.

- ✓ *List Snapshots*

Acção que permite listar todos os Snapshots existentes no Grupo de Servidores.

Nessa listagem, é possível executar acções como "Revert Snapshot" que reverte um determinado Snapshot para o servidor específico ou "Delete Snapshot" que permite a remoção de um Snapshot do sistema.

- ✓ *Create Snapshot*

Acção que permite a criação de Snapshots no Grupo de Servidores.

Existem campos obrigatórios como o nome do Snapshot e a selecção de um ou mais servidores de uma lista de servidores pertencentes ao Grupo de Servidores. Opções como "Include Memory State" ou "Quiesce filesystem", estão também disponíveis para selecção aquando da criação do Snapshot.

- ✓ *Add Stereotype Disks*

Acção que permite adicionar discos a um Grupo de Servidores. O disco adicionado fica acessível a todos os Servidores do Grupo de Servidores. São considerados "Stereotype Disks" todos os discos existentes no grupo excepto discos de boot.

Qualquer acção de adição de discos a um Grupo de Servidores só é possível se não existirem Snapshots relativos a qualquer Servidor desse Grupo de Servidores.

Esta acção lista todos os "Stereotype Disks" já existentes. O utilizador seleciona o que deseja clonar (disco base) e é adicionado um novo disco ao Grupo de Servidores. Esta acção carece da aprovação do administrador.

- ✓ *Add Virtual Disks*

Acção que permite adicionar discos virtuais a um Grupo de Servidores. O disco adicionado fica acessível a todos os Servidores do Grupo de Servidores.

Existem campos de preenchimento obrigatório como o nome do disco e tamanho. O webservice fornece um nome único e 10Gb de tamanho por defeito, o utilizador pode alterar estes campos. Esta acção carece da aprovação do administrador. Além de aprovar ou desaprovar este pedido, pode inclusive definir um novo custo.

- ✓ *Add Physical Disks*

Acção que permite adicionar discos físicos a um Grupo de Servidores. O disco adicionado fica acessível a todos os Servidores do Grupo de Servidores.

Igualmente como a adição de discos virtuais, existem campos de preenchimento obrigatório como o nome do disco, tamanho e nesta opção o tipo de RAID. Acção que carece igualmente da aprovação do administrador.

Tabela 4 - Funções associadas ao Grupo de Servidores

Função	Descrição
getLogicalServerGroup	Obtém os detalhes associados a um Grupo de Servidores Lógico em XML para visualização na infowindow e opção de download.
addServersToLogicalServerGroup	Adiciona um Servidor a um Grupo de Servidores. Opção permissvel apenas se o número de servidores activos ("activeServerCount") for menor que o número máximo de servidores ("maxServerCount").

removeServersFromLogicalServerGroup	Remove um Servidor de um Grupo de Servidores Lógicos. Opção permissvel apenas se o número mínimo de servidores (variável “minServerCount”) for menor que o número de servidores activos (variável “activeServerCount”).
editLogicalServersServersGroup	Permite a edição de todos os Servidores pertencentes a um Grupo de Servidores Lógicos
powerOnLogicalServersGroup	Liga todos os Servidores que pertencem a um mesmo Grupo de Servidores Lógicos
powerOffLogicalServersGroup	Desliga todos os Servidores que pertencem a um mesmo Grupo de Servidores Lógicos
powerCycleLogicalServersGroup	Faz o reboot de todos os Servidores que pertencem a um mesmo Grupo de Servidores Lógicos
activateLogicalServerGroup	Activa todos os Servidores que pertencem a um mesmo Grupo de Servidores Lógicos
deactivateLogicalServerGroup	Desactiva todos os Servidores que pertencem a um mesmo Grupo de Servidores Lógicos

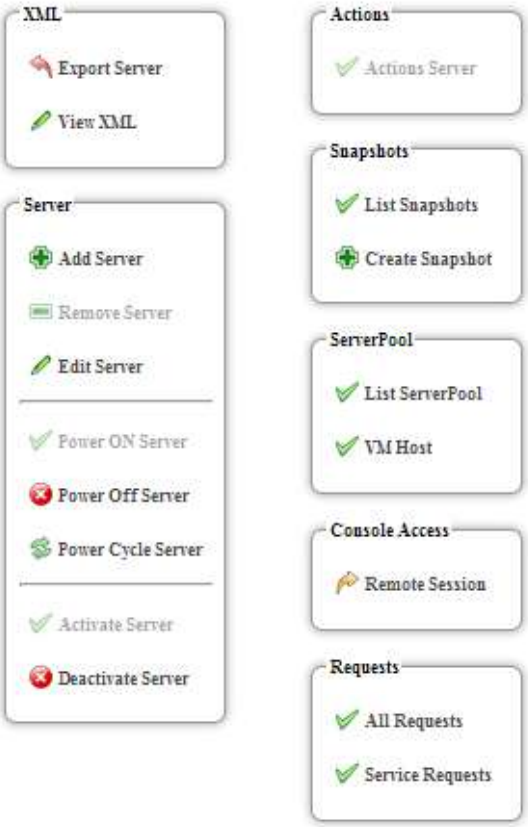
5.5.2.5 Servidores

Um Grupo de Servidores é constituído por uma lista de Servidores. Cada grupo, como já foi referido, pode ser completamente díspar dos demais dentro de um mesmo Serviço. O mesmo já não acontece nos Servidores dentro de um mesmo Grupo de Servidores. As configurações dos Servidores dentro de um mesmo Grupo de Servidores tem todos as mesmas características, podendo variar apenas no numero de discos, memória e processamento.

O utilizador poderá obter informações como nome do Servidor, status, se é ou não virtual, se está em cloud, se é o servidor base, memória e processamento para futuros upgrades com progressbars para facilmente visualizar o estado actual, custo, datas de criação e modificação, localização do ficheiro da VM (acessível apenas ao administrador) e as listas associadas de discos, software e interfaces.

Segue-se as acções possíveis de execução pelo utilizador aquando da visualização de toda a informação relativa aos Servidores.

Figura 35 - Execução tab "Server"



✓ *Export Server*

Permite o download em ficheiro XML dos detalhes do Servidor em visualização, o utilizador poderá consultar mais tarde em modo offline os detalhes desse Servidor. Detalhes que ficam disponíveis para visualização no clique do icon do nome do Servidor respectivo no menu lateral esquerdo.

✓ *XML View*

Permite que seja possível a visualização de todos os detalhes do Servidor directamente numa InfoWindow dentro do webservice. A informação em XML torna-se vantajosa ao utilizador para a visualização de detalhes mais pormenorizados e organização da infra-estrutura.

✓ *Remove Server*

A diferença desta acção no Servidor para a mesma acção no Grupo de Servidores é o facto da remoção ser do Servidor em visualização. Para remover mais do que um Servidor, essa operação terá obrigatoriamente que ser no Grupo de Servidores.

- ✓ *Edit Server*

Como na remoção de servidores, a diferença desta acção para a mesma acção no Grupo de Servidores é que a edição de Servidores na tab Servidor apenas afectará o Servidor em visualização e não os restantes Servidores do Grupo de Servidores.

- ✓ *Power ON Server*

Acção que permite a ligação do Servidor em visualização e que este fique disponível para utilização. Esta acção é dependente do status do Servidor, fica activa quando "serverStatus=='OFF'"

- ✓ *Power OFF Server*

Acção que permite que um Servidor seja desligado e que este fique indisponível para utilização. Esta acção é dependente do status do Servidor, fica activa quando "serverStatus==UP"

- ✓ *Power Cycle Server*

Acção que permite fazer o reboot a um Servidor e este continue disponível para utilização. Esta acção é dependente do status do Servidor, fica activa quando "serverStatus==UP"

- ✓ *Activate Server*

Permite a activação do Servidor em visualização. No final desta operação, o Servidor fica em modo "serverStatus==UP" e ficam automaticamente disponível para utilização.

- ✓ *Deactivate Server*

Permite a desactivação do Servidor em visualização. No final desta operação, o Servidor fica em modo "serverStatus==DEACTIVATED".

- ✓ *Actions Server*

Como no Grupo de Servidores, os "triggered actions" associadas a cada Servidor são parte integrante do Serviço criado. Se o Servidor em visualização tiver "triggered actions", esta acção estará visível ao utilizador para listagem.

- ✓ *List Snapshots*

Acção que permite listar todos os Snapshots associados ao Servidor em visualização. Todas as acções nos Snapshots são relativas ao Servidor em questão.

- ✓ *Create Snapshot*

Acção que permite a criação de Snapshots no Servidor.

Existem campos obrigatórios como o nome do Snapshot. O nome do Servidor é preenchido automaticamente com o nome do Servidor que está a ser visualizado. Opções como "Include Memory State" ou "Quiesce filesystem", estão também disponíveis para selecção aquando da criação do Snapshot.

- ✓ *List ServerPool*

Acção que permite a listagem de todas as pools de Servidores existentes no webservice. Na imagem seguinte podemos visualizar quatro pools de Servidores que foram criadas pelo administrador.

"Create Server Pool" é uma acção atribuídas apenas ao administrador com campos obrigatórios como nome da Pool.

"Delete Server Pool" é outra acção atribuídas apenas ao administrador, permitindo a remoção de uma Pool de Servidores do sistema.

Esta operação não remove Servidores nem Serviços mas remover todas as associações já existentes entre Servidores e utilizadores. Como acção importante, o webservice evidencia isso mesmo com uma InfoWindow de aviso ao administrador.

- ✓ *VM Host*

Acção que permite a listagem de todos os hosts físicos associados a uma determinada Pool de Servidores.

Nesta operação obtêm-se detalhes relativos ao host como é o caso do nome, id, endereço IP, domínio DNS, tipo de virtualização, modelo, cluster, memória total e consumida, dados dos processadores e ainda dados relativos ao armazenamento (DataStores) e rede (VirtualSwitch).

- ✓ *Remote Session*

Acção que permite a listagem de todas as Sessões Remotas atribuídas ao Servidor em visualização.

Esta acção é dependente do status do Servidor, fica activa quando "serverStatus=='UP'"

A imagem seguinte evidencia as duas ligações existentes para a VM do Servidor específico. Duas ligações relativas às duas Interfaces disponíveis no Servidor, uma na rede "HP Network" com o endereçamento IP: 16.23.187.5 e outra na rede local da infra-estrutura, rede "CloudSystem" com o endereçamento IP: 192.168.10.203

As opções de Sessões Remotas possíveis são as que já foram descritas na listagem das "Remote Sessions" do Serviço. Neste caso, direccionadas apenas ao um único Servidor.

Tabela 5 - Funções associadas aos Servidores

Função	Descrição
getLogicalServer	Obtém os detalhes de um Servidor Logico em formato XML para visualização na infowindow e opção de download

findLogicalServer	Obtém os Detalhes de um Servidor Lógico específico
addServer	Addiciona um Servidor a um Grupo de Servidores Lógicos se o número de servidores activos (variável "activeServerCount") for menor que o número máximo de servidores (variável "maxServerCount").
removeServer	Remove um Servidor de um Grupo de Servidores Lógicos se o número mínimo de servidores (variável "minServerCount") for menor que o número de servidores activos (variável "activeServerCount").
editLogicalServer	Permite a edição de memória e processamento de um determinado Servidor Lógico
powerOnLogicalServer	Liga um Servidor Lógico específico
powerOffLogicalServer	Desliga um Servidor Lógico específico
powerCycleLogicalServer	Reboot de um Servidor Lógico específico
activateLogicalServer	Activação de um Servidor Lógico específico
deactivateLogicalServer	Desactivação de um Servidor Lógico específico
getRdpFile	Obtém o ficheiro RDP para download e posterior acesso remoto
getLogicalServersAndIPs	Obtém uma lista de todos os IPs das Interfaces de um determinado Servidor Lógico
getVmHost	Obtém os detalhes do Host associado a um Servidor Lógico

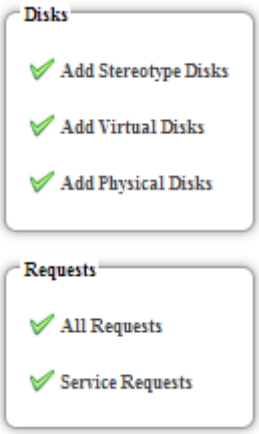
5.5.2.6 Disco

Cada servidor tem necessariamente um ou mais discos associados; estes podem ser rotulados de "Stereotype", "Virtual" ou "Physical". Um disco "Stereotype" corresponde a um disco-base, i.e., um disco que existe no template (do servidor) excepto disco de boot; os restantes, como o próprio nome indica, são discos virtuais ou físicos.

Nesta tab, o utilizador pode obter informação do nome do disco, tamanho, tipo de RAID, tipo de Storage, se é ou não Bootable, Shareable, tipo de Raw Mapping (VIRTUAL, PHYSICAL, NONE e OTHER), custo associado, ficheiro do disco (virtual - acessível apenas ao administrador) e data de criação e modificação.

Segue-se as acções possíveis de execução pelo utilizador aquando da visualização de toda a informação relativa aos discos.

Figura 36 - Execução tab “Disks”



✓ *Add Stereotype Disks*

Acção já enunciada no Grupo de Servidores. Como estamos a visualizar um disco e as suas configurações, esta acção estará apenas disponível se o disco em visualização não for bootable. Como já referido, apenas os discos que não sejam bootable são considerados “Stereotype Disks” e poderão ser criados novos discos com base no disco em questão.

Como no Grupo de Servidores, também nos discos, esta opção só será possível se não existirem Snapshots relativos a qualquer Servidor desse Grupo de Servidores. Esta acção carece da aprovação do administrador.

Tabela 6 - Funções associadas aos Discos

Função	Descrição
addDiskToLogicalServerGroup	Adiciona um "Stereotype Disk" a um Grupo de Servidores Lógicos. Operação que carece da aprovação do Administrador.
addNewDiskToLogicalServerGroup	Adiciona um disco virtual ou físico a um Grupo de Servidores Lógicos. Operação que carece da aprovação do Administrador. O tamanho do disco é automaticamente determinado pelo tipo do Grupo de Servidor Lógico.

5.5.2.7 Software

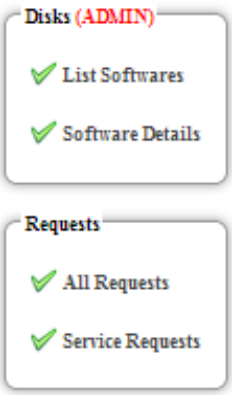
Recursos como Software também fazem parte dos recursos do Servidor. Mediante a instalação prévia do "Server Automation" no MOE, aquando da criação do Template, poderão ser associados nos recursos de cada Servidor uma lista de software como lista de packages com SO, software antivírus, base de dados, etc, previamente instalados em VM e posteriormente transformados em Tamplate.

Cada Servidor pertencente a um mesmo Grupo de Servidores terá uma mesma lista de software base associado.

Informações relativas ao software também estão disponível ao utilizador como nome, tipo, tipo de OS, tipo de Deployment Service, ficheiro de configuração do OS (acessível apenas ao administrador), tipo de virtualização, localização do ficheiro relativo ao software (acessível apenas ao administrador) e data de criação e modificação.

Segue-se as acções possíveis de execução pelo utilizador aquando da visualização de toda a informação relativa ao Software.

Figura 37 - Execução tab "Software"



- ✓ *List Softwares*

Acção disponível apenas ao administrador, permite a listagem de todo o Software disponível no sistema. O software pertencente ao Grupo de Servidores/Servidor está disponível directamente no menu lateral esquerdo.

Mediante a instalação prévia do "Server Automation" no MOE, aquando da criação do Template, poderá ser associado nos recursos de cada Grupo de Servidores/Servidor uma lista de software como lista de packages com SO, software antivírus, base de dados, etc.

- ✓ *Software Detals*

Acção disponível apenas ao administrador, permite a visualização de todos os detalhes do Software.

Tabela 7 - Funções associadas ao Software

Função	Descrição
listSoftware	Obtém a lista de todo o Software disponível no webservice. Operação executada apenas pelo Administrador ou Arquitecto de toda a infra-estrutura de orquestração
getSoftware	Obtém os detalhes de um Software específico. Operação executada apenas pelo Administrador ou Arquitecto de toda a infra-estrutura de orquestração

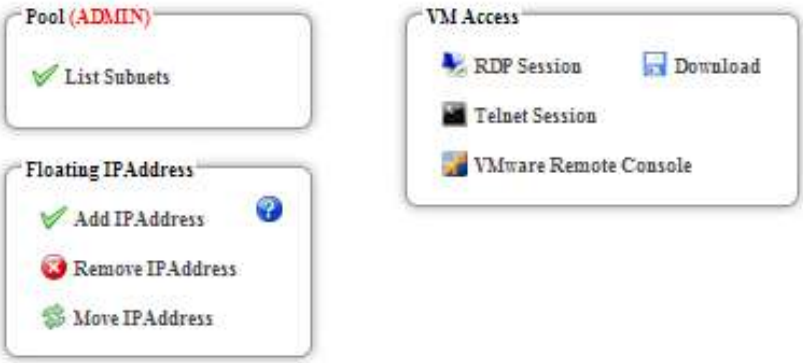
5.5.2.8 Interface

As Interfaces (NICs) fazem igualmente parte das configurações base de cada Servidor podendo, no entanto, existir redundância ou não.

Informações das Interfaces disponível na tab respectiva como nome, se é a interface primária, data de criação, modificação e informações sobre a rede como nome, se é pública, partilhada, tipo e endereço IP com informações como o endereço IP propriamente dito, tipo de atribuição, custo e uma lista de NatEntrys (Network Address Translation).

Segue-se as acções possíveis de execução pelo utilizador aquando da visualização de toda a informação relativa as Interfaces.

Figura 38 - Execução tab "Interface"



- ✓ *List Subnets*

Acção disponível apenas ao administrador, permite a listagem de todas as redes existentes no sistema com informações como nome da rede, endereço, mascara, source, tipo de IP, se é uma rede pública e os endereçamentos DHCP e estácticos, número de IPs usados e em uso, etc.

✓ *Add IPAddress*

Acção que permite atribuir automaticamente (pelo sistema) um IP flutuante a uma Interface. Uma Interface associada a um Servidor dentro de um determinado Serviço.

Com esta acção, o utilizador solicita um IP flutuante de uma pool de IPs que o administrador tenha criado. Nesse sentido, o utilizador apenas solicita um IP flutuante para uma Interface (NIC0) de um determinado Servidor pertencente a um Serviço. Como o utilizador está a visualizar a Interface, os campos são automaticamente preenchidos. Operação apenas permitida em servidores lógicos em cloud.

✓ *Remove IPAddress*

Acção que permite remover um IP flutuante. Neste caso, o utilizador tem que fornecer o endereço IP flutuante previamente adquirido.

✓ *Move IPAddress*

Acção que permite mover um determinado IP flutuante entre instâncias. Na eventualidade do utilizador já for detentor de um IP flutuante, pode move-lo atribuindo-o à Interface em visualização. Operação apenas permitida em servidores lógicos em cloud.

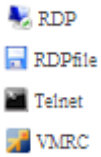
✓ *VM Access*

Acções de acesso remoto directamente na Interface.

Esta acções são dependente do status do Servidor, ficam activas quando "serverStatus=='UP'"

Como estamos directamente a visualizar uma Interface, o acesso remoto é feito através dessa Interface, desse IP conhecido.

As opções de Sessões Remotas possíveis são as que já foram descritas na listagem das "Remote Sessions" do Serviço.



RDP (Remote Desktop Protocol)
Dowload do ficheiro RDP para posterior ligação
Telnet
VMRC (VMware Remote Console)

Tabela 8 - Funções associadas às Interfaces

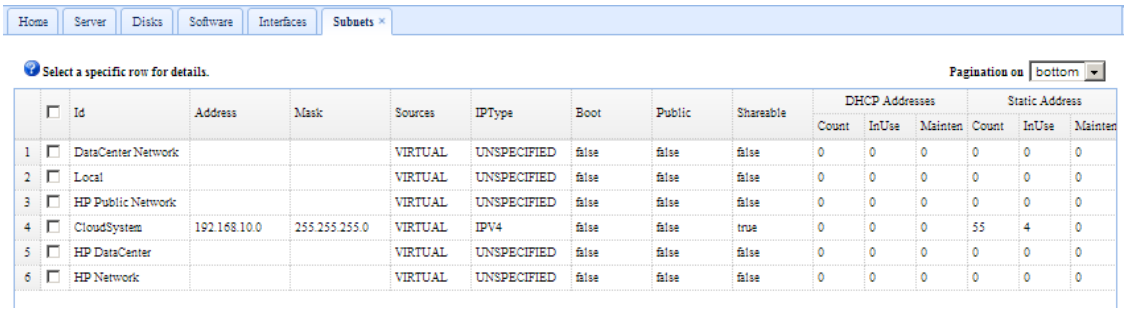
Função	Descrição
getVMID	Obtém o ID de uma VM atarvés do endereço IP da Interface

executeRDP	Execução directa do comando RDP "mstsc.exe /v:IPAddress /console"
addFloatingIPAddress	Adiciona automaticamente um endereço IP flutuante a uma determinada Interface. Operações de IP flutuante não são suportadas para Servidores Lógicos que não estejam em cloud
removeFloatingIPAddress	Remove um determinado endereço IP flutuante de um Servidor Lógico. Operações de IP flutuante não são suportadas para Servidores Lógicos que não estejam em cloud
moveFloatingIPAddress	Associa um determinado endereço IP flutuante já existente a uma Interface. Operações de IP flutuante não são suportadas para Servidores Lógicos que não estejam em cloud

5.5.2.9 Redes

As Redes, como o Software, terão que estar previamente configuradas e implementadas para posterior uso em Templates e Serviços. A listagem de redes existentes na infra-estrutura está apenas disponível ao Administrador ou Arquitecto de toda a infra-estrutura de orquestração e permite a visalização de todas as redes existentes com informações como o nome da rede, gama de endereços IP, mascara, tipo de IPV e se é um endereçamento estáctico ou obtido por DHCP.

Figura 39 - Listagem das redes existentes



	Id	Address	Mask	Sources	IPType	Boot	Public	Shareable	DHCP Addresses			Static Address		
									Count	InUse	Mainten	Count	InUse	Mainten
1	DataCenter Network			VIRTUAL	UNSPECIFIED	false	false	false	0	0	0	0	0	0
2	Local			VIRTUAL	UNSPECIFIED	false	false	false	0	0	0	0	0	0
3	HP Public Network			VIRTUAL	UNSPECIFIED	false	false	false	0	0	0	0	0	0
4	CloudSystem	192.168.10.0	255.255.255.0	VIRTUAL	IPV4	false	false	true	0	0	0	55	4	0
5	HP DataCenter			VIRTUAL	UNSPECIFIED	false	false	false	0	0	0	0	0	0
6	HP Network			VIRTUAL	UNSPECIFIED	false	false	false	0	0	0	0	0	0

Segue-se as acções possíveis de execução aquando da visualização de toda a informação relativa às Redes.

Tabela 9 - Funções associadas às Redes

Função	Descrição
listSubnets	Obtém uma lista com todas as redes existentes no webservice.
listSubnet	Obtém os detalhes de uma rede específica.

5.5.2.10 Snapshots

Os Snapshots são um recurso valioso quando nos deparamos com problemas de inconsistências, desastres, segurança, etc, que, não sendo backup, permite a recuperação do servidor em segundos, contrariamente a restauração a partir de um backup tradicional, recuperação ou reinstalação de um SO ou qualquer outra técnica com o seu exagerado tempo de consolidação.

A listagem de todos os Snapshots está disponível aquando da visualização da informação relativa ao Grupo de Servidores onde são visualizados todos os Snapshots pertencentes ao grupo e no Servidor onde apenas são visualizados os Snapshots desse Servidor em visualização.

Uma lista relacionada de Snapshots sucessivos com informações como nomes do Serviço, do Grupo de Servidores e do Servidor, assim como nome do Snapshot, a sua relação parental e se está ou não activo, ficam disponíveis ao utilizador.

Segue-se as acções possíveis de execução aquando da visualização de toda a informação relativa aos Snapshots.

Figura 40 - Execução tab "Snapshot"

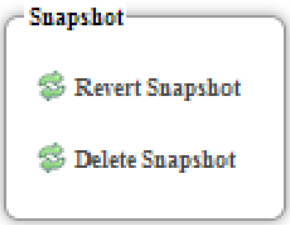


Tabela 10 - Funções associadas aos Snapshots

Função	Descrição
listServerSnapshots	Obtém a listagem de todos os Snapshots de um Servidor Lógico específico
listServerGroupSnapshots	Obtém a listagem de todos os Snapshots de um Grupo de Servidor Lógico

snapshotCreate	Cria um Snapshot de um Servidor específico
revertServerSnapshot	Reverte um determinado Snapshot para o Servidor correspondente
deleteServerSnapshot	Remove um Snapshot do sistema

5.5.2.11 Request

A visualização do estado actual das acções do utilizador é considerado um dos pontos chave desta *framework*. O simples update da "progressbar" carece de uma perfeita sincronização entre o Webservice e o MOE.

O utilizador poderá visualizar sempre uma listagem de todos os seus pedidos ou pormenoriza-los por relações: 1) Utilizador/Serviços; 2) Template/Serviços; 3) Serviço.

A listagem de todos os pedidos está disponível aquando da visualização da informação relativa a qualquer recurso, sendo que a opção 1) é geral e sempre disponível e as opções 2) e 3) são específicas a recursos em visualização. Ficam assim disponíveis informações como tipo, data submissão do pedido, começo e fim, status, progresso (%), nome do Serviço ao qual o pedido está associado, etc.

Segue-se as acções possíveis de execução aquando da visualização da listagem dos pedidos do utilizador.

Figura 41 - Execução tab "Requests"

Opções como "Aproval State" e "Continue" de pedidos que carecem da aprovação do Administrador, ficam apenas acessíveis aos utilizadores com permissões de administração e a opção de cancelamento acessível aos utilizadores que solicitaram esses pedidos.

Tabela 11 - Funções associadas aos Requests

Função	Descrição
listRequests	Listagem de todos os pedidos do utilizador
listRequestsTemplateDetails	Listagem de todos os pedidos do par utilizador/Template
listRequestsServiceDetails	Listagem de todos os pedidos do par utilizador/Serviço
getRequestInfo	Obtém detalhes de toda a informação associada a um determinado pedido
getPercentComplete	Obtém a percentagem de um pedido
getRequest	Obtém os detalhes particulares de um pedido
setRequestApprovalState	Aprovação de um pedido feito por um utilizador e que esteja no estado: 'PAUSED_FOR_APPROVAL'. Operação executada apenas pelo Administrador ou Arquitecto de toda a infra-estrutura de orquestração
continueRequest	Permite a continuação de um pedido feito por um utilizador e que esteja no estado: 'IN_PROGRESS'. Operação executada apenas pelo Administrador ou Arquitecto de toda a infra-estrutura de orquestração
cancelRequest	Cancelamento de um pedido

5.5.2.12 Pools de Servidores

Uma pool de servidores é um grupo de um ou mais hosts de virtualização com a uma mesma arquitetura, com acesso às mesmas redes virtuais e físicas e recursos de armazenamento. As Pools de servidores permitem balanceamento de carga, recursos de alta disponibilidade e partilha de alguns recursos para todos os membros dessa pool.

Informações das Pools disponível na tab respectiva como nome, ID da organização, Ids dos Hosts e o grupo de utilizadores associados a cada Pool de Servidores.

Segue-se as acções possíveis de execução aquando da visualização de toda a informação relativa a uma determinada Pool de Servidores.

Figura 42 - Execução tab “ServerPools”

Tabela 12 - Funções associadas às Pools de Servidores

Função	Descrição
listServerPools	Obtém uma listagem de todas as Pools de Servidores para a criação de um Serviço.
listServerPoolsTab	Obtém todas as Pools de Servidores existentes no webservice. Informação disponibilizada na tab serverPools
listVmHostIds	Obtém uma lista com todos os Hosts associados a uma Pool de Servidores
getServerPool	Obtém os detalhes de uma Pool de Servidores
createServerPool	Permite a criação de uma nova Pool de Servidores. Operação executada apenas pelo Administrador ou Arquitecto de toda a infra-estrutura de orquestração
deleteServerPool	Remove uma Pool de Servidores do systema. Operação executada apenas pelo Administrador ou Arquitecto de toda a infra-estrutura de orquestração

5.6 Camada de Dados

Numa orientação a objectos, os objectos considerados “persistentes” são todos os objectos que continuam a existir após o término de execução do programa, por norma, guardados numa base de dados relacional ou mesmo em ficheiros em disco. Já os objectos considerados “não persistentes” deixam de existir quando o programa termina, são objectos armazenados em tempo de execução e são usados apenas durante esse período.

5.6.1 Dados não persistentes

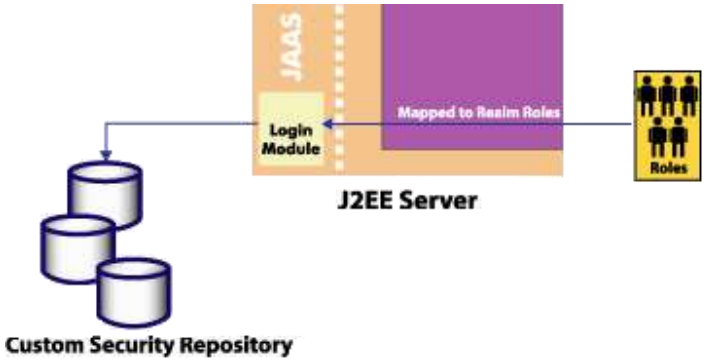
Os dados não persistentes, nesta *framework*, destinam-se exclusivamente ao utilizador, permissões de acesso e segurança. Internamente, o servidor guarda, em tempo de execução, dados relativos ao utilizador como é o caso do username

(*userPrincipal*), password (*IOClientPasswordHandler*), privilégios de acesso (*rolePrincipal*), grupo de utilizadores (*userGroups*) e credenciações necessária (*TrustAllX509TrustManager*) para comunicação com a API do MOE. A comunicação é estabelecida (*IOSoapServicePort_Client*) e quando existir a solicitação de determinados conteúdos, estes são disponibilizados ou não, mediante o "tipo" (*role*) de utilizador e suas permissões.

Neste contexto, a autenticação é todo um processo de determinar se uma entidade é quem quem diz ser e a autorização o processo de conceder permissões de acesso a recursos. Logicamente, a autenticação precede a autorização.

O módulo login implementa isso mesmo, todos os passos de autenticação, credenciação e permissões, uma técnica conhecida como Realm JAAS (Java Authentication and Authorization Service) como conjunto de interfaces de programação de aplicações (APIs) com opções de autenticações em modo Basic e Form, usando para isso os denominados realms: File e JDBC (Java Database Connectivity) em runtime.

Figura 43 - Realm JAAS (Java Authentication and Authorization Service)



5.6.2 Dados persistentes

5.6.2.1 Cookies HTTP

Os dados persistentes destinam-se especialmente ao utilizador e foram idealizados nesse sentido, tornar a relação utilizador/webservice mais amigável e acima de tudo prática e independente do "status" da infra-estrutura.

Os tão conhecidos COOKIEs HTTP são dados armazenado no navegador/browser do utilizador quando este acede a um determinado site. É um mecanismo usado para guardar as informações relativas a um utilizador como username, formatação de conteúdos, actividades, escolhas, cliques, etc, para quando aceder novamente a esse site, essas configurações anteriores estarem pré-definidas e configuradas.

Os cookies de autenticações em sites web são o método mais comum, e nesta *framework*, este método foi usado para guardar dados relativos ao utilizador. O servidor não precisa de suportar o peso das ligações síncronas para saber se um determinado utilizador está logado ou não e neste sentido, este método torna-se eficaz.

Os dados guardados e relativos ao utilizador que são dados como username, role, layout do webservice (gtheme), layout do menu lateral esquerdo (ttheme), informação relativa às regiões do webservice, visualizada ou omitida (nregion, wregion, eregion, sregion) e um boolean (islogin) para uma mensagem de boas vindas, aquando do login. Cada vez que um determinado utilizador alterar o layout do webservice ou fechar o painel de visualização de determinada informação, são passos que são guardados e durante toda a navegação, estas acções permanecerão guardadas, mesmo em futuros logins desse utilizador.

5.6.2.2 Modelo ER

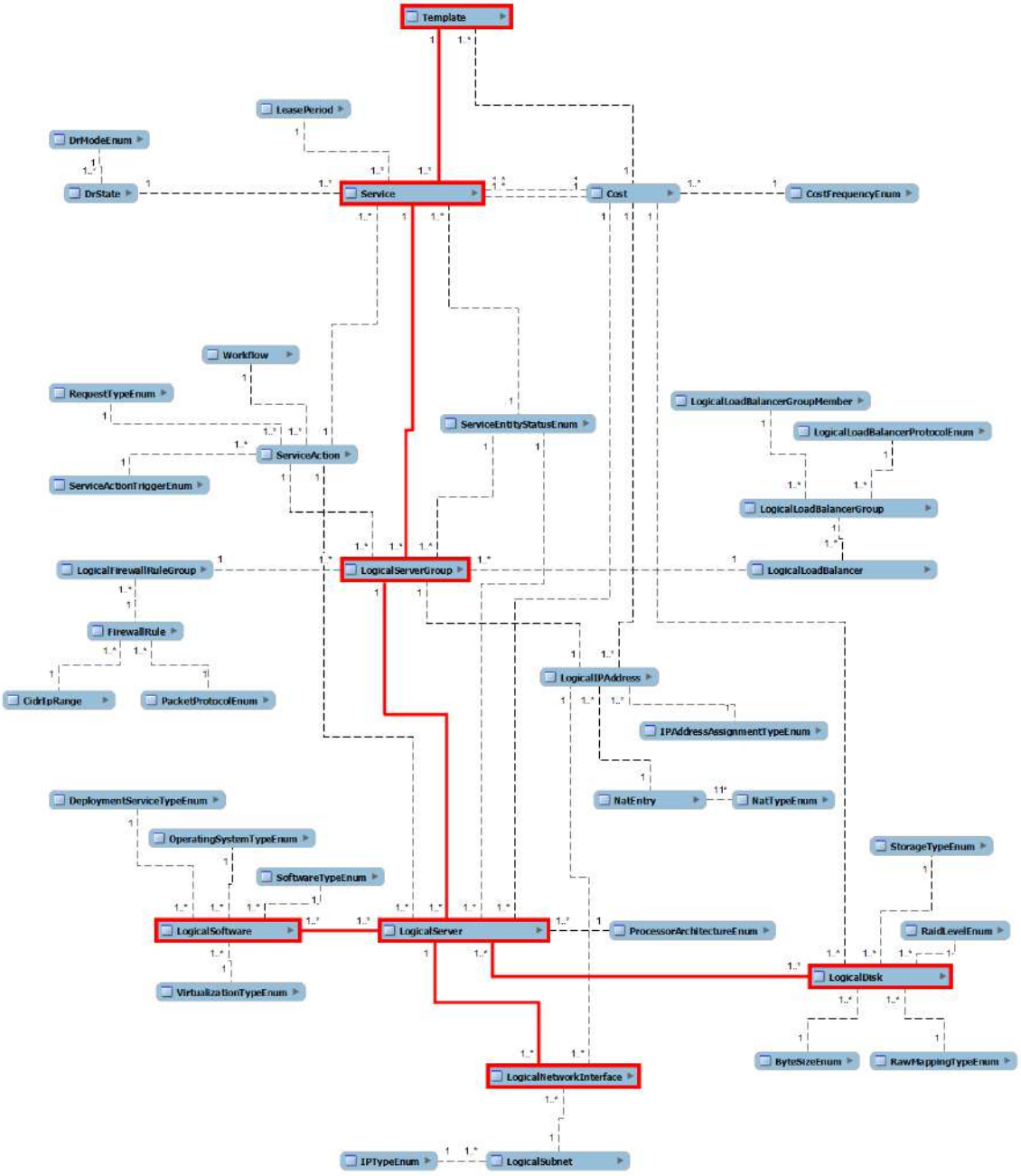
Uma infra-estrutura como o MOE é desenhada e implementada para funcionamento online e todas as alterações como criação de um Serviço, edição desse Serviço, criação de Discos, etc, são alterações com acção imediata na infra-estrutura. Infelizmente, nem tudo corre sempre como queremos e quando mais precisamos de uma ou outra implementação ou configuração é quando o serviço está em baixo e por uma ou outra razão, ficamos impossibilitados de o fazer.

A implementação de uma Base de Dados com toda a informação relativa ao utilizador colmata sempre estas falhas imprevistas. Um implementação, independente da infra-estrutura, que permite ao utilizador ter acesso total aos seus dados, crie, edite, apague recursos em modo offline com sincronização com o MOE automáticamente ou por acção do utilizador.

Um módulo útil e prático que facilita o acesso e configurações dos serviços do utilizador. Acções que carecem de algum tempo de implementação após a sua solicitação. Acções agarradas ao utilizador, ao seu tempo e disposição e não à infra-estrutura. Acções tidas pelo utilizador que, sem preocupação com tempos de execução e implementação, cria, edita ou apaga o seu Serviço e numa futura ligação, tudo é sincronizado automaticamente.

Segue-se o modelo ER implementado que descreve conceptualmente os dados e seus relacionamentos em UML. Em realce, os principais recursos.

Figura 44 - Modelo ER



Pela análise do modelo ER, facilmente se percebe como os recursos se relacionam entre sí. Segue-se uma explicação do modelo e dos seus recursos principais.

Cada Template tem uma lista de Serviços associados que aquando da sua implementação tiveram esse Template como base e apresenta um e um só custo total que corresponde a soma de todos os recursos implementados e permitidos nesse Template.

Um Serviço, como já referido, tem sempre por base um e um só Template, um e um só "*LeasePeriod*" que corresponde ao tempo de utilização e um e um só custo total associado. Este custo é igualmente a soma de todos os recursos instanciados e em uso nesse Serviço. No Serviço, é possível implementar uma nova Entidade denominada "*ServiceAction*" que corresponde a um e um só conjunto de acções a executar automaticamente antes, durante ou depois da criação desse Serviço. O Serviço encontra-se sempre em um e um só estado (*ServiceEntityStatusEnum*) de cada vez que é acedido. Logicamente, como referido ao longo do presente relatório, um Serviço é constituído por uma lista de Grupos de Servidores numa relação de um para muitos.

Um Grupo de Servidores, como o Serviço, apresenta também um e um só conjunto de acções a executar automaticamente antes, durante ou depois da criação desse Serviço nesse Grupo de Servidores em específico. Novas Entidades são aqui retratadas como segurança (*LogicalFirwallRuleGoup*) e balanceamento de carga (*LogicalLoadBalancer*) com a relação de muitos para um. Um Grupo de Servidores está sempre associado a um e um só Serviço e é constituído por uma lista de Servidores numa relação de um para muitos. Uma nova Entidade aparece no Grupo de Servidores (LogicalIPAddress) com a atribuição de um IP único ao Grupo.

Um Servidor relaciona-se com o Grupo de Servidores numa relação de muitos para um, contrariamente à relação com as Entidades LogicalSoftware, e LogicalDisk cuja relação é de muitos para muitos. Já a relação com *LogicalNetworkInterface* é apenas uma relação de um para muitos, um Servidor pode ter muitas Interfaces mas uma Interfaces está sempre associada a um e um só Servidor.

Por último, como componentes do Servidor, Software, Discos e Interfaces. Um mesmo Software pode estar instanciado em muitos Servidores, assim como muitos Servidores podem ter uma lista de Software. Os Discos, com a opção "shared" podem pertencer a vários Servidores e vice-versa. As Interfaces, além de pertencer a um e um só Servidor, tem igualmente um e um só IP associado.

5.7 Feedback

O presente trabalho, como já referido, teve sempre como objectivo último a facilidade de acções do utilizador, visualização e organização da disponibilização de toda a informação e nesse intuito, ao longo de toda a elaboração, foram desencadeadas várias apresentações ao Orientador da Tese, professor Paulo Lopes e Orientador da elaboração na HP, Rui Ramos. Os Feedbacks serviram para redireccionamentos e orientações de programação.

A disponibilização da informação por Tabs transmite toda a informação ao utilizador de uma forma organizada, facilmente acessível e rápida na percepção dessa informação e suas acções possíveis associadas.

A alteração aconselhada pelo Rui Ramos na edição de Servidores do Grupo de Servidores, como a passagem de listagens para um simples "*button*" com um sinal "+" para adicionar Servidores ou um sinal "-" para remover, tornou-se mais prática e intuitiva.

Disponibilização de mais meios de acesso às VMs como a opção VMRC (VMware Remote Console) ou RDP (Remote Desktop Protocol) são sempre opções válidas e úteis para o utilizador final.

A disponibilização da informação e das acções associadas à medida que o utilizador vai descendo no menu em árvore (menu lateral esquerdo) torna-se útil pela visualização hierárquica de todos os recursos e acima de tudo prático ao que o utilizador realmente pretende, com um simples clique.

Personalização do layout aconselhada pelo professor Paulo Lopes com os diversos "Themes", opções de visualização/ocultação de informação, etc, com as valências já descritas no início deste capítulo.

Opção "Datagrid" onde a disponibilização da informação passa de simples Tabs para listagem em tabelas. Visualização útil quando a informação é demasiada, evitando assim o scroll do utilizador na busca da restante informação, bastando para isso selecionar a "quantidade" de informação que pretende visualizar naquele momento.

Feedbacks sempre positivos e atempados para as alterações necessárias a efectuar no *framework*.

5.8 Questões principais

No decorrer da elaboração deste webservice, foram seguidos caminhos e implementações que, de um modo geral, se tornaram práticas e úteis, existindo no entanto algumas desaconselhadas pela especificidade deste tipo de implementações.

Foi implementada a opção de visualização e acções em modo offline. Foi criada uma base de dados que suportava todos os dados do MOE a sincronizar aquando do refresh do webservice ou por acção do utilizador. Tal opção foi desaconselhada pelo facto do sistema estar sempre online, disponível e ter custos associados.

A transformação de datas "epoch2date" e "date2epoch" foi uma das opções que mais trabalho deu pelo facto da API não indicar que o referido número é a data em época e as casas decimais que entram na conversão. Uma época em Unix (o conhecido Unix timestamp) é o número em segundos que distam de 01 de Janeiro de 1970, ou seja, é o tempo Unix 0 que corresponde à meia-noite de 01/01/1970. Por norma, uma data em época é um número inteiro de 32 bits. As duas funções "epoch2date" e "date2epoch" tratam dessas converões para o utilizador poder observar as datas numa formatação legível. Esta formatação ainda carece de uma

transformação do dia com o mês para ser apresentada no formato europeu, dia/mês/ano.

Existe código HTML que é criado dinamicamente, esta opção tornou-se bastante útil pela disponibilização de grandes quantidades de informação nos mesmos "*input boxes*" como é o caso de toda a informação do Servidor dentro de um mesmo Grupo de Servidores. A informação é diferente mas o layout é o mesmo, por isso, uma organização adequada de IDs criados dinamicamente, faz toda a diferença.

O tratamento de erros é algo que por vezes é decurado no dueto utilizador-webservice. Um webservice quer-se amigável e nesse intuito, páginas em branco completamente diferentes do layout do webservice com erros como 404 ou 500 são desaconcelhadas. Assim, foi inicialmente implementado esse tratamento directamente no ficheiro "WebContent\WEB-INF\ web.xml" que redirecionava o utilizador para páginas HTML com layout igual ao webservice, com o respectivo número do erro e a mensagem correspendente para informação. Uma mudança e redireccionamento drástico, obrigando o utilizador a voltar atras e tentar de novo. Optou-se pelo tratamento directo em javascript com a visualização da mensagem de erro numa "InfoWindow" sem redireccionamentos nem novas páginas.

A API do MOE apresenta igualmente alguns pontos que precisam de ser melhorados em futuras versões. Foram implementados alguns caminhos secundários quando tudo poderia ser mais fácil se a API fornecesse essa implementação directamente:

- ✓ *Informação detalhada dos Templates*

O objectivo deste webservice é criar serviços e após criação, fazer uso dos recursos. A implementação de um Serviço tem por base um Template que lhe estará sempre associado. Templates existentes num Catálogo de Serviços e com informação sobre os recursos que ficaram disponíveis após a criação do Serviço.

Naturalmente, o utlizador percorre todos os Templates na busca da informação que melhor se adapte às suas necessidades. Se esse informação se apresenta escassa, o utilizador não vai ficar a saber dos detalhes dos recursos que poderá usar aquando da criação do serviço.

A API não fornece informação detalhada dos Templates, apenas uma informação muito resumida, não disponibilizando ao utilizador a informação associada, por exemplo, ao custo de cada servidor, memória, processador, número de interfaces, software, etc.

O presente webservice já disponibiliza toda essa informação ao utilizador, sendo que este pode navegar no Catálogo de Serviços do menu lateral esquedo e visualizar toda a informação detalhada dos Templates, escolher o que melhor se enquadre dentro das suas necessidades e finalmente, criar o Serviço.

- ✓ *VMID*

O utilizador pode assim usar os recursos depois do Serviço criado. Para a criação de uma sessão remota, o ID da VM era essencial. Tal informação não existe na API do MOE. Uma listagem de VMIDs: por utilizador, associadas a cada serviço, por servidor, por interface, qualquer uma destas quatro opções era válida.

- ✓ *"customTemplateAttributes" na criação do Serviço*

Na criação de um Serviço, existem campos de preenchimento automático, campos obrigatórios e outros opcionais. Assim, é necessário fornecer o "serviceName" (obrigatório), templateName (automatico), leasePeriod (automatico mas o utilizador pode mudar), hostname (obrigatório), serverPoolNames (automático mas o utilizador pode mudar), ownerEmailAddress (obrigatório), note (opcional), billingCode (opcional) e customTemplateAttributes (opcional). A falta da API reside neste último campo, "*customTemplateAttributes*" onde o utilizador poderia escolher previamente o número de servidores, memória, processadores, tipo de endereçamento das interfaces, etc.

- ✓ *"computeServerIds" do ServerPools*

Uma informação valiosa para a criação de Pools de Servidores, não sabendo os IDs dos "*computeServerIds*", não é possível chegar aos "*Compute Resources*" para poder adicionar recursos às Pools. Outra falha detectada.

5.9 Visão global

Uma interface baseada no protocolo REST (Representational State Transfer) para troca de informação e direcionada aos recursos disponibilizados na Web apresenta-se como vantagem quando estamos a lidar com isso mesmo, recursos.

Um conjunto de princípios como HTTP e URIs que, usando REST, exploram a arquitectura Web para o seu benefício usando a simplicidade não só de transporte, como também de protocolos.

Espelhadas as vantagens do protocolo REST, o presente webservice torna-se assim mais simples em comunicação e no acesso aos recursos sem o atraso da sobrecarga apreciável na construção de mensagens do protocolo SOAP (Simple Object Access Protocol) do MOE.

Figura 45 - Mensagens SOAP na invocação de processos do MOE

Figura 46 - Obtenção de recursos através da API RESTfull do WEBSERVICE

Comparativamente à implementação em sí, no MOE existe um único menu de acesso a todas as acções possíveis, as operações de Grupo tornam-se assim cansativas e desmotivadoras, arrantando o utilizador para uma sequência de cliques até a acção final. Opções como edição do Grupo de Servidores não existem, tal é possível apenas na edição directa em cada Servidor. No webservice basta um clique e é executada uma acção de grupo como por exemplo desligar todo o Grupo de Servidores. A edição do Grupo de Servidores é arrastada a todos os Servidores do Grupo apenas com uma simples edição de memória e processamento no Grupo.

Como nas acções de Grupo, se o utilizador pretender uma acção numa interface específica, terá que percorrer a sequência e cliques, contrariamente ao presente webservice com acção directa quer no menu lateral esquerdo, quer na visualização directa da informação da interface.

Toda a informação a apresentar relativamente aos recursos é vasta, requerendo uma organização inteligente nessa apresentação, com opções de visualização/ocultação dessa informação se o utilizador assim o entender. Acções possíveis aquando da visualização de determinada informação devem estar visíveis e acessíveis. No MOE isso é difícil ou quase impossível porque a interface não foi desenhada desse modo, a informação apresentada torna-se desorganizada e desmotivada a leitura.

5.10 Resumo

Este capítulo começa por descreve a arquitectura usada e as suas camadas, seguindo-se um exemplo simplificado das relações da *framework* e plataformas (MOE e VMware). São apresentados conceitos como rapidez, organização e personalização nm desenho guiado pela Interface do Utilizador. Como núcleo deste capítulo, é feita toda a descrição da *framework* por camadas da Arquitectura apresentada: apresentação, lógica/serviços e dados, onde são descritas todas as acções passiveis de serem executadas, por recurso. Por último, os *feedback* tidos ao longo da implementação, algumas questões consideradas importantes e finalmente, uma visão global como foco.

6. Considerações Finais

Esta tese resultou de uma proposta de tema de tese que a HP Portugal submeteu ao Departamento de Informática da FCT/UNL, e que, após algumas interacções, resultou no tema intitulado *Framework para a construção de "portais de negócio" para gestão de solicitações IaaS na HP Cloud*.

A primeira fase, de Preparação de Dissertação, ocorreu no 1º semestre do ano lectivo de 2012/2013, e realizou-se essencialmente neste departamento, pontuada por esporádicas reuniões na HP Portugal (Quinta da Fonte, Paço De Arcos). A segunda, de Elaboração da Dissertação, decorreu imediatamente após a primeira, e cumpriu-se totalmente nas instalações da HP Portugal.

Durante a primeira fase estudou-se o paradigma de computação na cloud, os seus perímetros (modelos de implantação) e modelos de serviço. Estudou-se também a virtualização, tecnologia base da computação na cloud, desde a sua origem ao seu estado actual. Estudaram-se ainda as plataformas de virtualização de servidores com maior presença nos centros de dados, o *ESXi* da *VMware*, o *XenServer* da *Citrix Systems*, e o *Hyper-V* da *Microsoft*, e ainda a tecnologia proprietária da HP, designada *HP Integrity Virtual Machines*. Estudaram-se ainda as tecnologias de armazenamento, abordando-se os aspectos de integridade, fiabilidade e desempenho oferecidos pelos *Redundant Array of Independent Disks*, as redes de armazenamento orientadas ao bloco, *Storage Area Networks*, e aos ficheiros, *Network Attached Storage*. A virtualização do armazenamento – tanto ao nível da sua utilização para armazenar VMs como para constituir "discos" de VMs foi estudada em seguida. Os aspectos da virtualização das redes foram também estudados.

Abordados todos os conceitos e tecnologias base, seguiu-se o estudo da *HP Cloud*, em particular a *HP CloudSystem Matrix* no seu ambiente *Matrix Operating Environment* como infra-estrutura de trabalho da segunda parte da dissertação: um vasto conjunto de componentes software e hardware que vão desde os servidores ProLiant, sistemas de armazenamento EVA, infra-estruturas de rede (switches e NICs), etc, até ao ambiente de operação Matrix com o vCenter. Numa perspectiva mais conceptual estudamos os papéis dos "utilizadores da cloud", administrador, arquitecto e consumidor de serviços, bem como a função do Catálogo de Serviços, do Template, e a possibilidade de definir operações de orquestração da infra-estrutura.

Com a noção exacta do trabalho a implementar, foram estudadas diversas linguagens e paradigmas, e suas vantagens e inconvenientes. Finalmente, foi apresentada uma arquitectura da solução e um cronograma para a segunda fase dos trabalhos. O esforço dispendido na primeira fase foi, por não ter outra tarefa que não fosse a realização da preparação, de uma quase total dedicação a esta.

Durante a segunda parte desta dissertação, a *full-time* na HP, foi desenvolvido um protótipo do *framework* totalmente funcional. A realização conseguida cumpre na íntegra

a calendarização proposta na primeira fase conseguindo-se, à custa de um razoável esforço adicional, evitar atrasos e falhas.

As linguagens usadas foram as já descritas no capítulo 5, e incluem, na camada de apresentação, do lado do cliente, HTML, CSS, JavaScript, JQUERY, AJAX e, do lado do servidor, JSP. Na camada lógica/serviços usou-se fundamentalmente Java e REST.

O desenvolvimento do *framework* foi guiado por duas considerações: uma, do lado do programador e com o pensamento em futuras actualizações, e outra do lado do utilizador com uma interface organizada, intuitiva e de rápido acesso e acima de tudo, que possibilitasse a interacção com todos os recursos do Serviço instanciado, quer a nível de grupos quer a nível dos recursos individuais. Assim, na óptica do programador, organizou-se toda a programação em torno do "recurso", como entidade fundamental, em todas as camadas em que "ele" existe. Deste modo, actualizações ao *framework* podem ser feitas olhando somente para o recurso a actualizar, quer a alteração seja ao nível da informação ou das acções que este suporta.

Em cada uma das duas vertentes, programador e utilizador, foram implementadas diversas ajudas para que estes melhor entendam toda a estrutura e implementação; o referido apoio está disponível no próprio protótipo e em manuais de ajuda em anexo a este relatório.

6.1 Conclusões

A primeira conclusão que apresentamos, sem falsas modéstias, é a de que o portal é, sob o ponto de vista dos resultados conseguidos, e face ao objectivo, um sucesso. Não só é operacionalmente funcional, cumprindo o objectivo essencial, mas também o é na usabilidade. Destacamos, sumariamente, alguns aspectos desta última:

- A visualização da informação, assim como as acções possíveis são guiadas pelos recursos: o utilizador apenas selecciona o recurso e tem tudo disponível, informação e acções que pode executar, tudo associado a esse, e apenas esse, recurso.
- No menu lateral esquerdo do protótipo o utilizador está sempre a visualizar a "posição" de um recurso na hierarquia, tornando-se assim muito fácil executar acções de grupo ou acções individualizadas, bastando para isso, um simples clique.

Na versão 5 da API do MOE detectamos algumas insuficiências que devem ser corrigidas em futuras versões. Nomeadamente,

- Há atributos dos templates cuja consulta apenas é possível ao administrador, quando são fundamentais para uma melhor experiência do utilizador; referimo-nos, por exemplo, aos valores máximos de um dado recurso (e.g., número de máximo de CPUs, custo de um dado disco) não serem disponibilizados aquando da criação de um Serviço. Note-se que o protótipo

ultrapassa esta limitação executando o "init" da sessão como administrador, e carregando nessa altura todos esses atributos.

- Para aceder à consola de um servidor é necessário conhecer o ID da VM, o que não é fornecido pela API MOE; o protótipo ultrapassou esta limitação usando a API disponibilizada pelo VMware vCenter.

Em conclusão, oferecemos um protótipo amigável, intuitivo, com uma curva de aprendizagem muito curta, organizado na vasta informação a disponibilizar e usando, na sua realização tecnologias estado-da-arte no formato de informação, XML e JSON. Um protótipo prático no que ao acesso e utilização dos recursos diz respeito, objectivos últimos de todo este trabalho.

Fica assim aberta uma porta para a equipa de desenvolvimento de software da HP nas próximas decisões sobre o caminho a seguir no desenvolvimento de novas implementações e futuras versões da API. O protótipo foi submetido a testes de usabilidade que podem ser considerados em futuras integrações.

6.2 Trabalho futuro

Um framework que se queira actualizado nunca pode ser considerado acabado; novas actualizações em resposta a solicitações do mercado, ou a alterações na(s) API(s), nos hipervisores, ou decorrentes da necessidade de "hibridização" com outras clouds fazem desse framework um trabalho actualizado mas nunca terminado.

No desenvolvimento deste framework tivemos acesso a uma infra-estrutura que, como é natural (pela complexidade do software necessário), apenas incluía alguma das opções do vasto leque HP CSM; em particular, a) apenas dispunha da pilha de virtualização VMware (ESXi e vCenter), e b) não dispunha do módulo *Storage Provisioning Manager* (SPM).

Do exposto surgem imediatamente duas propostas para trabalho futuro: a primeira passa por dotar o portal de uma capacidade de "identificação automática" da(s) plataformas de virtualização que estão acessíveis ao utilizador, disponibilizando as ferramentas adequadas – por exemplo, se o utilizador apenas tivesse acesso à infra-estrutura Hyper-V, teria apenas o ícone da consola Hyper-V (em vez da consola VMware); a segunda, seria ampliar as funcionalidades do framework para suportar a interacção com a API do SPM, cumprindo assim um dos objectivos que era permitir ao consumidor IaaS escolher as tecnologias de armazenamento – SSD em vez de FC, FCoE em vez de iSCSI, etc..

Uma outra avenida para trabalho futuro seria a avaliação e eventual realização do framework sobre a API OpenStack...

6.3 Resumo

Neste capítulo foi descrito todo o trabalho efectuado nesta dissertação, bem como os resultados obtidos e ainda os "alertas" relativos a aspectos que consideramos merecedores de atenção por parte da HP. Tal só foi possível pelo detalhe com que toda a infra-estrutura e API foram estudadas e testadas. Foram referidas as principais contribuições e destacaram-se as melhorias relativamente ao portal actualmente disponibilizado pela HP.

Finalmente, deixam-se algumas propostas para trabalhos futuros...

Referências bibliográficas

[1] Understanding the HP CloudSystem Reference Architecture. [Online]. Available: http://h20195.www2.hp.com/V2/GetPDF.aspx/4AA0-5550ENW.pdf

[2] Amazon Elastic Compute Cloud (Amazon EC2): [Online] Available: http://aws.amazon.com/pt/ec2/

[3] GoGrid Cloud Hosting: [Online] Available: http://www.gogrid.com/

[4] Eucalyptus: [Online] Available: http://www.eucalyptus.com/

[5] Google AppEngine: [Online] Available: https://developers.google.com/ /appengine/

[6] Microsoft Azure: [Online] Available: http://www.windowsazure.com/pt-br/

[7] Google Apps: [Online] Available: http://www.google.com/enterprise/apps/ /business/

[8] Agensen, O. et al. The Evolution of an x86 Virtual Machine Monitor. ACM Operating Systems Review, Vol. 44, Number 4, December 2010, pp. 3-18.

[9] VMware Virtualization Products for Virtual Servers, Virtual Desktops, and Data Center. [Online]. Available: http://www.vmware.com/products/

[10] Comparison of Platform Virtual Machines. Wikipedia, the Free Encyclopedia, Dezembro 12, 2012. [Online]. Available: http://en.wikipedia.org/w/index.php? ?title=Comparison_of_platform_virtual_machines&oldid=527691634

[11] Vsphere-51-configuration-maximums, VMware Inc, 2012.

[12] Barham, P. et al. Xen and the art of virtualization. In Proceedings of the nineteenth ACM symposium on Operating systems principles (SOSP'03), 2003, pp. 164-177.

[13] Zhang, X. and Dong, Y. Optimizing Xen VMM Based on Intel® Virtualization Technology. In Proceedings of the 2008 International Conference on Internet Computing in Science and Engineering (ICICSE '08). IEEE Computer Society, pp. 367-374.

[14] CTX134582 - XenServer 6.1.0 Release Notes - Citrix Knowledge Center. [Online]. Available: http://support.citrix.com/article/CTX134582

[15] Haga, Y., et al. Windows Server 2008 R2 Hyper-V Server Virtualization. Fujitsu Sci. Tech. J., Vol. 47, No. 3, 2011, pp. 340-355.

[16] Feature Comparison: Windows Server 2008 R2 Hyper-V and Windows Server 2012 Hyper-V, Windows Server, Microsoft Corporation, 2012.

[17] Hyper-V Scalability in Windows Server 2012. [Online]. Available: http://technet.microsoft.com/en-us/library/jj680093.aspx

[18] Patterson, D. et al. A Case for Redundant Arrays of Inexpensive Disks (RAID). Proceedings of the 1989 ACM-SIGMOD International Conference on the Management of Data, ACM, 1989, pp. 109-116.

[19] Whitehouse, S. The GFS2 Filesystem. Proceedings of the Linux Symposium, June 27th-30th 2007, Ottawa, Canada.

[20] Schmuk, F. and Haskin, R. GPFS: A Shared-Disk File System for Large Computing Clusters. Proceedings of the Conference on File and Storage Technologies (FAST'02), 28–30 January 2002, Monterey, CA, pp. 231–244.

[21] Vaghani, S. Virtual Machine File System. ACM Operating Systems Review, Vol. 44, Number 4, December 2010, pp. 57-70.

[22] Pawlowski, B. et al. NFS version 3 design and implementation. Proceedings of the Summer USENIX Conference, June 1994, pp 137-152.

[23] Zhou, S. Virtual Networking. ACM Operating Systems Review, Vol. 44, Number 4, December 2010, pp. 80-85.

[24] HP Sitescope 112 Data Sheet [Online]. Available: http://www.hp.com/hpinfo/newsroom/press_kits/2012/HPDiscover2012/HP_Sitescope_112_Data_Sheet.pdf

[25] Veritas Storage Foundation™ and High Availability Solutions Application Note: Support for HP-UX Integrity Virtual Machines, Symantec Corporation, 2011.

[26] Kennedy, S. et al. StoRHm: a protocol adapter for mapping SOAP based Web Services to RESTful HTTP format. Electronic Commerce Research, September 2011, Volume 11, Issue 3, pp. 245-269.

[27] Silvia Schreier. Modeling RESTful applications. In Proceedings of the Second International Workshop on RESTful Design (WS-REST '11), Cesare Pautasso and Erik Wilde (Eds.). ACM, New York, NY, USA, 2011, pp. 15-21.

[28] Axioms. [Online] Available: http://www.w3.org/DesignIssues/Axioms.html

[29] Sandoval, Jose. Restful Java Web Services. Master core REST concepts and create RESTful web services in Java. Packt Publishing, November 2009, pp. 43-179

[30] Burke, Bill. Restful Java with Jax-RS. O'Reilly Media, Inc. November 2009, 1st ed.

[31] Jersey 1.17 User Guide. [Online] Available: https://jersey.java.net/documentation/1.17/index.html

[32] Jerome Louvel, Thierry Templier, and Thierry Boileau. Restlet in Action: Developing Restful Web Apis in Java. Manning Publications Co., 2012, Greenwich, CT, USA, pp. 1-165

[33] Johnson, Sally. Cloud providers can enable 'self-service' IT with a cloud portal. [Online]. Available: http://searchcloudprovider.techtarget.com/feature/Cloud-providers-can-enable-self-service-IT-with-a-cloud-portal?asrc=EM_ERU_22442845&utm_medium=EM&utm_source=ERU&utm_campaign=20130709_ERU%20Transmission%20for%2007/09/2013%20(User Universe:%20607658)_myka-reports@techtarget.com&src=5144272

[34] jQuery EasyUI framework. [Online]. Available: http://www.jeasyui.com/

[35] jQuery UI Library. [Online]. Available: http://jqueryui.com/demos/

[36] jQuery API. [Online]. Available: http://api.jquery.com/

Anexo A

Virtualização: Vantagens e inconvenientes

A virtualização tem vindo a crescer e a ganhar terreno principalmente nas médias e grandes empresas. Tem vindo a demonstrar a sua mais-valia na gestão das TIs, uma mudança completa e um aliviar de trabalho para administradores e gestores.

A.1 Vantagens da tecnologia

Um mundo onde passa a ser possível executar software incompatível com o SO host, executar SO's distintos num mesmo conjunto de hardware, isoladamente, sem interferências, etc, uma mudança drástica na tradicional visão de um SO para um conjunto de hardware, apenas.

- ✓ *Segurança:* Dentro de um servidor podemos ter vários VMs isoladas e independentes do SO host. Tal garante que qualquer interferência no SO host (erros/ataques acidentais ou intencionais) não afecte as VMs nele instaladas;
- ✓ *Isolamento:* Qualquer processo de uma VMs não pode interferir nos processos de outra VMs, este isolamento, intimamente ligado à segurança vem de encontro a um dos princípios básicos aquando da implementação desta tecnologia. Esta vantagem garante-nos confidencialidade entre VMs, assim como integridade e contenção de erros entre VMs (segurança);
- ✓ *Custos:* A quantidade de diferentes servidores existentes para diferentes SO's e aplicações é agora compactada num mesmo hardware onde existe a possibilidade de termos várias VMs com diferentes SO's com as mais variadas aplicações. Temos um tudo-em-um que vai reduzir os custos de uma organização na altura do upgrade, bastando para isso o upgrade de um único servidor que suporta todos os serviços da organização;
- ✓ *Energia:* Propriedade intimamente ligada aos custos. Com a compactação do hardware são dispensando todos os demais equipamentos, quer de sustentação de toda uma infra-estrutura, quer os próprios equipamentos de apoio. É assim conseguida uma economia de energia bastante significativa para a organização;
- ✓ *Espaço Físico:* A compactação é inerente á diminuição do espaço ocupado. Um Data Center tem sempre o problema de "falta de espaço" e por isso, a compactação de hardware associada à virtualização vem resolver não só este problema de espaço como também a própria refrigeração e circulação de ar dentro do Data Center;

- ✓ *Hardware:* Está provado que cada servidor, em média, gasta cerca de 30% do seu poder de processamento para executar a função para a qual foi implementado. Assim sendo, estamos na presença de um subaproveitamento deste poder computacional. A ideia anterior á virtualização de adquirir um novo servidor para uma nova aplicação, cai agora por terra com o objetivo de aproveitar ao máximo o poder computacional de cada servidor, aproveitando melhor todo esse hardware disponível;
- ✓ *Implementação de SO's:* A associação de um SO por cada máquina fica agora também em desuso uma vez que a virtualização possibilita-nos a instalação de vários SO's em cada máquina, melhor, além de vários SO's, estes também podem ser diferentes entre si (Windows, linuz, etc). Esta vantagem permite assim instalar SO's cujo hardware original (host) nem sequer os suportam, como o caso do SO MAC (Macintosh, Apple) em hardware x86 ou SO Windows em notebooks (Apple);
- ✓ *Migração:* A camada que suporta a virtualização permite-nos esta mais-valia. A migração não só do ambiente em si como do próprio hardware, bastando para isso uma simples cópia entre discos virtuais;
- ✓ *Testes de software e hardware:* A virtualização simula tanto um software como um hardware, logo, testes em software, firmware e hardware podem ser realizados nestas VMs construídas especificamente para esse fim. Estes primeiros testes podem ser efectuados sem a necessidade de um SO instalado de raiz ou mesmo a presença física do hardware específico;
- ✓ *Redução de administradores e gestores:* Esta compactação associada à virtualização, como já referido, reduz espaço ocupado, reduz hardware, reduz configurações e gestão individual, abrindo caminho à centralização. Assim, a redução da equipa responsável pela administração e gestão é evidente.

A.2 Desvantagens da tecnologia

São notórias as vantagens da virtualização mas problemas como incompatibilidades entre aplicações e perda de desempenho pelo simples adicionar de mais uma camada à arquitectura terão que ser consideradas aquando da escolha.

- ✓ *Segurança:* A segurança, neste caso, prende-se à própria camada adicionada à arquitectura (VMM ou hypervisor). Se o próprio SO host tiver algum problema de raiz, implementado ou adquirido, também o próprio VMM contemplará este erro assim como todas as VMs instaladas nesse SO host;
- ✓ *Gestão:* O principal problema da implementação desta tecnologia é a gestão da própria virtualização, os ambientes virtualizados precisam de ser instanciados, geridos, configurados, etc. É neste ponto que se prende a preocupação das organizações pelos seus investimentos nesta tecnologia, trata-se de uma opção que, por norma, é licenciada;

- ✓ *Desempenho:* Mais uma camada de software (VMM ou hypervisor) implica mais processamento; Incerteza da quantidade de VMs/processador sem que seja afectado a QOS (Quality Of Service) e a garantia dessa emulação em termos da disponibilização máxima que o host pode dispensar como é o caso do processamento e memória gráfica.

Anexo B

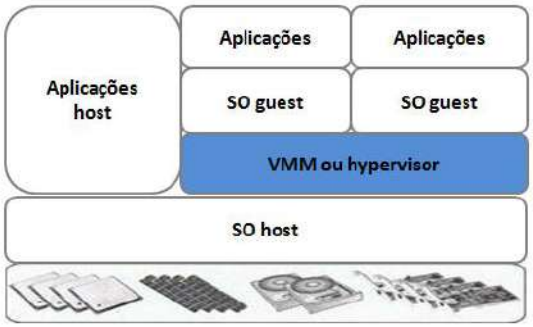
Virtualização: Arquitectura

A base de uma arquitectura de virtualização é o VMM ou hypervisor, o Virtual Machine Manager responsável pela criação, gestão, isolamento e preservação do estado da VM, assim como toda a orquestração do acesso aos recursos do sistema host.

O VMM permite que seja possível a execução de vários SO's em VMs, contudo está limitado a SO's que possam ser executados nativamente no processador físico do sistema. Essa valência torna-se hoje uma das maiores procuras de todos os sistemas de virtualização.

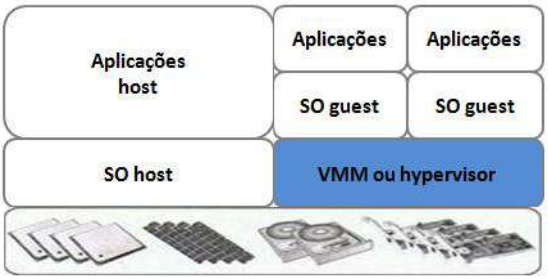
Em termos de implementações arquitecturais temos VMM's: Tipo-2, modelo "hybrido" e Tipo-1. O VMM ou hypervisor do Tipo-2, figura 47, é implementado dentro do próprio SO real (Linux ou Windows - hostsystem) e executando paralelamente, é mais um processo. Neste caso, trata-se de uma camada hypervisor própria como um segundo e distinto nível de software. Os sistemas operativos convidados correm num terceiro nível acima do hardware. Nesta arquitectura, o SO guest acede directamente o SO host (nativo) por intermédio de uma API cedida pelo hipervisor ao SO guest. O SO guest acede assim ao hardware através de um device driver específico pelo hypervisor e pelo SO host, sendo por isso uma implementação que apresenta menor desempenho e maior sobrecarga ao sistema. O Java VM, VMware Workstation, VMware Player, Sun Microsystems VirtualBox e KVM são alguns exemplos que usam esta arquitectura.

Figura 47 - VMM ou hypervisor do Tipo-2



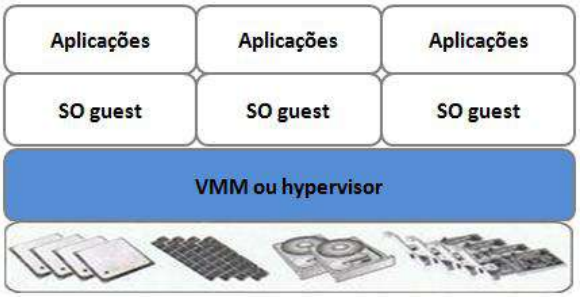
No modelo "híbrido", figura 48, o VMM não é executado uma camada acima do SO host (ou abaixo) mas sim concorrentemente (peer) com o SO host e não depende de instruções específicas no processador. Microsoft Virtual Server 2005 R2 e Virtual PC são alguns exemplos que usam esta arquitectura.

Figura 48 - VMM ou hypervisor do “híbrido”



No VMM ou hypervisor do Tipo-1, figura 49, é executado directamente no hadware, a camada mais baixa de todas as partições da VM, sem necessidade da existência de qualquer SO host. O SO guest acede directamente ao hardware através do hipervisor, tal é possível pelas modificações feitas no SO guest e no hipervisor. Esta é a implementação que atinge níveis mais elevados de eficiência por isso permite uma maior densidade de VMs, é a implementação clássica de arquitecturas VM. VMware ESX/ESXi, Citrix XEN Server e Microsoft Hyper-V são alguns exemplos que usam esta arquitectura.

Figura 49 - VMM ou hypervisor do Tipo-1



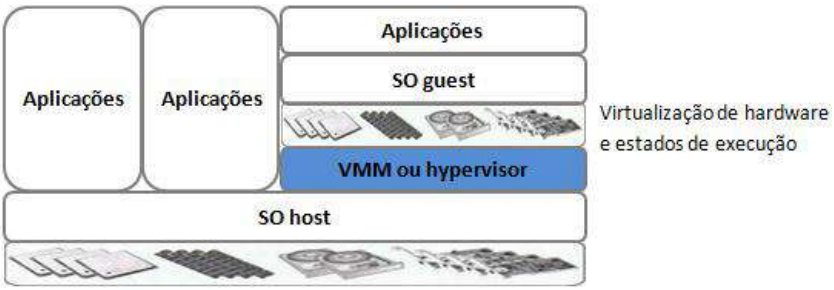
Anexo C

Virtualização: Tipos de Implementação

C.1 Emulação de Hardware

Neste tipo de virtualização, figura 50, pretende-se a virtualização completa do hardware, o seu comportamento/estados de execução (ciclos de clock, conjunto de instruções, pipeline, memória cache), por isso é a forma de virtualização mais complexa. Todo o hardware da VM é criado via software no sistema hospedeiro para emulação do hardware proposto criando assim grandes overheads[7], com fracos desempenhos comparativamente ao hadware real. Este tipo de virtualização apresenta mais-valias como o facto de poder ser usado um SO guest sem qualquer modificação ou adaptação, os programadores podem fazer testes de firmware e hardware em hardware que não o real nem necessitam da existência deste e ainda a valência de ser possível emular hardware que, na maior parte das vezes é bastante diferente do hardware real.

Figura 50 - Emulação de Hardware



C.2 Virtualização completa

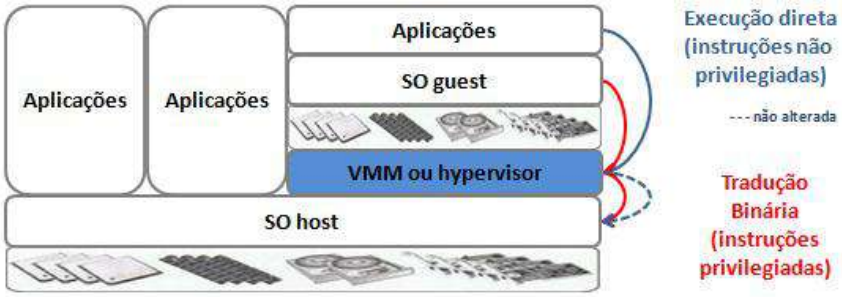
A virtualização completa, figura 51, também conhecida como "nativa", é em todo igual a emulação de hardware excepto à não representatividade dos estados de execução do hardware emulado, por esse facto, esta técnica de virtualização apresenta resultados mais satisfatórios no seu desempenho mas ainda aquém dos SO's em execução nativa. Esta técnica permite assim a execução de todos os SO's originais, sem alteração, fornecendo uma réplica de todo o hardware dessa máquina, levando-os a ter a ilusão de estar a executar directamente no hardware.

Os grandes obstáculos desta técnica foram desde logo as diferenças arquitecturais e os comportamentos particulares das instruções, muitas impossíveis de serem emuladas ou capturadas pois muitas dependem do nível de privilégio. Assim, para lidar com a

[7] Overhead: Diferença de performance (latência/lentidão) no SO guest

heterogeneidade de processadores e comportamento de instruções, usa-se uma abordagem chamada de Tradução Binária. Nesta abordagem, a VMM analisa todas as instruções da VM, quer instruções não privilegiadas que depois acedem ao hardware com drivers genéricos, quer instruções privilegiadas e quando na presença destas, faz a emulação e reescreve dinamicamente o código. Este teste a todas as instruções acarreta uma latência significativa no desempenho.

Figura 51 - Virtualização Completa

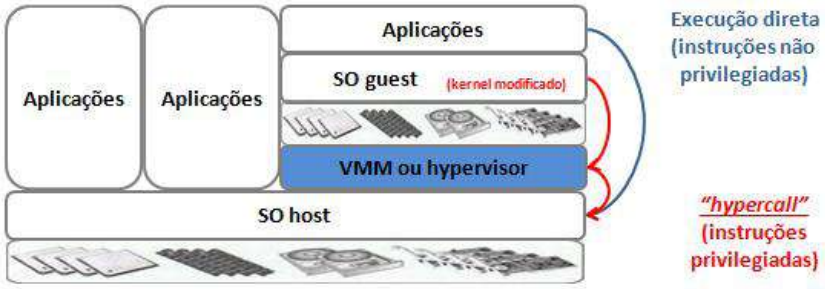


C.3 Para-Virtualização (PV)

A para-virtualização ou virtualização assistida por SO, figura 52, é uma alternativa á virtualização completa. O SO guest é modificado (perdendo a portabilidade) para melhorar a eficiência usando todos os "limites" do sistema. A ideia é acelerar a execução das instruções e para isso a VMM testa apenas instrução que podem alterar o estado do sistema (instruções sensíveis) aumentando significativamente o desempenho. Esta substituição de uma instrução sensível pelo tratador de interrupção de software é chamada de hypercall, são instruções executadas através de tradução binária. O SO guest é assim modificado para chamar a VMM sempre que estamos na presença dessas instruções sensíveis.

Esta alternativa de virtualização com a modificação do kernel, também permite que o SO guest aceda directamente aos recursos de hardware com os drivers da própria máquina virtual e deste modo, não usando os drivers genéricos (virtualização completa), usam a capacidade total dos dispositivos. Assim, a para-virtualização apresenta um ganho significativo em relação à virtualização completa, compensando as modificações implementadas no SO guest.

Figura 52 - Para-Virtualização

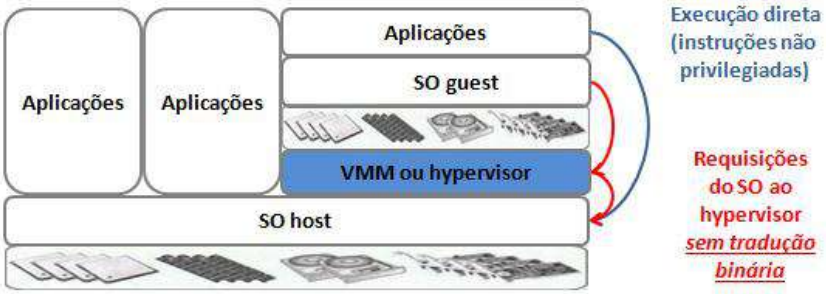


C.5 Virtualização assistida por Hardware (HVM)

As técnicas descritas apresentam alguns inconvenientes que enfraquecem o desempenho como o teste a todas as instruções e tradução das instruções privilegiadas na Virtualização Total e a alteração do SO guest na Para-Virtualização, agarrando o SO á arquitectura, limitando a compatibilidade e suporte.

A Virtualização assistida por Hardware, figura 53, é aplicada directamente nos processadores e restante hardware, são usadas extensões de virtualização do processador e hardware para virtualização dos guest's, fornecendo assim um recurso que permita ao hypervisor executar ainda mais próximo do hardware. As instruções sensíveis são agora entregues sem a necessidade da tradução binária aumentando significativamente o desempenho mas ainda um pouco mais lento que a PV. A opção do uso de drivers como a PV torna-se um avanço notório neste tipo de implementação. Não emular I/O e armazenamento e ainda oferecer uma plataforma de VMs com SO's não modificados é uma junção das mais-valias dos dois mundos (PVHVM ou PV-on-HVM drivers). Torna-se assim possível, por exemplo, obter um óptimo desempenho no conjunto SO guest Windows + XenServer.

Figura 53 - Virtualização assistida por Hardware



C.4 Recompilação dinâmica

Esta técnica de virtualização, recompilação dinâmica (dynamic recompilation - DynaRecs), também denominada como tradução dinâmica (dynamic translation) é bastante utilizada. Traduz as instruções de um determinado formato para outro formato, durante a própria execução do programa, permitindo assim a criação do ambiente nativo

do programa. Essa tradução é relativa a instruções do SO guest e das suas aplicações, mais próximas do SO host. Assim, o VMM ou hipervisor analisa, reorganiza e traduz as sequências de instruções emitidas pelo SO guest em novas sequências de instruções (código de mais alto nível) compiladas na linguagem nativa do sistema host para que esse código gerado seja mais eficiente.

Essa eficiência de código prende-se com adaptação de instruções à interface ISA do sistema real, detectar e tratar instruções sensíveis e ainda analisar, reorganizar e optimizar sequências de instruções do SO guest com o fim último da eficiência e desempenho da sua execução. Nesta ultima análise, é usual guardar em cache a tradução dos blocos de instruções frequentes, melhorando assim ainda mais o desempenho. É notória a vantagem desta técnica como o acesso a código em tempo de execução não alcançável a um compilador estático mas em contrapartida, também exige muito mais processamento.

Anexo D

Detalhes da plataforma e restrições da VMWARE

Tabela 13 - Dados gerais VMWARE

Criador	VMM		Tipo	Finalidade	Uso	Host	VMs /host	Servidor dedicado	Gestão Central	Gestão Remota	Guest	OS's Host	OS's Guest	Licença
VMware	Player		Tipo 2	Descktop, Utilizadores finais	Criar, Testar, Segurança	Intel e AMD (32-64 bits)	1-2	Não	Não	Não	Intel e AMD (32-64 bits)	Windows	Windows ✓ Windows Server 2012, 8, Server 2008 R2, 7, Server 2008, Vista, Server 2003, XP, 2000, NT 4.0, ME, 98, 95, MS-DOS 6.22 e Windows 3.1x Linux ✓ Asianux Server, CentOS, Debian, Fedora, Mandrake, Mandriva, Novell, openSUSE, Oracle Enterprise, Red Hat Enterprise, Red Hat, Sun Java Desktop System, SUSE Enterprise, SUSE, Turbolinux, Ubuntu UNIX e outros OS's ✓ Mac OS X Server, eComStation, FreeBSD, IBM OS/2 Warp, Netware, Solaris, SCO	Livre
	Server GSX		Tipo 2		Consolidar, Testar, Desenvolver							Windows, Linux		Proprietário **(desc.)**
	Workstation		Tipo 2		Criar, Testar, Desenvolver		< 10	Não	Não	Não		Windows, Linux		Proprietário
	Fusion		Tipo 2									Mac OS X		Proprietário
	vSphere	ESXi	Tipo 1	Datacenter, Infra-estrutura Cloud	Consolidar, Disponibilidade, Cloud, Criar, Testar, Desenvolver		512	Sim	Não	Sim		Sem OS		Livre, Proprietário
		ESX	Tipo 1				512	Sim	Sim	Sim		Sem OS		Proprietário **(desc.)**

Tabela 14 - Detalhes VMWARE

VMM (Configurações máximas)		Cluster	Hosts	VM´s	CPU		RAM		Storage/Disco		Rede	Outros
					Host	VM	Host	VM	Host	VM		
VMware Player 5.0		-	-	-	4	8	Sem lim	8GB (32bits) 64GB (64bits)	N/A	(2TB)	-	-
VMware Server 2.0		-	-	64	16	2	Sem lim	8GB	N/A	4 IDE + 60 SCSI (950GB)	-	descontinuado
VMware Workstation 9.0		-	-	-	igual	8	Sem lim	8GB (32bits) 64GB (64bits)	N/A	4 IDE + 60 SCSI (2TB)	10 NICs SO Windows 255 NICs SO Linux	Snapshots
VMware vSphere 5.1	VM	-	-	-	-	64	-	1TB	-	4 IDE + 60 SCSI (2TB)	10 NICs	128MB gráfica
	ESX / ESXi	32	-	512/host 4000/cluster	32 fisicos 160 logicos	32	2TB	1TB	2048 vdisc (64TB)	4 IDE + 60 SCSI (2TB)	32portas Ethernet a 1Gb 4 NICs físicos a 10Gb Portas Infiniband, etc	-
	vCloud Director	-	2000	30000	-	-	-	-	10000 DC	-	10000	-
	vCenter Server	-	1000	15000	-	-	-	-	500	-	10 vCenter Servers ligados	-

Anexo E

Detalhes da plataforma e restrições da Citrix

Tabela 15 - Dados gerais Citrix

Criador	VMM ou hypervisor	Tipo	Finalidade	Host	Guest	OS's Host	OS's Guest	Licença
Xensource	Xen	Tipo 1	Criar, Testar, Desktop, Desenvolver, Consolidar, Disponibilidade, Cluster e Cloud	x86, x86-64, IA-64	x86, x86-64, IA-64	NetBSD, Linux, Solaris	FreeBSD, NetBSD, Linux, Solaris, Windows XP, Windows Server 2003, Plan 9 **Versão 6.1:** Ubuntu, CentOS, Red Hat Enterprise Linux, Oracle Enterprise Linux , Windows 8 (32-bit/64-bit), Windows Server 2012	GPL – livre Advanced Edition - Proprietário Enterprise Edition - Proprietário Platinum Edition - Proprietário

Tabela 16 - Dados detalhados Citrix

VMM (Configurações máximas)	cluster	VM´s	CPU		RAM		Storage/Disco		Rede/host	Outros
			Host	VM	Host	VM	Host	VM		
XenServer 6.1	16	150/host 1600/cluster	160 lógicos 900 virtuais	32	1 TB	128 GB	512 virtuais	16 virtuais NFS e LVM (2TB)	7 NIC's virtuais/VM 16 NIC's físicos 512 NIC's virtuais 800 VLAN's	4 GPUs/host.

Anexo F

Detalhes da plataforma e restrições da HP

Tabela 17 - Dados gerais HP

Criador	VMM ou hypervisor	Tipo	Finalidade	Host	Guest	OS's Host	OS's Guest	Licença
Hewlett-Packard	Integrity Virtual Machines	Tipo 1	Criar, Testar, Desktop, Desenvolver, Consolidar, Disponibilidade, Cluster e Cloud	IA-64	IA-64	HP-UX 11i v3	HP-UX 11i v2 e v3, Windows Server 2003 (SP1 e SP2 RHEL AP 4.4, 4.5 SLES 10 SP	Proprietário

Tabela 18 - Dados detalhados HP

VMM (Configurações máximas)	cluster	VM´s	CPU		RAM		Storage/Disco		Rede	Outros
			Host	VM	Host	VM	Host	VM		
HP Integrity Virtual Machines 6.1		512	HP-UX limit	16	HP-UX limit	128GB	HP-UX limit	4 IDE + 256 SCSI (2TB)	62 NIC's	

Anexo G

Detalhes da plataforma e restrições da Microsoft

Tabela 19 - Dados gerais Microsoft

Criador	VMM ou hypervisor	Tipo	Finalidade	Host	Guest	OS's Host	OS's Guest	Licença
Microsoft	Virtual PC 2007	Tipo 2	Criar, Testar, Hobby, Estação de trabalho, Negócios	x86, x86-64	x86	Windows Vista (Business, Enterprise, Ultimate), XP Pro, XP Tablet PC Edition	DOS, Windows, OS/2, Linux (SUSE, Xubuntu), OpenSolaris (Belenix)	Proprietário
	Virtual PC 7 for Mac	Tipo 2		PowerPC	x86	Mac OS X	Windows, OS/2, Linux	Proprietário
	Virtual Server 2005 R2	Tipo 2	Server, Server farm	x86, x86-64	x86	Windows 2003, XP	Windows NT, 2000, 2003, Linux (Red Hat, SUSE)	Proprietário
	Hyper-V Server 2008 R2	Tipo 1	Criar, Testar, Desktop, Desenvolver, Consolidar, Disponibilidade, Cluster e Cloud	x86-64 (Intel VT-x ou AMD-V)	x86-64, x86 (até 8 CPUs físicos)	Windows 2008 com Hyper-V, Windows Hyper-V Server	Windows 2000, 2003 e 2008, Windows XP e Vista, Linux (SUSE 10)	Proprietário
	Hyper-V Server 2012	Tipo 1		x86-64 (Intel VT-x ou AMD-V apenas para RemoteFX)	x86-64, (até 64 CPUs físicos)	Windows 2012 com Hyper-V	Windows NT Linux (SUSE10, RHEL 6, CentOS 6)	Proprietário

Tabela 20 - Dados detalhados Microsoft

VMM (Configurações máximas)	cluster	VM´s	CPU		RAM		Storage/Disco		Rede	Outros
			Host	VM	Host	VM	Host	VM		
Hyper-V Server 2008 R2	16	384/host 1000/cluster	8 físicos 64 lógicos 512 virtuais	4	1 TB	64GB	No limit	4 IDE + 256 SCSI (2TB)	-	-
Hyper-V Server 2012	64	1024/host 8000/cluster	64 físicos 320 lógicos 2048 virtuais	64	4 TB	1 TB	No limit	4 IDE + 256 SCSI (VHD 2 TB) (VHDX 64 TB)	32 NIC's 10GB migração/cluster	REST, powershell , snapshots, suporte VMConnect para RemoteFX

Anexo H

Ambiente de armazenamento VMware ESXi e Citrix Xen

H.1 VMware (VmWare In)

VMFS (Virtual Machine File System), é um sistema de ficheiros num Cluster, é proprietário da VMware e foi optimizado para operações de entrada e saída (I/O) com ficheiros grandes, realidade das imagens das máquinas virtuais. Representa uma solução escalável e de alto desempenho para hospedagem de VMs no bloco de armazenamento, oferecendo partilha e isolamento. O serviço VMFS constitui a base de aplicações num cluster VMware como vMotion (onde existe partilha de dados), Storage VMotion, agendamento de recursos distribuídos, alta disponibilidade e tolerância a falhas. Os metadados da máquina virtual são serializados para ficheiros e o software VMFS fornece uma interface POSIX (Portable Operating System Interface) - API, com a opção de consola Command Line (CL) e utilidades de interfaces) - para cluster com operações seguras de gestão de máquinas virtuais.

Este tipo de armazenamento de dados pode ser feito em qualquer dispositivo de armazenamento com base em ligações de armazenamento como:

- ✓ *Armazenamento de ligação direta (SCSI local):* ligação local direta aos discos de armazenamento interno ou externo, geralmente SATA, IDE (Integrated Drive Electronics), EIDE, etc;
- ✓ *Fibre Channel:* protocolo de transporte usado em SAN's onde o Fibre Channel encapsula comandos SCSI sobre o protocolo FC;
- ✓ *FCoE:* O uso do FC sobre Ethernet onde o tráfego FC é encapsulado em quadros Ethernet;
- ✓ *iSCSI:* iSCSI transporta comandos SCSI sobre redes IP;
- ✓ *IP:* ligação IP aos datastores em NAS suportados pelo vSphere ESX.

Vista do armazenamento pelo SO Guest

Uma VM na arquitectura ESX usa espaço em disco para armazenar os seus metadados de configuração e os próprios dados, um pequeno número de imagens, aproximadamente 30 a 100 por VM. O espaço de armazenamento em disco contem estes ficheiros de texto muito pequenos (bytes ou KB) mas também contém ficheiros do próprio disco virtual que normalmente são ficheiros muito grandes (centenas de MB ou GB) e ainda ficheiros de swap (ficheiros guardados pelo SO para aumento virtual da memória de trabalho (RAM)).

Um dos princípios da virtualização diz-nos que um SO guest não deve suportar a carga da gestão de dispositivos físicos, deve concentrar-se em fornecer um ambiente de

execução de aplicações. Assim, um dos objetivos da arquitetura VMFS e em particular do vSphere ESX é simplificar todo este acesso de armazenamento ao SO guest. O hardware virtual apresentado a um SO guest vSphere inclui um conjunto de controladores SCSI e IDE que possuem drivers comuns disponíveis em quase todos os sistemas operacionais. O SO guest vê um disco físico simples ligado através de um controlador padrão, gravado no formato VMDK (Virtual Machine Disk), formato da VMWare. Esta visão virtualizada de armazenamento apresentada ao SO guest tem uma série de vantagens, como a gestão facilitada pelo facto de serem usados controladores padrão com drivers incluídas nos SO's, o facto de suportar diferentes ligação e acessos ao armazenamento como iSCSI e FCoE disponíveis nos SO's sem que estes suportem os protocolos nativos e a melhoria de eficiência e disponibilização na recuperação de desastres com a disponibilização de múltiplos caminhos no acesso aos dados, contrariamente ao peso da redundância de múltiplos servidores físicos.

H.2 Xen Server (Citrix)

O VMM ou hypervisor Xen é apenas responsável pela gestão de CPU, memória e interrupções, ou seja, não tem conhecimento do I/O de Rede e Armazenamento. Para lidar com entradas e saídas de Rede e Armazenamento, o XenServer dispõe de um domínio de controlo, uma VM especializada denominada "Dom0" (Domínio 0), com SO Linux e privilégios especiais, é considerado parte integrante do XenServer aquando da instalação. Alem de aceder diretamente o hardware, controla todos os acessos de I/O e interage com as outras VMs. Esta VM especial contem os drivers padrão do código aberto Linux de uma vasta diversidade de hardware e uma interface de controlo (consola ou interface gráfica) para controlo do sistema. Contém ainda uma pilha de controlo (toolstack) para fazer toda a gestão das VMs, as conhecidas "DomU's" (Domínios dos utilizadores) relativamente a sua criação, destruição e configuração.

O XenServer da Citrix armazena as suas VMs diretamente no formato Microsoft VHD (Virtual Hard Disk) num disco local não partilhado (EXT) ou contrariamente ao VMware, não dispondo do seu próprio sistema de armazenamento, aproveita as capacidades nativas de armazenamento como por exemplo um sistema de armazenamento partilhado como o NFS. Com armazenamentos baseados em iSCSI ou FC, o XenServer estende o VHD do sistema de ficheiros para o padrão open source LVM (Logical Volume Manager) para gestão de volumes numa LUN, abstraindo os volumes físicos em volumes lógicos. Permite ainda que seja possível a criação de uma LUN por cada VDI.

O XenServer oferece assim suporte de armazenamento de VMs em discos locais, iSCSI ou SAN baseado em FC ou NAS, estes dois últimos para migrações e alta disponibilidade. Ferramentas como XenCenter Storage Repository para implementação e

StorageLink para integração de armazenamentos com snapshots, clones[8] e "thin provisioning"[9] são uma mais-valia no XenServer.

Cada host XenServer pode aceder a um ou vários dispositivos de armazenamento diferentes simultaneamente. Estes dispositivos de armazenamento podem ser dedicados apenas a um host ou partilhados entre hosts permitindo que uma imagem de disco virtual (VDI) num determinado sistema de armazenamento possa ser iniciada em qualquer host.

Armazenamento local

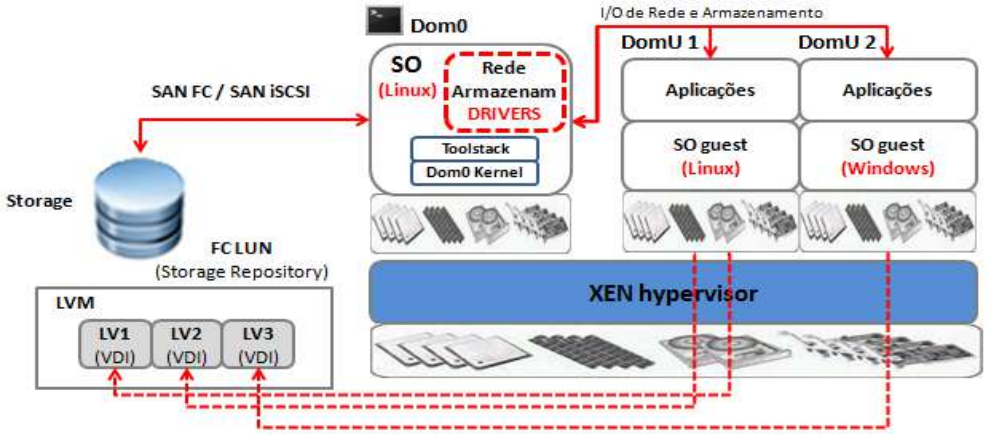
Este tipo de armazenamento é local ao host XenServer (IDE/PATA, Serial ATA (SATA), SCSI, e Serial Attached SCSI (SAS)) e neste contexto, são suportados dois tipos de armazenamento, um sistema de ficheiros EXT e LVM (Logical Volume Manager) numa LUN;

Uma LUN num SAN iSCSI ou SAN Fibre Channel

Nestes dois tipos de armazenamento (tratados diferentemente pelo XenServer), figura 54, os ficheiros são apresentados ao XenServer como ficheiros normais ou mesmo num sistema de ficheiros LVM numa LUN;

É criada uma LUN onde todos os servidores se ligam. Essa LUN é dividida em volumes lógicos onde apenas uma e uma só VM ligada a um desses volumes lógicos de cada vez.

Figura 54 - LUN num SAN iSCSI ou SAN FC



Network File System (NFS)

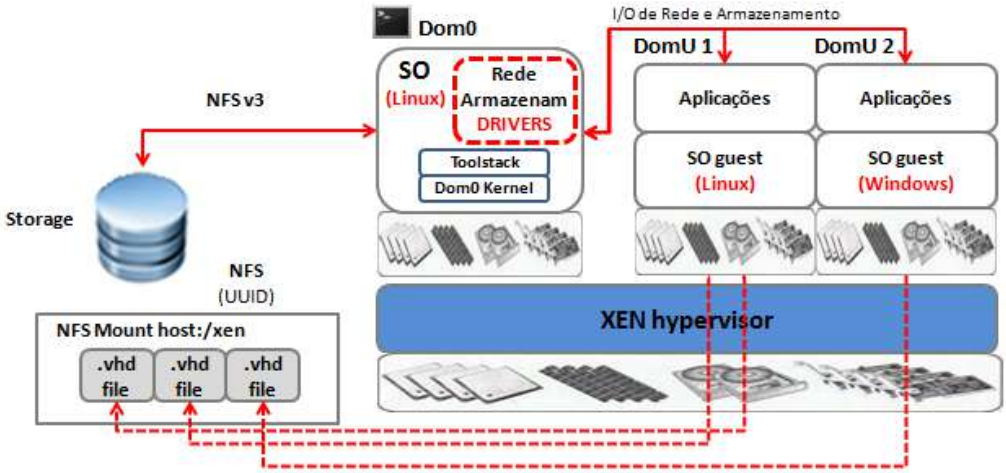
Neste tipo de armazenamento, figura 55, é criado um NFS onde todos os servidores se ligam e partilham ficheiros VDI que é o disco físico na visão de uma VM. O XenServer lida com este tipo de armazenamento como um conjunto e discos virtuais (VDI's) armazenados no formato Microsoft VHD, especialmente adequado para ambientes NSF. As imagens das VMs são armazenadas como ficheiros neste formato

[8] Clone: Um programa ou hardware que imita o produto original e pode ser executado de igual modo como se fosse original;
[9] Thin provisioning: Método de otimização da eficiência com que o espaço disponível é utilizado numa SAN.

(VHD), um disco rígido virtual. Estes ficheiros permitem que VMs sejam instanciadas em qualquer host de uma pool de recursos e possam ser feitas migrações entre servidores com o XenMotion.

XenServer suporta NFS sobre TCP/IP. O volume NFS localizado num servidor NFS é acedido, montado e usado para as necessidades convenientes de armazenamento, será criada uma diretoria de partilha com um nome que é o Identificador Único Universal (UUID) atribuído a esse armazenamento.

Figura 55 - Network File System (NFS)



Opções de armazenamento de terceiros como NetApp ou Dell EqualLogics

O armazenamento é feito por dispositivo específicos de armazenamento como NetApp (Network Appliance) ou Dell EqualLogics como fornecedores líderes de armazenamento de ficheiros em rede e sistemas de distribuição de conteúdo. A gestão do armazenamento é suportada pelo XenCenter.

Vista do armazenamento pelo SO Guest

Existem 2 tipos de VMs no XenServer, "Dom0" e "DomU", ambas VM guest's diferindo no seu propósito. O Dom0 de controlo e parte integrante de todo o sistema XenServer, dispõe de acesso privilegiados ao hardware, tem drivers dos dispositivos e é instanciada aquando do boot do sistema. Os DomU's são totalmente isolados do hardware e comunicam diretamente com o VMM ou hypervisor que controla a memória e o processador do host. A comunicação com os restantes periféricos é feita sob o controlo da VM Dom0 (domínio 0, domínio de controlo), com os drivers específicos de dispositivos como Rede e Armazenamento. Os DomU's são conhecidos como domínios sem privilégios.

Em termos de virtualização, o Xen suporta em simultâneo VM guests com para-virtualização (PV) (virtualização assistida por SO) ou virtualização assistida por hardware (HVM). Na PV, apesar da modificação do SO, não requer virtualização do processador nem do hardware uma vez que dispõe de drivers para aceder diretamente usando a capacidade total dos dispositivos. Na HVM são usadas extensões de virtualização do

processador e hardware host para virtualizar os guets mas não há modificação do SO guest, é mais lento que PV mas apresenta a valência da portabilidade. A ultima aposta da Citrix visando ótimos desempenhos foi a junção destes dois mundos, para-virtualização e virtualização assistida por hardware (PVHVM ou PV-on-HVM drivers).

Uma VM (DomU) no mundo virtual de um XenServer vê o seu armazenamento como um disco local (imagens VDI guardadas na sua representação Microsoft VHD). Na realidade, o XenServer alocou uma percentagem do armazenamento real e através do hypervisor e Dom0 fez toda a abstração, tornando-o disponível como um recurso de um disco para essas VMs. O SO guest do Dom0 é Linux e neste SO os discos físicos são representados como ficheiros desses dispositivos (/dev/sda). Os ficheiros dos dispositivos de armazenamento podem representar discos conectados localmente (armazenamento local) ou um número de unidade lógica (LUN) numa SAN, sendo a gestão feita por LVM e ainda a opção NFS.

Anexo I

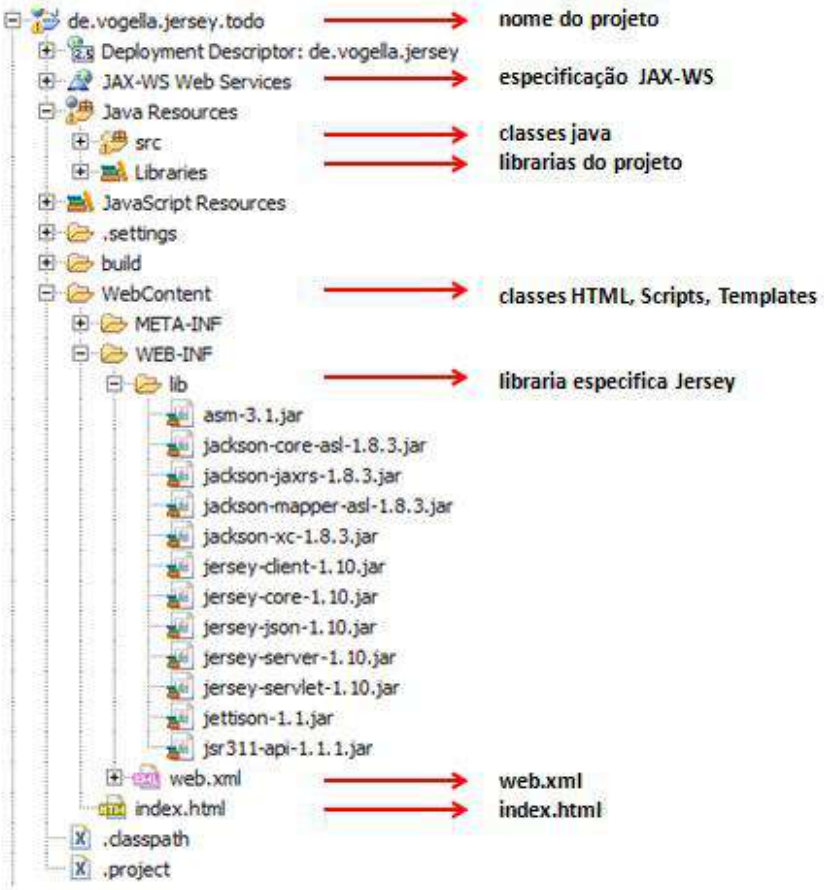
Uma exemplificação: "Dynamic Web Project"

I.1 Criação de um projecto Web Service

Um projecto Web Service, figura 56, pode ser criado com o IDE Eclipse IDE for Java EE Developers. Configurado todo o ambiente, podemos criar um projecto Web Service mediante várias abordagens, variando nas suas configurações e especificações, naturalmente:

- ✓ *Web, Dynamic Web Project;*
- ✓ *JBoss Central, HTML5 Project;*
- ✓ *JBoss Tools, Create a Sample RESTfull Web Service;*
- ✓ *Maven, Maven Project;*
- ✓ *Web Service, Create a Sample RESTfull Web Service;*

Figura 56 - Projecto "Dynamic Web Project" no eclipse



Web.xml

Ficheiro responsável pelo registo do servlet[10] fornecido pelo Jersey. Um servlet responsável por analisar os pedidos http e selecionar a classe e método para responder ao pedido, mediante as anotações JAX-RS previamente colocadas aquando da criação do código. Web.xml define ainda o path sob o qual o Web Service será disponibilizado. URL: http://dominio:porta/nome-da-aplicacao/url-pattern/path-das-classes-rest

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee" xmlns:web="http://java.sun.com/xml/ns/javaee/web-
app_2_5.xsd" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd" id="WebApp_ID" version="2.5">

  <!-- nome da aplicação -->
  <display-name>de.vogella.jersey.todo</display-name>
  <servlet>
    <!-- suporte para aplicações Web baseado em servlet e filtro -->
    <servlet-name>Jersey REST Service</servlet-name>
    <servlet-class>com.sun.jersey.spi.container.servlet.ServletContainer</servlet-class>


    <init-param>
      <!-- define o pacote das classes de recurso java do Web Service -->
      <param-name>com.sun.jersey.config.property.packages</param-name>
      <param-value>de.vogella.jersey.todo.resources</param-value>

    </init-param>
    <load-on-startup>1</load-on-startup>
  </servlet>
  <servlet-mapping>
    <!-- "URL pattern" define parte da URL base onde o Web Service será disponibilizado -->
    <servlet-name>Jersey REST Service</servlet-name>
    <url-pattern>/rest/*</url-pattern>
  </servlet-mapping>
</web-app>
```

ex. http://localhost:8080/de.vogella.jersey.todo/rest/todos/1

- ✓ *domínio:* localhost
- ✓ *porta:* 8080
- ✓ *nome da aplicação (web.xml: display-name):* de.vogella.jersey.todo
- ✓ *URL base (web.xml: url-pattern do servlet-mapping):* rest
- ✓ *anotação @Path na classe Java:* todos
- ✓ *anotação @PathParam:* 1

[10] Servlet: Uma classe na linguagem de programação Java que processa dinamicamente pedidos e respostas, gerando dados HTML e XML para a camada de apresentação de uma aplicação Web..

Principais anotações JAX-RS

@PATH(your_path)	adiciona mais um parâmetro ao URL (URL+/your_path);
@POST	O método responde a um pedido HTTP POST.
@GET	O método responde a um pedido HTTP GET.
@PUT	O método responde a um pedido HTTP PUT.
@DELETE	O método responde a um pedido HTTP DELETE.
@Produces(MediaType. TEXT_PLAIN[,more-types])	Define o tipo MIME a ser entregue por um método @GET
@Consumes(type[,more-ypes])	Define o tipo MIME a ser consumido pelo método.
@PathParam	Usado para especificar valores da URL num parâmetro do método.

Uma outra anotação fundamental é **@XmlRootElement** para que o Jersey saiba que os atributos irão passar a nós no momento em que o XML como resposta ao cliente. De igual modo, se receber um XML no corpo de um pedido, o Jersei vai saber fazer a conversão desse XML para um objeto do tipo da classe em causa.

I.2 Exemplos de solicitações à API do MOE

Adicionar mais um processador a um Servidor

Operações sobre Servidores

Verb	URI	Description
GET	/servers	Lista servidores
GET	/servers/detail	Lista detalhes dos servidores
POST	/servers	Cria servidor
GET	/servers/id	Lista detalhes dos servidores
PUT	/servers/id	Atualiza dados dos servidores
DELETE	/servers/id	Remove servidor
…	…	…

Um flavor representa uma configuração de hardware disponível para um dado tipo de servidores, e é uma combinação única de espaço em disco, capacidade de memória, número de CPUs, etc.. Um servidor, é uma instância de uma máquina virtual e, portanto, é definido por um flavor.

"flavors": [{ "disk": 10, "id": 12, "name": "256_server", "ram": 256, “cpu”: 2, … }…]

Operações sobre Flavors

Verb	URI	Description
GET	/flavors	Lista flavors
GET	/flavors/detail	Lista detalhes dos flavors

GET	/flavors/id	Lista detalhe do flavor
…	…	…

O objectivo é alterar o número de CPUs de um servidor específico e, para tal, temos de saber o valor desse "campo" no flavor desse servidor.

Algoritmo, passo-a-passo:

1) Listar todos os servidores existentes

Esta operação obtém uma lista de servidores associados à conta de um determinado utilizador. Em cada servidor (da lista) um atributo indica o estado actual do servidor; servidores com o estado ACTIVE estão disponíveis para utilização.

Verb	URI	Description
GET	/servers	Lista todos os servidores (apenas os IDs e nomes)
GET	/servers/detail	Lista todos os servidores (todos os detalhes)

Código para respostas normais: 200, 203

Código para respostas de ERRO: serversFault (400, 500), serviceUnavailable (503), unauthorized (401), badRequest (400), overLimit (413)

Resposta da listagem dos servidores: XML (detalhes)

```
<?xml version="1.0" encoding="UTF-8"?>

<servers xmlns="http://az-1.region-a.geo-
1.compute.hpcloudsvc.com/v1.1/&
lt;tenant id>">

  <server id="1234" name="sample-server" imageId="1" flavorId="12"
status="BUILD"
  progress="60" hostId="e4d909c290d0fb1ca068ffaddf22cbd0">
    <metadata>
      <meta key="Server Label">Web Head 1</meta>
      <meta key="Image Version">2.1</meta>
    </metadata>
    <addresses>
      <public>
        <ip addr="67.23.10.132"/>
        <ip addr="67.23.10.131"/>
      </public>
      <private>
        <ip addr="10.176.42.16"/>
      </private>
    </addresses>
  </server>
  …
</servers>
```

De entre todos os servidores da lista, o utilizador pretende ALTERAR o número de CPUs do servidor (***server id="1234"***), ou seja, precisa aceder ao flavor (***flavorId="12"***).

2) Listar o flavor (***flavorId="12"***)

Esta operação obtém os detalhes de um flavor específico (id).

Verb	URI	Description
GET	/flavors/id	Lista detalhe do flavor específico

Código para respostas normais: 200, 203

Error ResponseCode(s): serversFault (400, 500), serviceUnavailable (503), unauthorized (401), badRequest (400), itemNotFound (404), overLimit (413)

Resposta com detalhes de um Flavor: XML

```
<?xml version="1.0" encoding="UTF-8"?>

<flavor xmlns="http://az-1.region-a.geo-1.compute.hpcloudsvc.com/v1.1/&
lt;tenant id>"
  id="1" name="256 MB Server" ram="256" disk="10" cpu="2" />
```

3) Alterar o número de CPUs de 2 para 3 do servidor (***server id="1234"***), flavor (***flavorId="12"***)

Esta operação permite que se atualize o numero de CPUs do flavor associado ao servidor. Facilmente se depreende que esta operação só será possível se o utilizador puder efectuar esta operação (Status Transition: ACTIVE).

Verb	URI	Description
PUT	/servers/**1234**/flavor/**12**/cpu/**3**	Atualiza dados dos servidores

Código para respostas normais: 204

Código para respostas de ERRO: serversFault (400, 500), serviceUnavailable (503), unauthorized (401), badRequest (400), badMediaType(415), itemNotFound (404), buildInProgress (409), overLimit (413)

- ✓ Status Transition: ACTIVE CONFIGURESERVER (if configureServer is true)

Pedido de Update do flavor do servidor: XML

```
<?xml version="1.0" encoding="UTF-8"?>

<server xmlns="http://az-1.region-a.geo-1.compute.hpcloudsvc.com/v1.1/&
lt;tenant id>"
server id="1234" flavorId="12" cpu="3" />
```

Anexo J

Principais ferramentas e suas configurações usadas na criação e desenvolvimento deste webservice:

- ✓ Plataforma Java
- ✓ Servidor Apache Tomcat
- ✓ Ambiente de desenvolvimento Eclipse
- ✓ JAR´s e Plugins
- ✓ Macromedia Dreamweaver
- ✓ Notepad++

J.1 Plataforma Java

Java é uma linguagem de programação orientada a objeto e considera-se diferente das outras linguagens convencionais pelo facto de ser compilada para bytecode e executado por uma máquina virtual, contrariamente à compilação para código nativo. A linguagem de programação Java é a linguagem convencional da Plataforma Java, que pode ser adquirida no site http://java.com/en/download/index.jsp.

Figura 57 - Plataforma Java

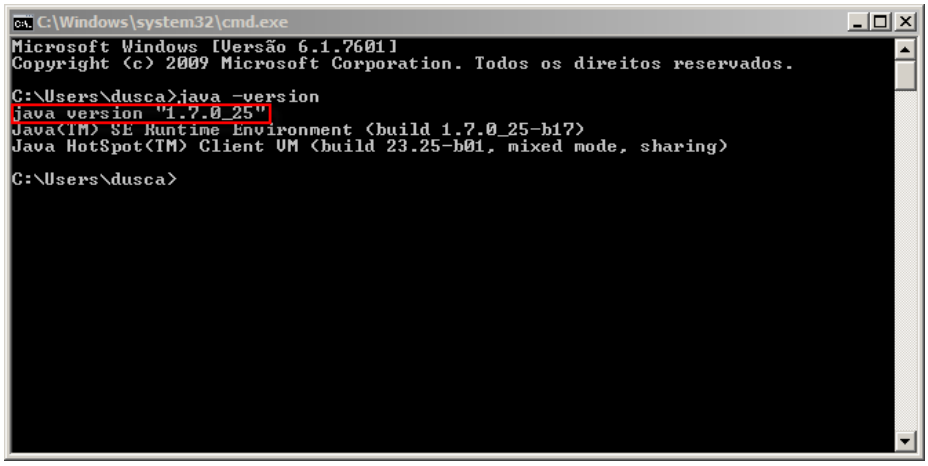
Como o RESTWEBSERVICE foi criado e executado em ambiente Windows, a explicação será feita tendo em vista este ambiente.

Assim, vamos verificar se temos a plataforma java instalada e se é a ultima versão. ‘Iniciar’ e executar ‘cmd’, abre-se uma consola de linha de comando do MS Windows, executamos o comando:

“java -version”

Este comando corresponde à chamada da máquina virtual para executar “bytecode”.

Figura 58 - Comando “java -version”



Java version “1.7.0_25 JDK ou JRE

JRE(Java Runtime Environment): Ambiente de execução Java (Java Virtual Machine). Como ambiente, será apenas possível a execução de código Java já compilado.

JDK (Java Development Kit): Kit de desenvolvimento Java, onde se inclui o compilador (javac) e o ambiente de execução Java (JVM). Neste caso, podemos compilar classes java directamente de código fonte e executa-las posteriormente no JRE, máquina virtual Java.

J.1.1 Instalação

Caso seja necessária a instalação da plataforma java, termos que fazer o download (versão 7, update 25), e efectuar a sua instalação. Por defeito, após a instalação, teremos os dois ambientes, JDK e JRE na directoria “C:\Program Files\Java”

Para testar a instalação do compilador (JDK), ‘Iniciar’ e executar ‘cmd’, abre-se uma consola de linha de comando do MS Windows, executamos o comando:

"javac"

Figura 59 - Instalação da Plataforma Java

Concluída a instalação da plataforma java, teremos que definir as variáveis de ambiente para a posterior execução do ambiente de desenvolvimento IDE (Integrated Development Environment) Eclipse e todas as dependências do Java.

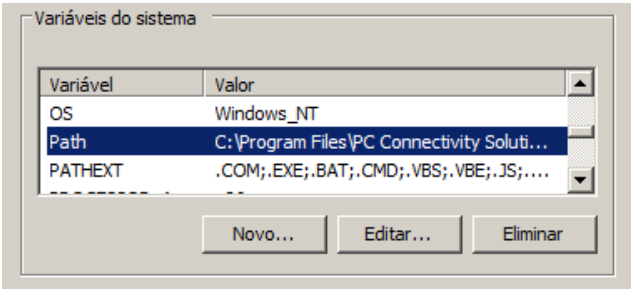
J.1.2 Variáveis de ambiente

"Iniciar", "Painel de controle", "Definições avançadas do sistema", "Variáveis de ambiente", "Variáveis de sistema":

- ✓ ***PATH*** – adicionar "directoria de instalação do JDK"\bin

  ex. C:\Program Files\Java\jdk1.7.0_25\bin

Figura 60 - Variável de sistema "Path"

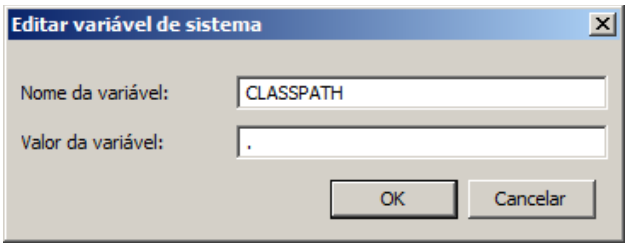


- ✓ ***CLASSPATH*** – criar esta variável

  Nome da variável "CLASSPATH"

  Valor da variável "."

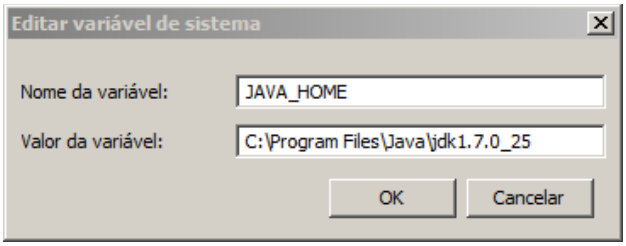
Figura 61 - Variável de sistema "CLASSPATH"



- ✓ ***JAVA_HOME*** – criar esta variável com a "directoria de instalação do JDK"\bin

  ex. C:\Program Files\Java\jdk1.7.0_25\bin

Figura 62 - Variável de sistema "JAVA_HOME"



J.1.3 Teste da instalação da plataforma java e configuração das variáveis de ambiente

Vamos verificar se as variáveis foram definidas correctamente. Para isso, com o Notpad++, vamos criar um ficheiro com o nome "Test.java" e incluir o seguinte código:

```
public class Test {

    public static void main(String[] args) throws Exception {

        System.out.println("Hello RESTWEBSERVICE");
    }

}
```

Finalmente, vamos compilar a classe na directoria onde foi criada e executá-la utilizando a máquina virtual Java. Na consola de linha de comando vamos executar os seguintes comandos:

- ✓ ***javac Test.java*** – Comando para compilar a classe java criada
- ✓ ***java Test*** - Comando para executar a classe Test

Se as variáveis de ambiente foram definidas correctamente, deveremos ter como resposta:

"Hello RESTWEBSERVICE"

J.2 Servidor Apache Tomcat

Apache Tomcat é uma implementação de software de fonte aberta das tecnologias Java Servlet e JavaServer Pages (JSP). È um servidor web HTTP em linguagem java, um servidor de aplicações JEE (Java Enterprise Edition). Tem a vantagem de incluir ferramentas para gestão e configuração que podem ser editadas directamente num ficheiro de configuração em XML.

O download deste servidor pode ser feito directamente no site da Tomcat: http://tomcat.apache.org/download-70.cgi

J.2.1 Instalação

Feito o download, basta descompactar o ZIP numa directoria, ex. C:\ onde será criado uma pasta com o nome do ZIP: C:\apache-tomcat-7.0.34. Como versão pré-configurada, disponibiliza de imediato as seguintes configurações:

- ✓ A porta de acesso ás aplicações é a 8080. Basta para isso digitar o URL seguido de ":" e a porta 8080.

  ex. http://localhost:8080/MOE

- ✓ Se não existirem redireccionamentos, é possível a listagem de directorias. Esta opção é sempre desactivada depois nas implementações.

Monitorização sempre constante, qualquer alteração dos ficheiros, não existe a necessidade de fazer um restart ao servidor.

Figura 63 - Directoria de instalação do Servidor Apache Tomcat

Nome	Tamanho	Data modificação	Tipo
bin		13-01-2013 15:28	Pasta de ficheiros
conf		14-01-2013 11:17	Pasta de ficheiros
lib		13-03-2013 17:52	Pasta de ficheiros
logs		11-04-2013 10:47	Pasta de ficheiros
temp		13-01-2013 15:28	Pasta de ficheiros
webapps		13-01-2013 15:28	Pasta de ficheiros
work		14-01-2013 11:17	Pasta de ficheiros
LICENSE	57 KB	04-12-2012 08:54	Ficheiro
NOTICE	2 KB	04-12-2012 08:54	Ficheiro
tomcat.ico	22 KB	04-12-2012 08:54	Ícone
Uninstall.exe	66 KB	13-01-2013 15:28	Aplicação

J.2.2 Execução

Para testarmos o servidor Apache Tomcat, basta corresmos o executável "Tomcat7.exe" da directoria "C:\Tomcat 7.0\bin" e aceder ao link: http://localhost:8080/

Figura 64 - Link localhost Apache Tomcat

J.3 Ambiente de desenvolvimento Eclipse

Eclipse é um IDE (Integrated Development Environment) ou ambiente integrado de desenvolvimento, é um programa com características e ferramentas de apoio ao desenvolvimento de software com a vantagem de ter a maior parte das ferramentas de desenvolvimento já integradas e prontas a usar. Uma orientação baseada em plugins que mediante incorporação, satisfazem as necessidades dos diferentes programadores.

O download deste IDE pode ser feito directamente no site: http://www.eclipse.org/downloads/. O package em questão para o desenvolvimento direcionado à web é o "Eclipse IDE for Java EE Developers" que já vem com a maior parte dos plugins destinados a este tipo de desenvolvimento.

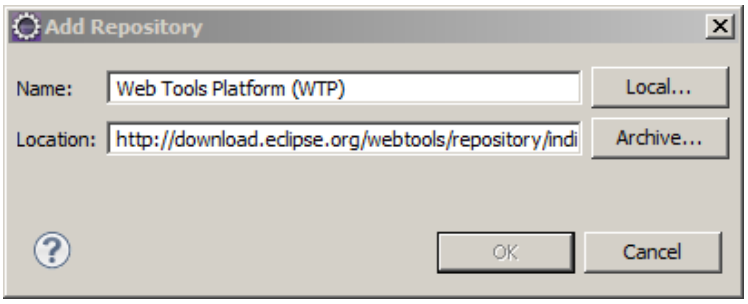
J.4 JARs e Plugins

O ambiente de desenvolvimento Eclipse, principalmente o "Eclipse IDE for Java EE Developers" já vem com a maior parte dos plugins por defeito para desenvolvimentos orientados à web mas há um indispensável, muito útil não só no acesso aos menus de configuração do projecto como também ferramentas de configuração do servidor Apache Tomcat, plugin "Web Tools Platform (WTP)".

Podemos fazer o download directo do plugin ou solicitar ao Eclipse que o faça por nós e instale directamente. Para isso, vamos ao menu "Help", "Install new Software", "add":

- ✓ *Name:* Web Tools Platform (WTP)
- ✓ *Location:* http://download.eclipse.org/webtools/repository/indigo/

Figura 65 - Add repository



Relativamente aos JARs, estes são dependents do projecto em sí e do que se pretende usar. JAR (Java Archive) é um formato de ficheiro que normalmente é usado para agregar muitos ficheiros de classes Java e metadados associados, recursos como textos e imagens, tudo num único ficheiro que depois funcionará como bibliotecas na plataforma Java.

Basta para isso colocar os JARs necessários na directoria "projectName/WebContent/WEB-INF/lib".

J.4 Macromedia Dreamweaver

O Adobe Dreamweaver, mais conhecido por Macromedia Dreamweaver é um software de desenvolvimento na Web, criada pela Macromedia.

Um software que fornece suporte para várias tecnologias web, tais como XHTML, CSS, JavaScript, Ajax, PHP, ASP, ASP.NET, JSP, ColdFusion e outras linguagens Server-side.

Toda a apresentação web da camada de apresentação do webservice foi desenvolvida e testada inicialmente neste software. Desenvolvimento de código HTML, JSP, incorporação de imagens, construção de tabelas, etc.

A.5 Notepad++

Notepad++ é um editor de texto e de código fonte aberto sob licença GPL. Tem a vantagem de suportar várias linguagens de programação como HTML, JSP, CSS, JQuery, Javascript e é optimo para alterações rápidas de código e sua visualização.

Anexo K

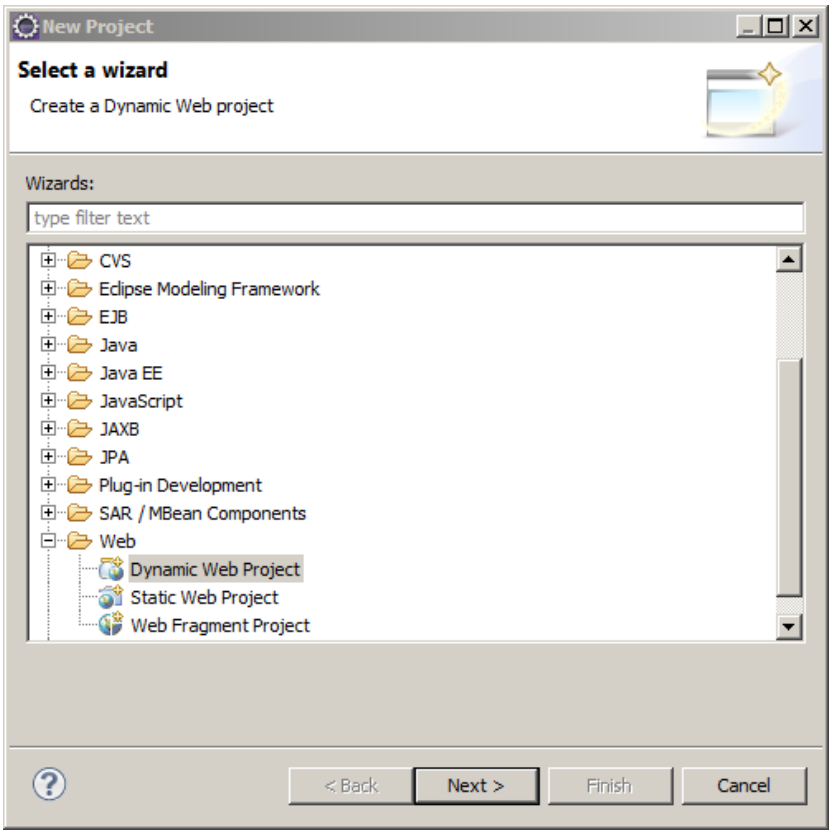
K.1 Criar um “Dynamic Web Project”

Para criar um “Dynamic Web Project” temos que iniciar o Eclipse e nesta primeira iniciação, vão ser testadas todas as variáveis de ambiente anteriormente criadas e criado um workspace onde serão guardados todos os nossos projectos.

Depois de iniciado o Eclipe, para criarmos um “Dynamic Web Project” seguimos os passos:

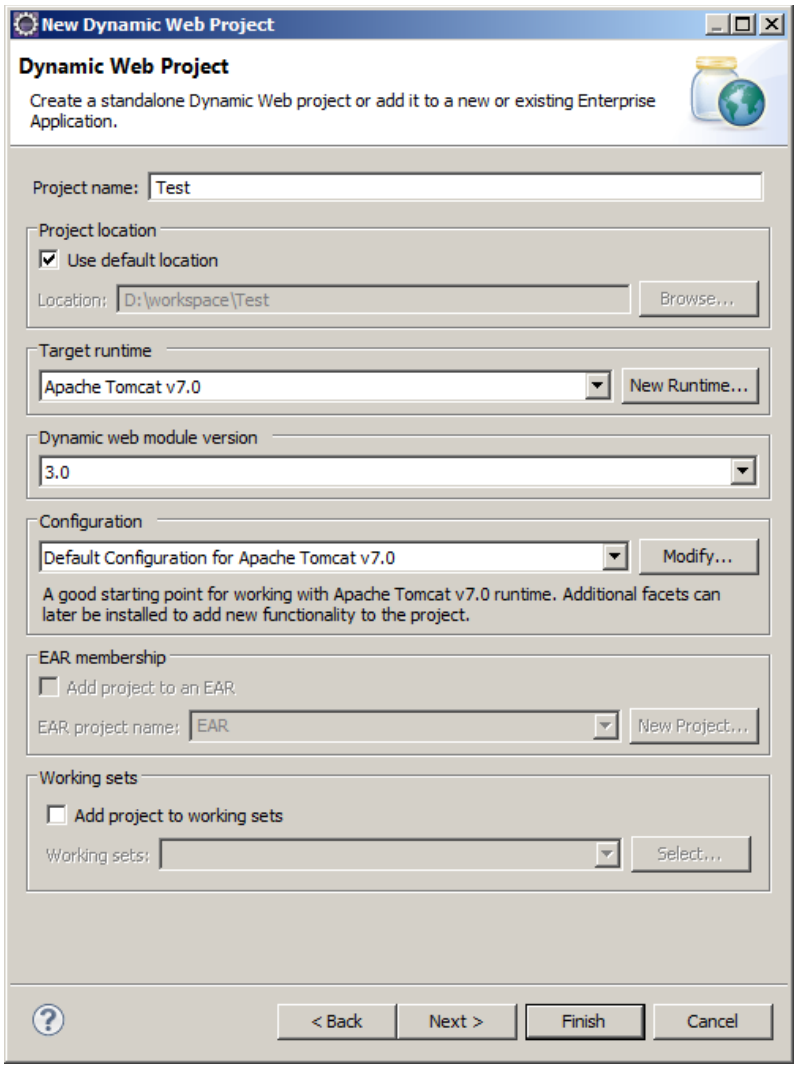
1. “File”, “New”, “Project”
2. Na próxima janela escolher “Web”, “Dynamic Web Project”

Figura 66 - Dynamic Web Project



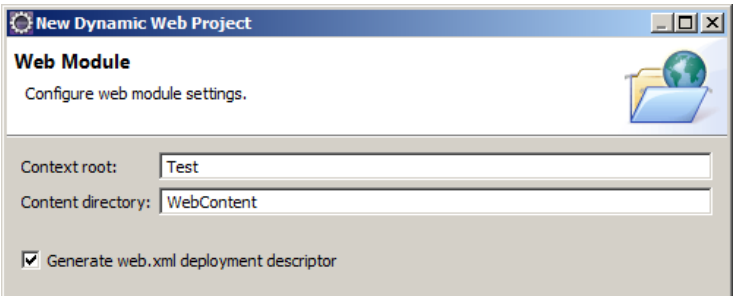
3. Clique em Next
4. Nas propriedades do projecto:
   - ✓ *Project name:* Test
   - ✓ *Target runtime:* Apache Tomcat v7.0
   - ✓ *Dynamic web module version:* 3.0
   - ✓ *Configuration:* Default Configuration for Apache Tomcat v7.0

Figura 68 - Propriedades do Dynamic Web Project



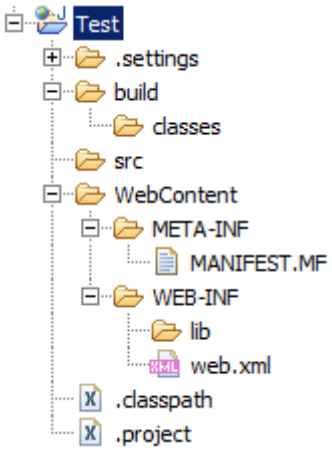
5. Clique em Next e depois outra vez em Next
6. Seleccione "Generate web.xml deployment descriptor" para a criação do web.xml onde serão incluídas todas as definições e configurações do projecto

Figura 69 - Web.xml do Dynamic Web Project



7. Clique em Next e todo o projecto é criado

Figura 70 - Directoria base do Dynamic Web Project



JavaSource (src)

Código fonte Java do projeto. Recursoso que são automaticamente compilado e os ficheiros gerados são adicionados ao diretório WEB-INF/classes.

WebContent

Localização de todos os recursos da Web, incluindo HTML, JSP, ficheiros gráficos, imagens, etc. Os ficheiros ficam disponíveis apenas nesta directoria ou subdirectorias quando o servidor é executado.

META-INF

Esta directoria contém o arquivo MANIFEST.MF, que é usado para mapear o caminho das classes para os ficheiros JAR dependentes que existem no projecto. Quando um JAR é adicionado à biblioteca, esta entrada é adicionada automaticamente neste ficheiro.

WEB-INF

Directoria que contém os recursos da Web de suporte para à aplicação, incluindo o ficheiro web.xml e as classes e lib.

/ Classes

Directoria das classes compiladas, é a directoria de saída do compilador Java. As classes ".class" são colocados neste directoria automaticamente quando o compilador Java compila os ficheiros de origem Java que estão no directoria de recursos Java (src).

/ Lib

Os ficheiros JAR de apoio a toda a aplicação. As classes existentes nos ficheiros jar ficam disponíveis para uso na aplicação.

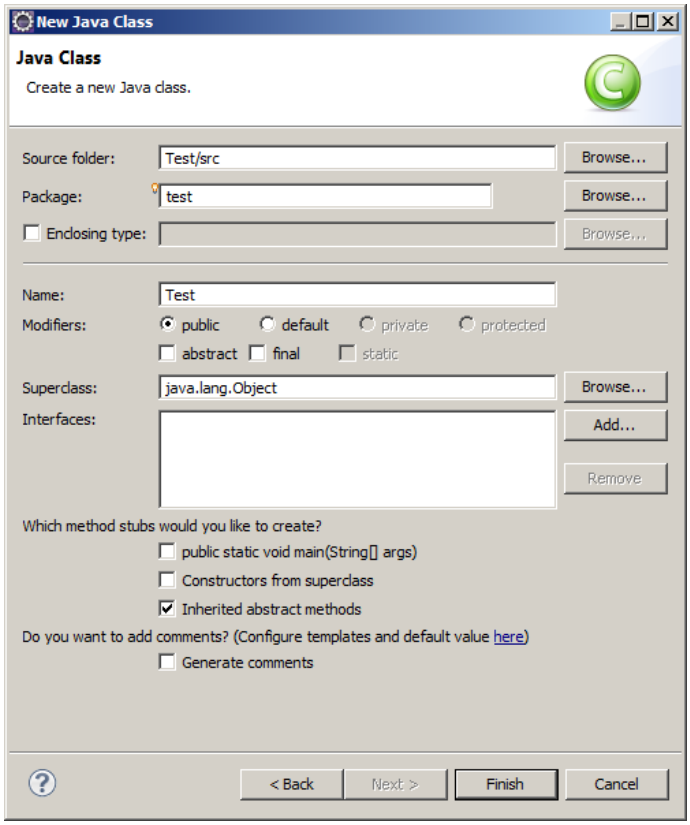
Web Deployment Descriptor

O descritor de implementação de toda a aplicação, o ficheiro web.xml.

K.2 Executar um “Dynamic Web Project”

Para testarmos o nosso “Dynamic Web Project” começamos por criar uma classe na directoria “src”. “File”, “New”, “Other”, “Class”

Figura 71 - Criar uma classe



Criamos uma classe com o package test e o nome Test e incluímos o seguinte código:

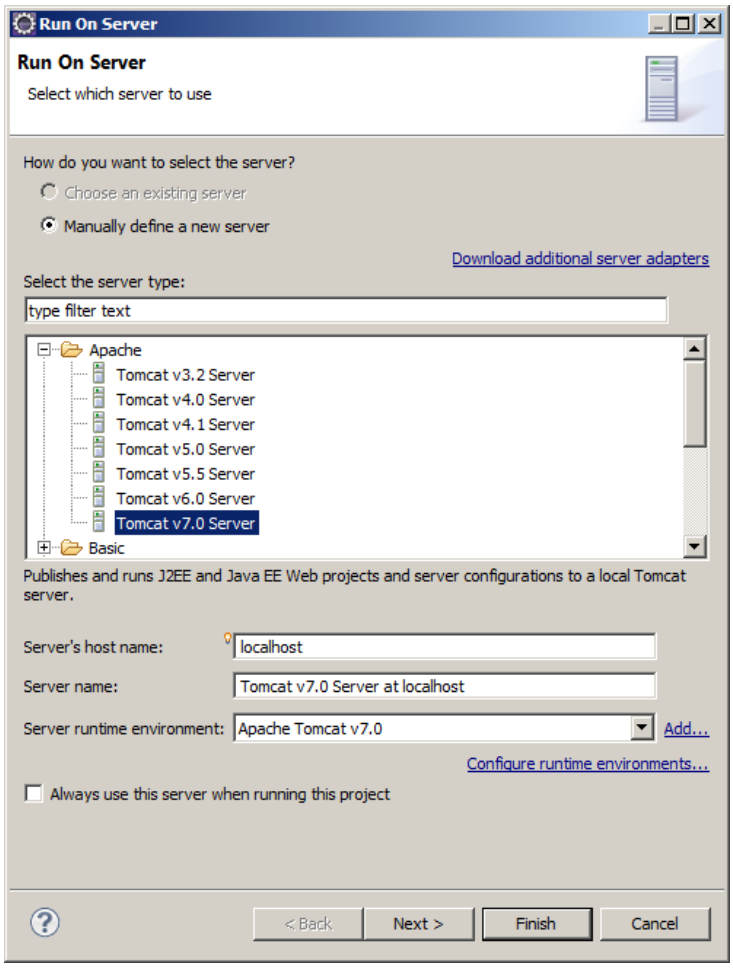
```
package test;

public class Test {

    public static void main(String[] args) throws Exception {

        System.out.println("Hello RESTWEBSERVICE");
    }
}
```

Criamos igualmente um ficheiro HTML na directoria WebContent, “File”, “New”, “file”, com o nome “test.html” e com o seguinte código:

```
<html>
    <head>
        <title>Dynamic Web Project HTML page</title>
    </head>
    <body>
        <h2> Hello RESTWEBSERVICE </h2>
    </body>
</html>
```

Executamos o projecto, no menu, clicamos em “Run”, “Run”, escolhemos “Run on Server” e a primeira vez que executarmos o projecto, temos que fazer a ligação ao servidor Apache Tomcat, “Apache”, “Tomcat v7.0 Server” e “Finish”

Figura 72 - Servidor Apache Tomcat dentro do Eclipse



O projecto está pronto a testar no link http://localhost:8080/Test/test.html

Figura 73 - Execução no Browser

Anexo L

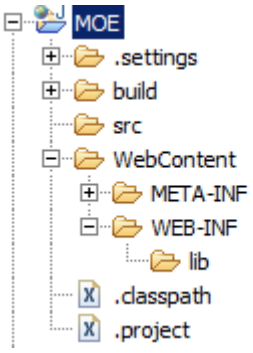
A implementação RESTWEBSERVICE - MOE é um webservice desenvolvido numa arquitectura orientada a Web (Web-Oriented Architecture - WOA), disponibilizando os seus recursos na Web através do protocolo REST (Representational State Transfer) para troca de informação. Vantagens como a simplicidade de interação entre sistemas heterogéneos, a facilidade de comunicação entre aplicações existentes em múltiplas plataformas e a independência de plataformas e linguagens de programação, são evidentes.

L.1 Importação do "Dynamic Web Project" MOE

A importação o projecto "MOE" (Matrix Operating Environment) para o ambiente de desenvolvimento Eclipse carece da criação de um "Dynamic Web Project" base no Eclipse.

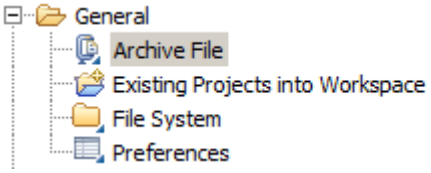
Assim, para a criação de um "Dynamic Web Project", seguir os passos descritos na alínea *K.1 Criar um "Dynamic Web Project"* do *Anexo K* alterando apenas o nome para "MOE". Não é necessário criar nenhuma classe java nem ficheiro HTML.

Figura 74 - Projecto base MOE



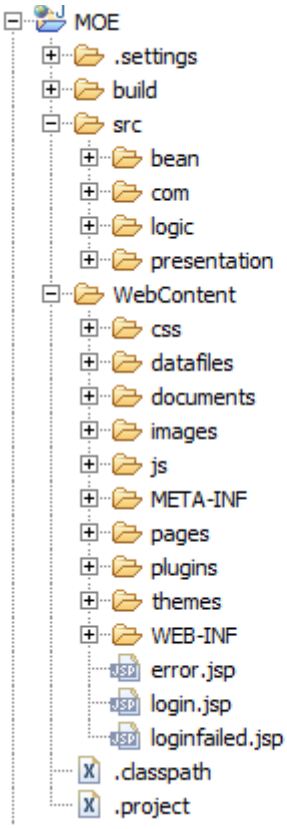
Depois do projecto MOE criado, basta importar o projecto real para o Eclipse: "File", "Import", "Archive File"

Figura 75 - Importação do projecto real MOE



Seleccionar o projecto a importar. O projecto deverá ficar com uma organização de directorias como na imagem seguinte:

Figura 76 - Projecto real MOE



L.2 Configuração inicial do MOE

Para que a execução do MOE seja feita sem erros e todas as ligações e credenciações sejam efectuadas, temos que criar pastas e ficheiros de leitura inicial do MOE. Assim, vamos criar uma directoria em C:\ com o nome "restconfig" (*C:\restconfig*) com 4 ficheiros:

- ✓ ***adminUsersList.txt***

Ficheiro com todos os administradores do MOE. Esta lista deverá ser igual à lista dos administradores do HPMOE. Um administrador por linha "domínio\admin" (ex. "hpptlab\paulo.pires")

Figura 77 - Ficheiro “adminUsersList.txt”

```
adminUsersList.txt
1   hpptlab\paulo.pires
```

✓ ***hp-io-v5.wsdl***

Ficheiro WSDL (Web Service Description Language) a definição das operações de IO disponíveis.

https:// <cms-ip-address>: 51443/hpio/controller/soap/v4 wsdl?

Neste ficheiro basta alterar o IP do servidor HPMOE, linha 5987:

<soap:address location="https://**16.23.187.6**:51443/hpio/controller/soap/v5"/>

Figura 78 - Ficheiro “hp-io-v5.wsdl”

```
hp-io-v5.wsdl
1   <wsdl:definitions xmlns:ns1="http://cxf.apache.org/bindings/xformat"
2       xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://v5.soap.io.hp.com/"
3       xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
4       name="IO" targetNamespace="http://v5.soap.io.hp.com/">
5       <wsdl:types>
6           <xs:schema xmlns:tns="http://v5.soap.io.hp.com/" xmlns:xs="http://www.w3.org/2001/XMLSchem
7               attributeFormDefault="unqualified" elementFormDefault="unqualified"
8               targetNamespace="http://v5.soap.io.hp.com/">
9               <xs:complexType name="LoadbalancerInfo">
10                  <xs:sequence>
```

Figura 79 - WSDL do HPMOE

```
https://16.23.187.6:51443/hpio/controller/soap/v5
ols » Outros marcadores
This XML file does not appear to have any style information associated with it. The document tree is shown below.
▼<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  ▼<soap:Body>
    ▼<soap:Fault>
       <faultcode>soap:Server</faultcode>
       <faultstring>Fault occurred while processing.</faultstring>
     </soap:Fault>
   </soap:Body>
 </soap:Envelope>
```

✓ ***login.config***

Ficheiro de apoio à credenciação (login) no MOE.

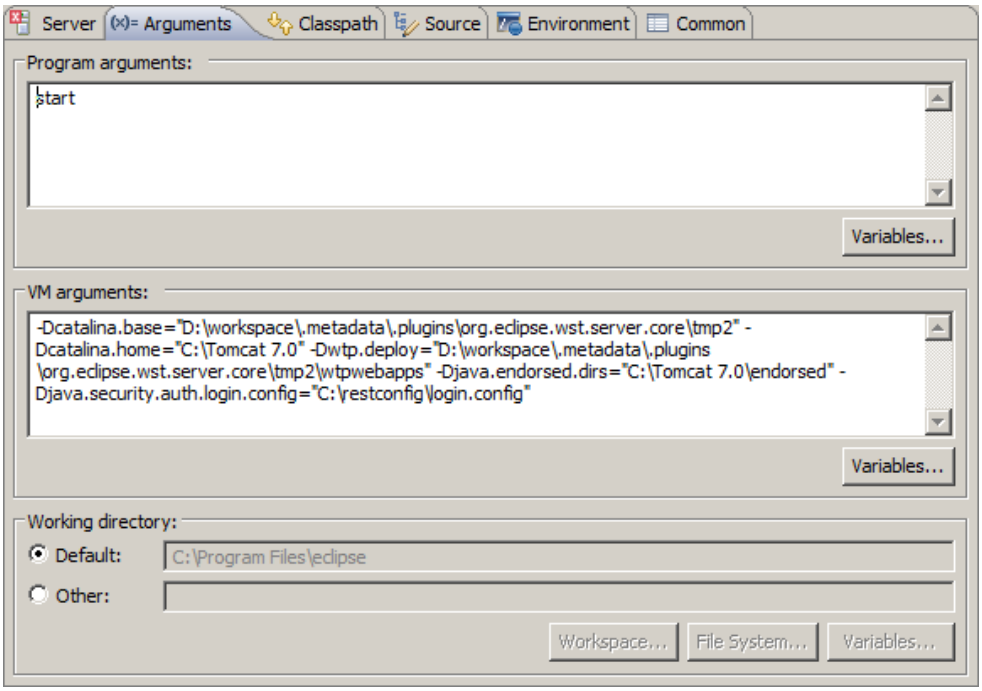
No âmbito da segurança em aplicações JavaEE, o servidor Apache Tomcat fornece uma implementação Realm JAAS (*Java Authentication and Authorization Service*) para que os programadores possam implementar módulos de login JAAS e integrá-los facilmente com as suas aplicações. JAAS permite assim autenticações em modo *Basic* e *Form*, usando para isso os denominados realms: File e JDBC (Java Database Connectivity) que, neste caso é em *runtime*.

Figura 80 - Ficheiro “login.config”

```
login.config
1  BytesLoungeLogin {
2      com.login.BytesLoungeLoginModule required debug=true;
3  };
```

A ligação ao ficheiro “login.config” é feita no menu do Eclipse, “Run”, “Run Configurations”, “Tomcat 7.0”, “Arguments”, “VM arguments”, adicionar a linha: -Djava.security.auth.login.config="C:\restconfig\login.config"

Figura 81 - Ficheiro “login.config”



✓ ***vmwareAccess.txt***

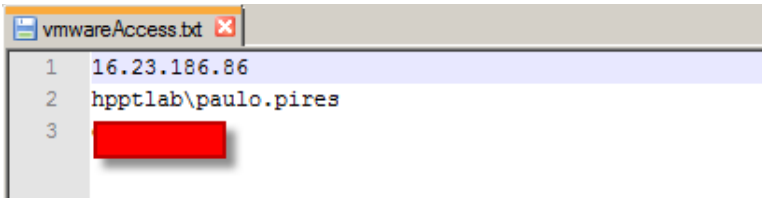
Ficheiro com dados de acesso à VMware para acesso à consola das VMs dos serviços criados.

Primeira linha: IP da VM onde está o VCenter da VMware

Segunda linha: Username do Administrador

Terceira linha: Password do Administrador

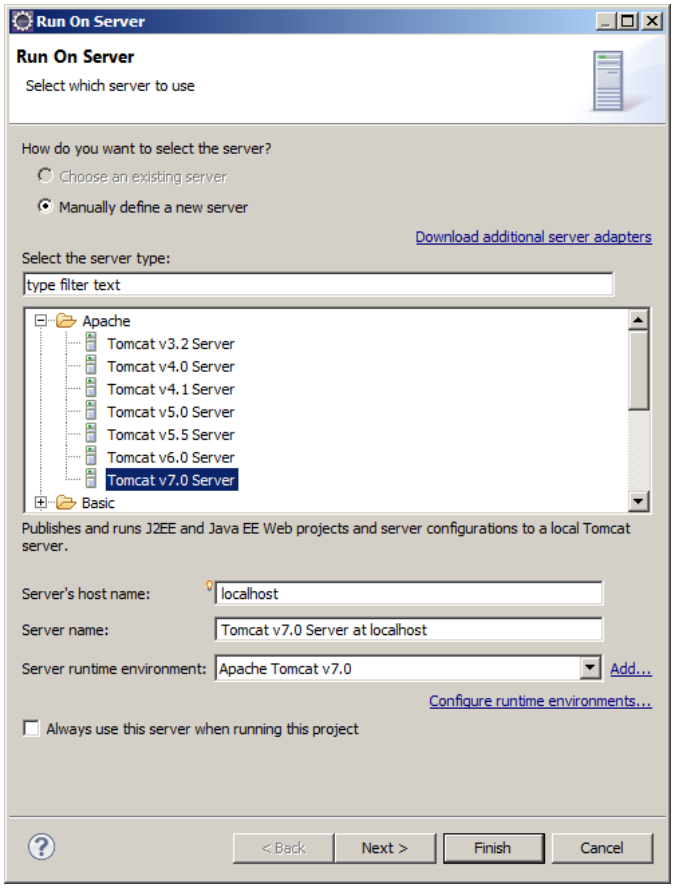
Figura 82 - Ficheiro “vmwareAcess.txt”



L.3 Execução do MOE

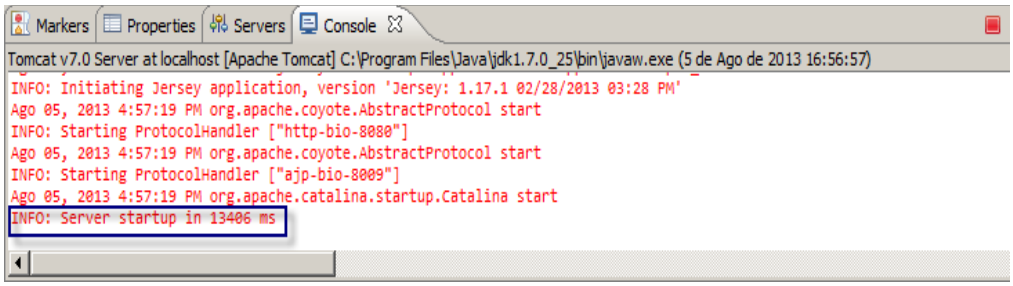
O MOE pode finalmente ser executado. Para isso, no menu, clicamos em “Run”, “Run”, escolhemos “Run on Server” e a primeira vez que executarmos o projecto, temos que fazer a ligação ao servidor Apache Tomcat, “Apache”, “Tomcat v7.0 Server” e “Finish”

Figura 83 - Servidor Apache Tomcat dentro do Eclipse



Aguardar que o projecto seja construído e compilado. Podemos acompanhar a evolução do estado directamente na consola do Eclipse.

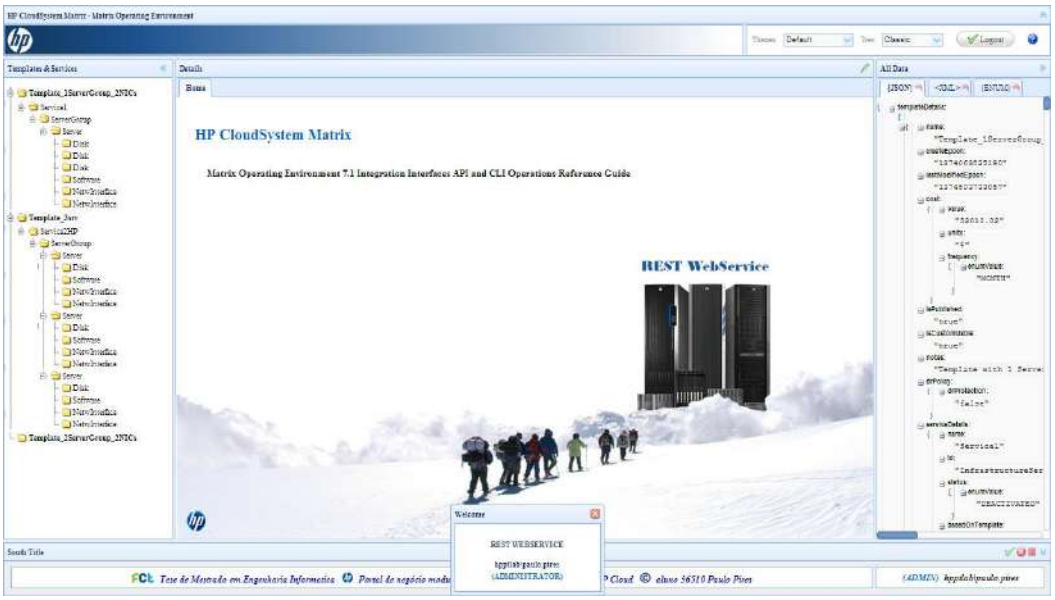
Figura 84 - Compilação do projecto MOE



Finalmente o projecto está disponível em: http://localhost:8080/MOE

Figura 85 - Login page MOE

Figura 86 - Data page MOE



Anexo M

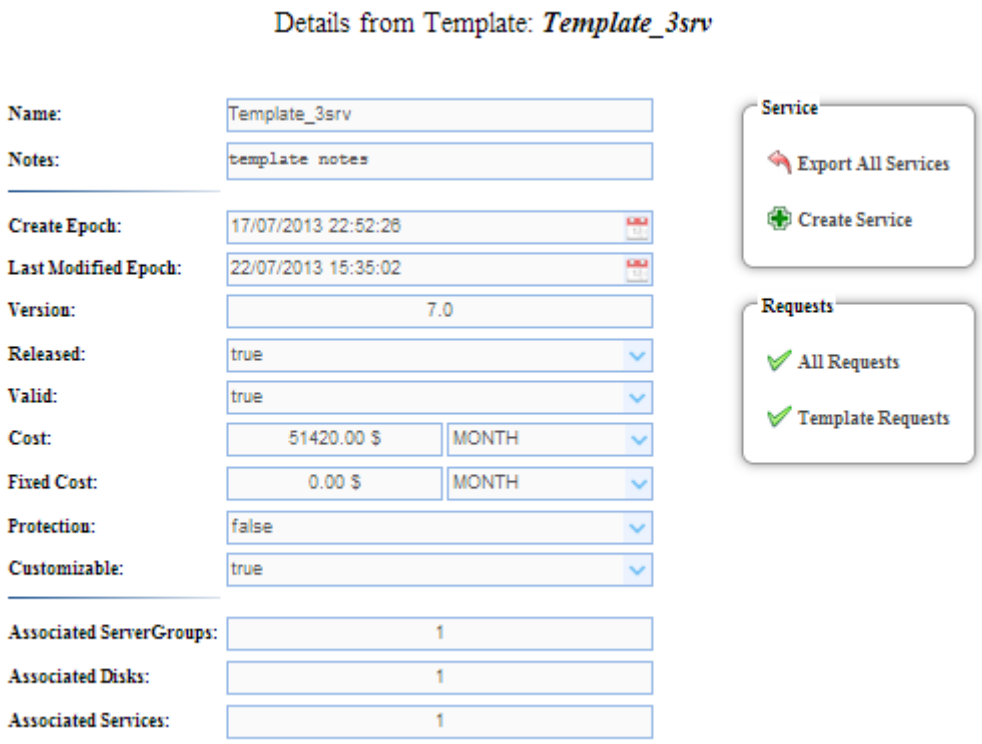
A criação de um serviço de um determinado utilizador tem por base as definições/configurações existentes num Template previamente definido e listado num Catálogo de Serviços. Configurações como detalhes dos servidores, memória, processadores, discos, custos, interfaces, limites máximos e mínimos de futuros updates ao serviço, etc, estão previamente descritos e implementados sob a forma de Templates nos quais o utilizador pode navegar, verificar as características de cada um e escolher um que melhor se adapte as suas necessidades.

Assim, o utilizador navega nos Templates existentes no Catálogo de Serviços do menu lateral esquerdo e seleciona o Template. No painel central do webservice, painel "Detalhes", são abertas as tabs dos detalhes do Template selecionado.

M.1 Detalhes do Template

Nas imagens seguintes podemos verificar os detalhes do Template selecionado. Nos detalhes gerais do Template obtemos informações como o nome do Template, a data de criação e modificação, versão, custo geral, se é ou não costumizável e o numero de ServerGroups associados, assim como discos e serviços que já foram criados tendo por base este template.

Figura 87 - Detalhes gerais do Template "Template_3srv"



Todos os detalhes do Template ficam disponíveis ao utilizador. Ao nível do grupo de servidores, informações como nome, quantidade mínima de servidores e que será a base do serviço criado e quantidade máxima de servidores, sinal que o serviço poderá crescer em termos de quantidade de servidores. Tipo de cluster e o numero de servidores lógicos existentes no grupo.

Figura 88 - Detalhes do ServerGroup "SvrGrp1" do Template "Template_3srv"

Details from ServerGroup: ***SvrGrp1***

ServerGroup Name:	SvrGrp1	
Min Server Count:	1	
Max Server Count:	3	
Active Server Count:	33%	1
Ordinal:	0	
Cluster Type:	UNSPECIFIED	
Associated Logical Servers:	1	

Relativamente aos servidores, o utilizador obtém informações como nome, hostname, tipo de virtualização, memória usada e quantidade máxima de memória, processadores usados e quantidade máxima de processadores, se é ou não virtual e se está ou não em cloud.

Figura 89 - Detalhes do Servidor "SvrGrp1-1"

Details from Server: ***SvrGrp1-1***

Server Name:	SvrGrp1-1	
Hostname:	#hostr01	
Virtualization Type:	ESX	
Max Memory Size:	1024	MB
Memory Size:	50%	512
Max Processor Count:	3	
Processor Count:	33%	1
Processor Speed MHz:	0	
Virtual:	YES	
InCloud:	UNSPECIFIED	

Informações detalhadas sobre os custos, em detalhes, ficam disponíveis ao utilizador. Informações sobre custo total, custo de base por cada servidor, custo por processador e memória.

Figura 90 - Detalhes do Servidor “SvrGrp1-1”

Details Cost from Server: ***SvrGrp1-1***

Server Name:	SvrGrp1-1	
Total Cost per Server:	51400.00 $	MONTH
Base Cost per Server:	100.00 $	MONTH
Cost per Processor:	100.00 $	MONTH
Cost per Memory:	100.00 $	MONTH

Os discos também apresentam informação como nome, tipo de storage, tamanho, raid level, se é ou não bootable, shareable e custo do disco.

Figura 91 - Detalhes do Disco “Disk1-1” do Servidor “SvrGrp1-1”

Details Disk from Server: ***SvrGrp1-1***

Name:	Disk1-1	
Storage Type:	VMFS	
Size:	24577	MB
Raid Level:	UNSPECIFIED	
Bootable:	YES	
Shareable:	NO	
Cost:	0.00 $	MONTH

Cada servidor tem um software pré-instalado ou instalado posteriormente pelo utilizador. Informações como nome, tipo, OS e virtualização estão disponíveis.

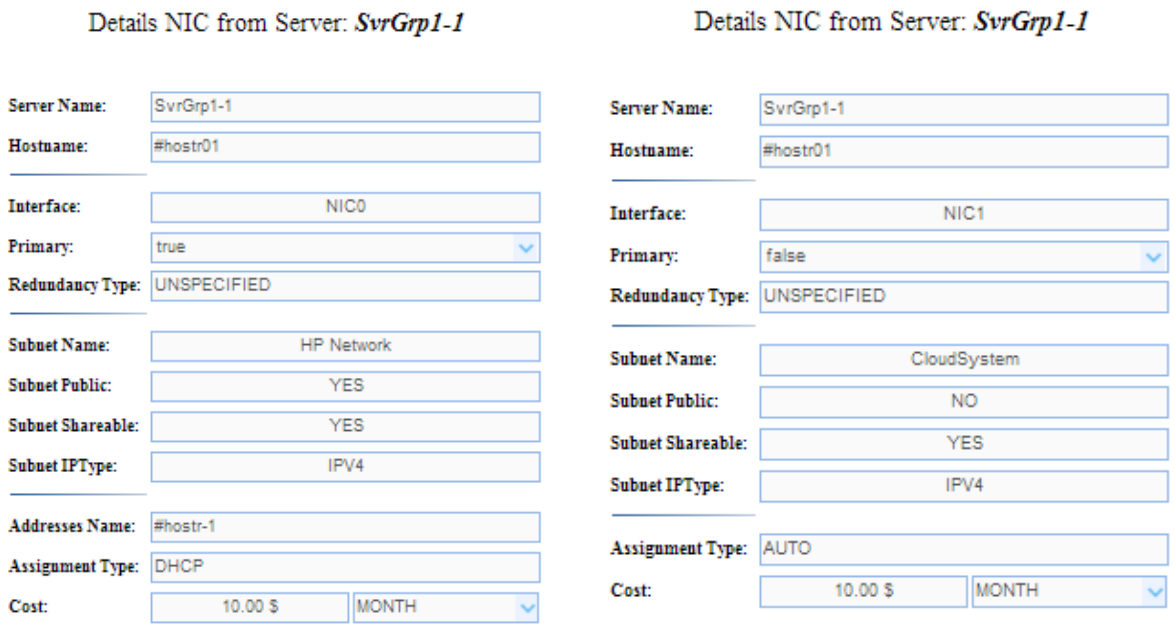
Figura 92 - Detalhes do Software “Windows 7” do Servidor “SvrGrp1-1”

Details Software from Server: ***SvrGrp1-1***

Name:	Windows 7 (CIFM02)
Notes:	Template UUID:42310cc0-afe7-abb4-9af5-2e02972c5eea
Type:	VM_TEMPLATE
OS Type:	WINDOWS
Virtualization Type:	ESX

Finalmente, detalhes sobre a/s interfaces. Neste caso duas interfaces, uma na rede HP (HP Network) e outra rede local (CloudSystem). Estão disponíveis informações como nome da interface, se é ou não primária, redundância, subnet, publica, shareable, tipo de IP, endereço, DHCP activo ou não e custo final.

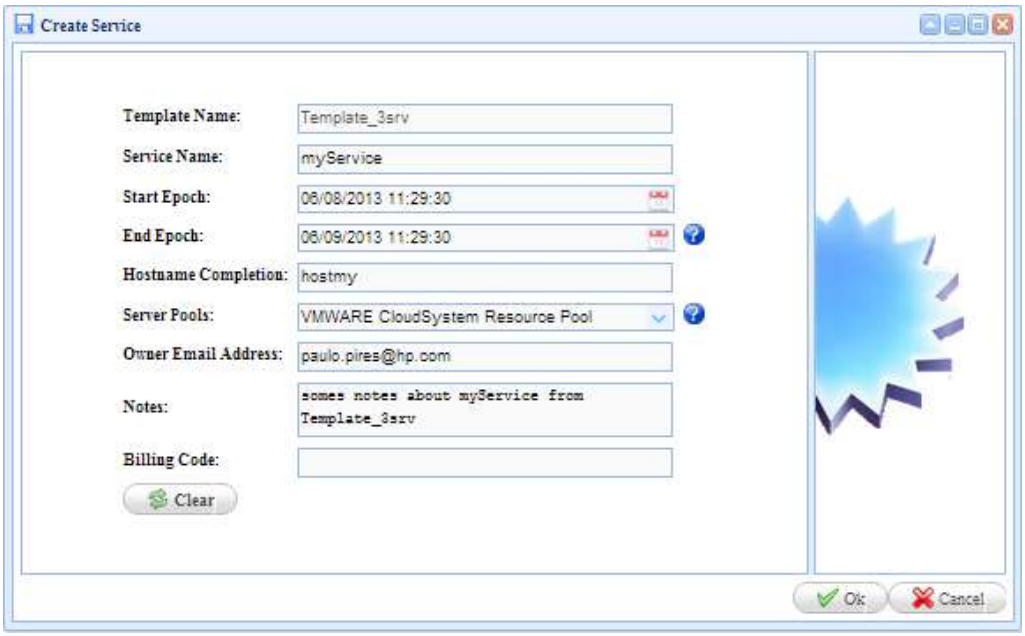
Figura 93 - Detalhes da/s Interfaces “NIC0” e “NIC1” do Servidor “SvrGrp1-1”



M.2 Criação de um Serviço

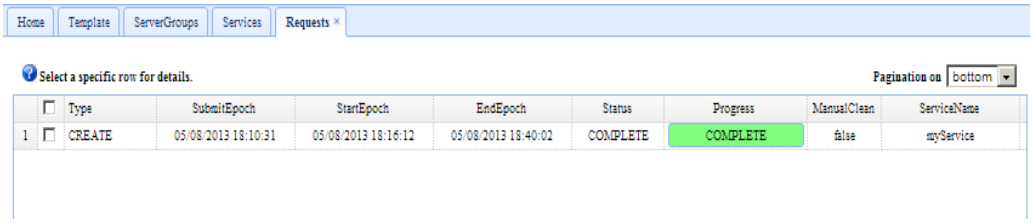
Selecionado o Template que melhor se enquadra nas necessidades do utilizador, clicar na opção Create Service , preencher todos os dados obrigatórios como nome do serviço, hostname, tempo atribuído ao serviço (acarreta custos) e email do utilizador. As opções “Notes” e “Billing code” são opcionais.

Figura 94 - Criação do serviço “myService”



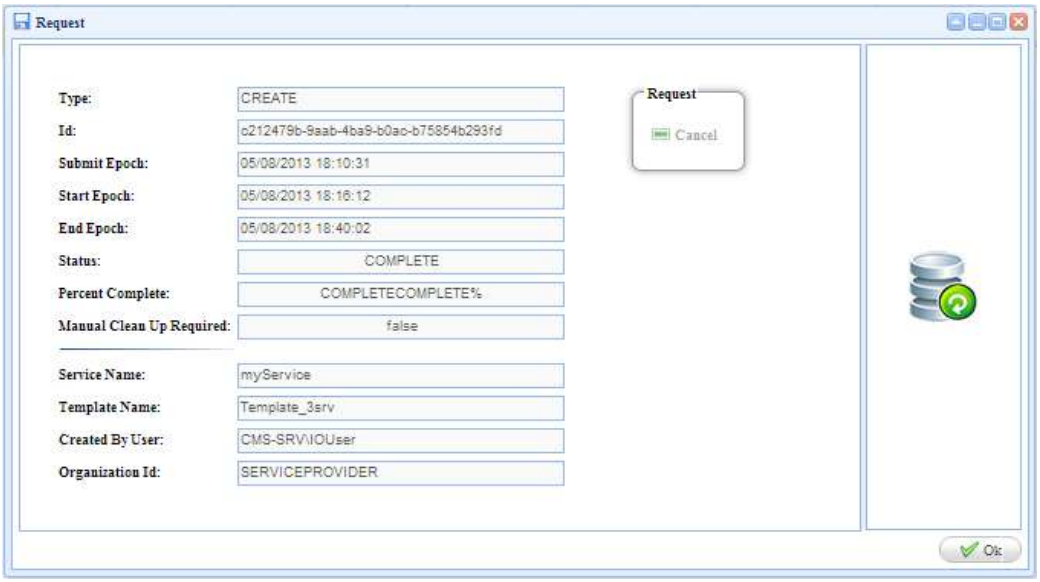
Após a solicitação para a criação do serviço, o utilizador é encaminhado para uma lista de pedidos existentes para aquele utilizador e serviço em específico. Pedidos que poderão ser cancelados (figura 95) mediante decisão do utilizador.

Figura 95 - Lista de pedidos do utilizador



A salientar que existem pedidos que carecem da aprovação do administrador por isso o pedido poderá ficar parado numa determinada percentagem (%) a espera dessa aprovação. Se o pedido não necessitar de aprovação, continuará a sua execução até estar completo (100%) e automaticamente será adicionado ao menu lateral esquerdo do utilizador.

Figura 96 - Eventual cancelamento de um pedido



Finalmente, o serviço é criado e todas as opções ficarão agora disponíveis ao utilizador como edição de memória e processadores, upgrade de servidores, exportação de dados em xml, visualização nesse formato, etc.

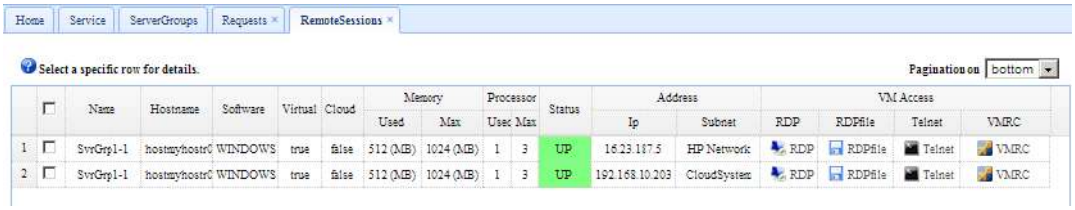
M.3 Acesso Remoto

Acesso remoto à maquina virtual (VM), como objectivo de toda a implementação está igualmente disponível para os servidores cujo status = "UP". Essa listagem poderá ser efectuada na tab "Serviços" em List Remote Session com quatro tipo de ligações:

- ✓ RDP — RDP (Remote Desktop Protocol) (figura 100)
- ✓ RDPfile — Dowload do ficheiro RDP para posterior ligação
- ✓ Telnet — Telnet (figura 101)
- ✓ VMRC — VMRC (VMware Remote Console) (figura 102)

Figura 97 - Serviço "myService" criado

Figura 98 - Listagem dos acessos remotos



Os acessos remotes poderão ser feitos através desta lista ou directamente nos servidores ou interfaces

Figura 99 - Acesso remoto na tab “Server”

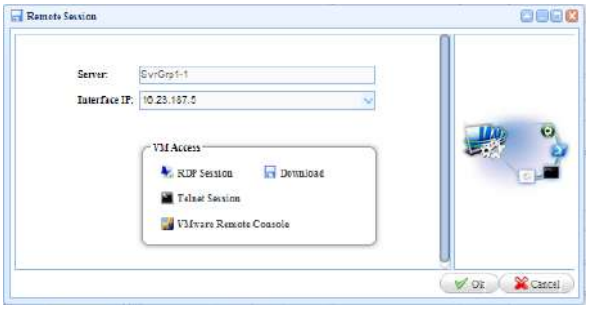


Figura 100 - Acesso remoto por RDP

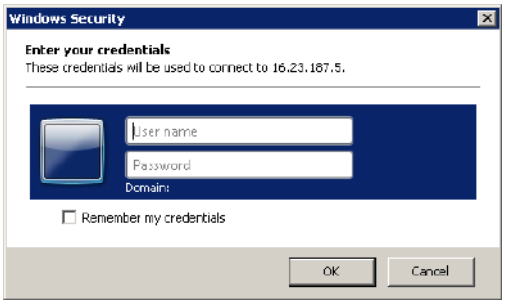


Figura 101 - Acesso remoto por Telnet

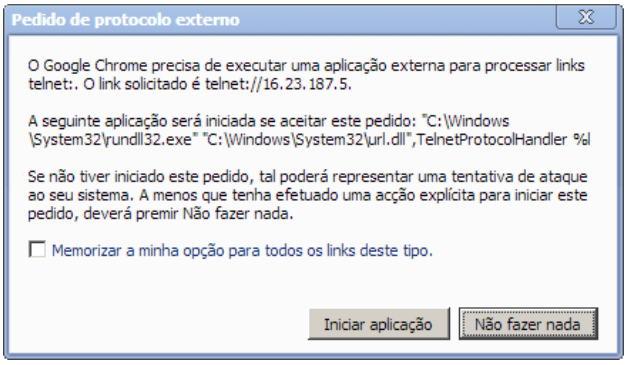
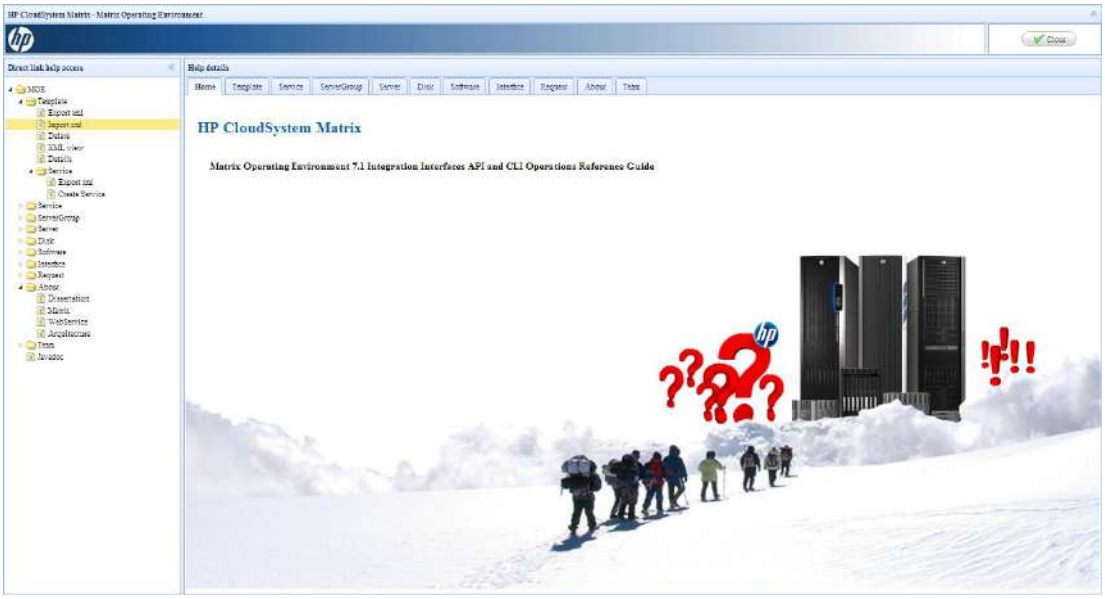


Figura 102 - Acesso remoto por VMRC

Anexo N

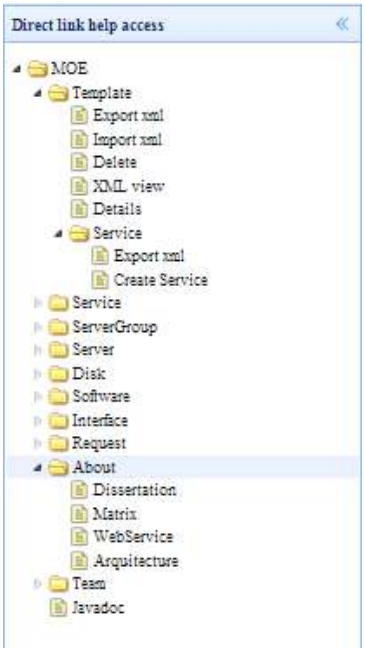
A informação disponibilizada é vasta e detalhada sobre os recursos disponibilizados e a sua manipulação pode criar alguma dúvida existencial. Uma página "MOE Help" de ajuda é sempre bem vinda e útil, bastando para isso clicar no ícone que o ajudará a compreender melhor tido o wevservice e sua distribuição e organização.

Figura 103 - Menu lateral esquerdo "MOE Help"



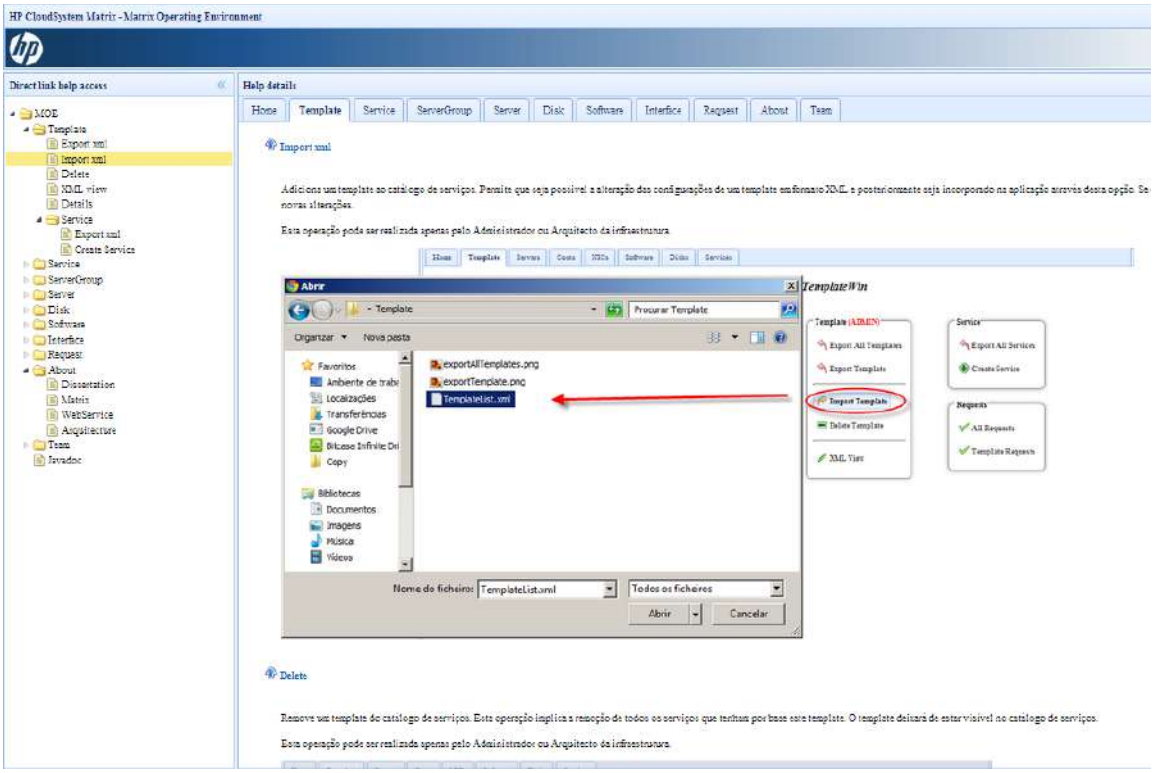
O menu lateral esquerdo, à semelhança da página principal do webservice, expelha a hierarquia da organização de toda a informação. O utilizador apenas tem que clicar na opção que deseja maior esclarecimento.

Figura 104 - Menu lateral esquerdo "MOE Help"



Toda a informação de ajuda está imediatamente disponível com as Tabs respectivas já abertas e prontas a consultar. O menu lateral esquerdo serve apenas para selecionar a Tab e dentro dessa Tab, selecionar imediatamente a informação pretendida. Se o utilizador tiver dúvidas sobre a a importação de um Template, basta abrir a Tab Template e scroll até à informação pretendida ou sentão, mais fácil, clicar apenas em "Import xml" dentro da pasta Template no menu lateral esquerdo e, com um simples clique, visualiza imediatamente a informação, nem necessidade de scroll.

Figura 105 - "Import xml" help

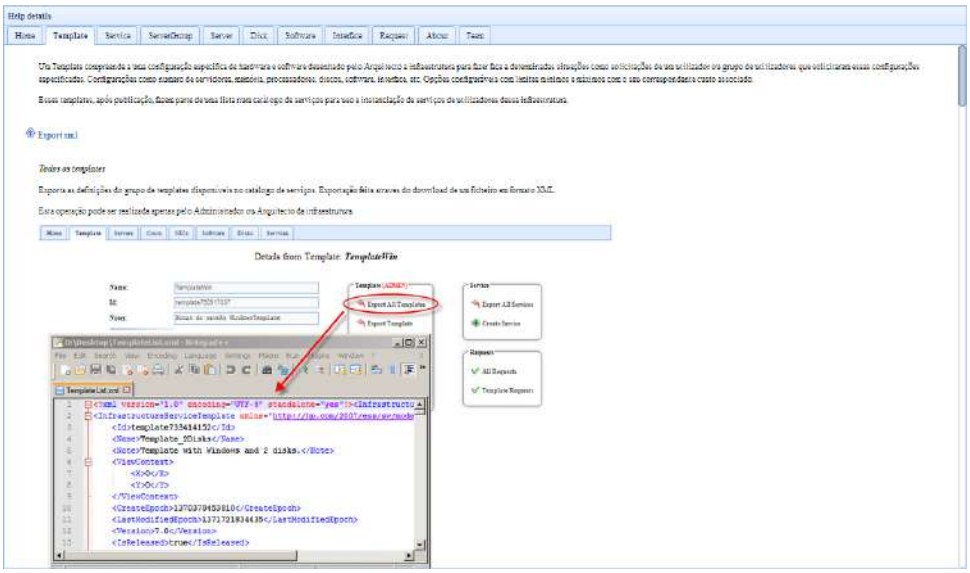


N.1 Template

Um Template compreende a uma configuração especifica de hardware e software desenhado pelo Arquitecto a infra-estrutura para fazer face a determinadas situações como solicitações de um utilizador ou grupo de utilizadores que solicitaram essas configurações especificadas. Configurações como número de servidores, memória, processadores, discos, software, interface, etc. Opções configuráveis com limites minimos e máximos com o seu correspondente custo associado.

Esses templates, após publicação, fazem parte de uma lista num catálogo de serviços para uso e instanciação de serviços de utilizadores dessa infra-estrutura.

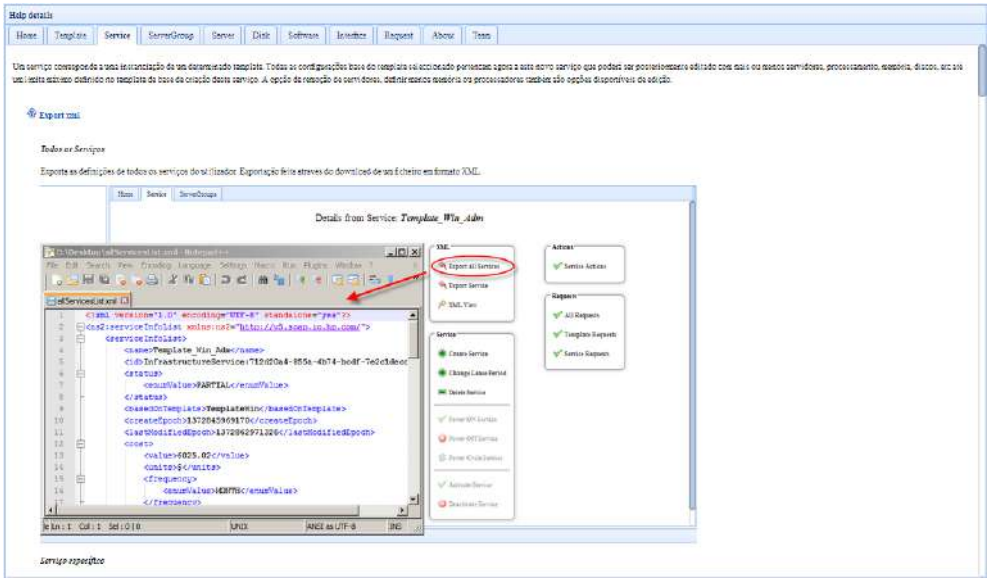
Figura 106 - Template info Tab



N.2 Service

Um serviço corresponde a uma instanciação de um determinado template. Todas as configurações base do template seleccionado pertencem agora a este novo serviço que poderá ser posteriormente editado com mais ou menos servidores, processamento, memória, discos, etc até um limite máximo definido no template de base de criação deste serviço. A opção de remoção de servidores, definir menos memória ou processadores também são opções disponíveis de edição.

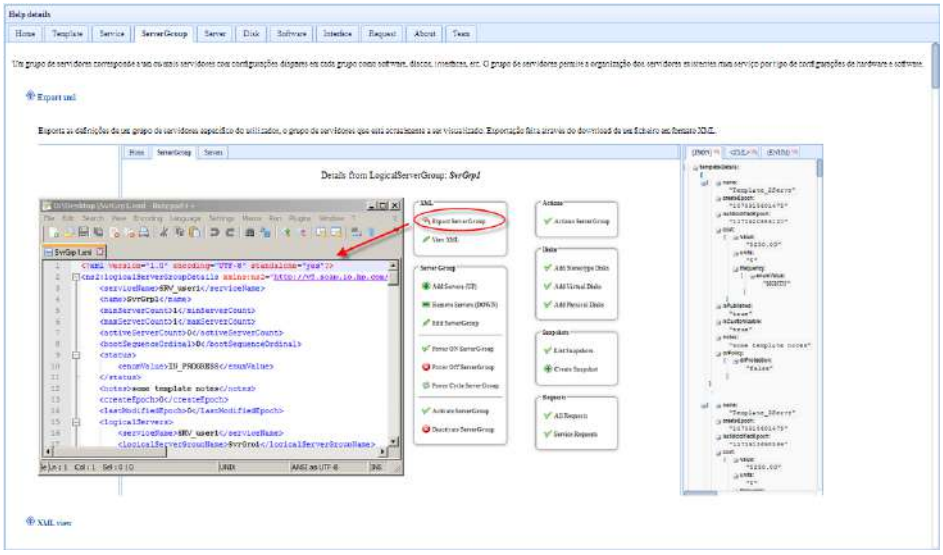
Figura 107 - Service info Tab



N.3 ServerGroup

Um Grupo de Servidores corresponde a um ou mais servidores com configurações díspares em cada grupo como software, discos, interfaces, etc. O grupo de servidores permite a organização dos servidores existentes num serviço por tipo de configurações de hardware e software.

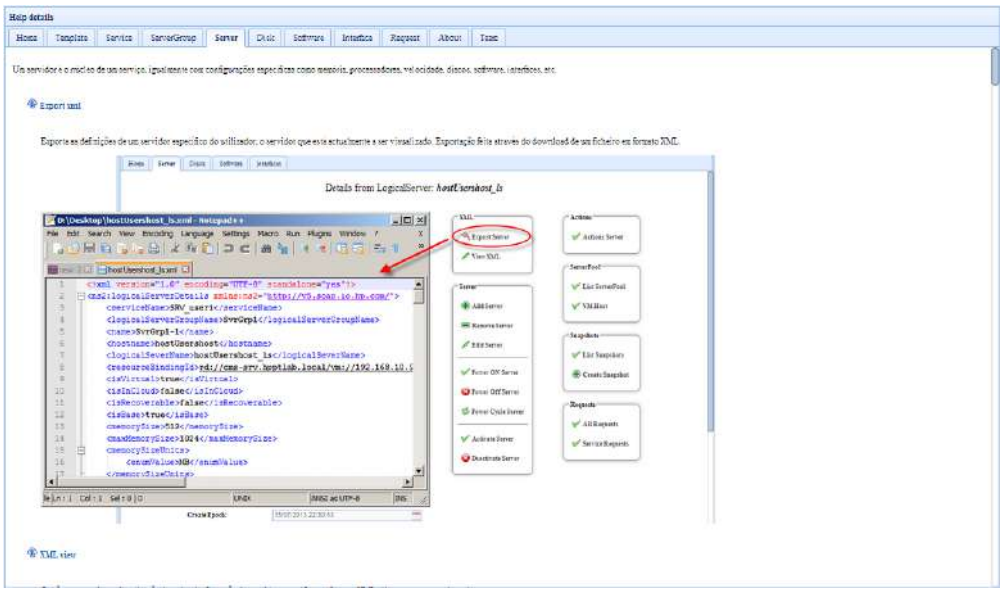
Figura 108 - ServerGroup info Tab



N.4 Server

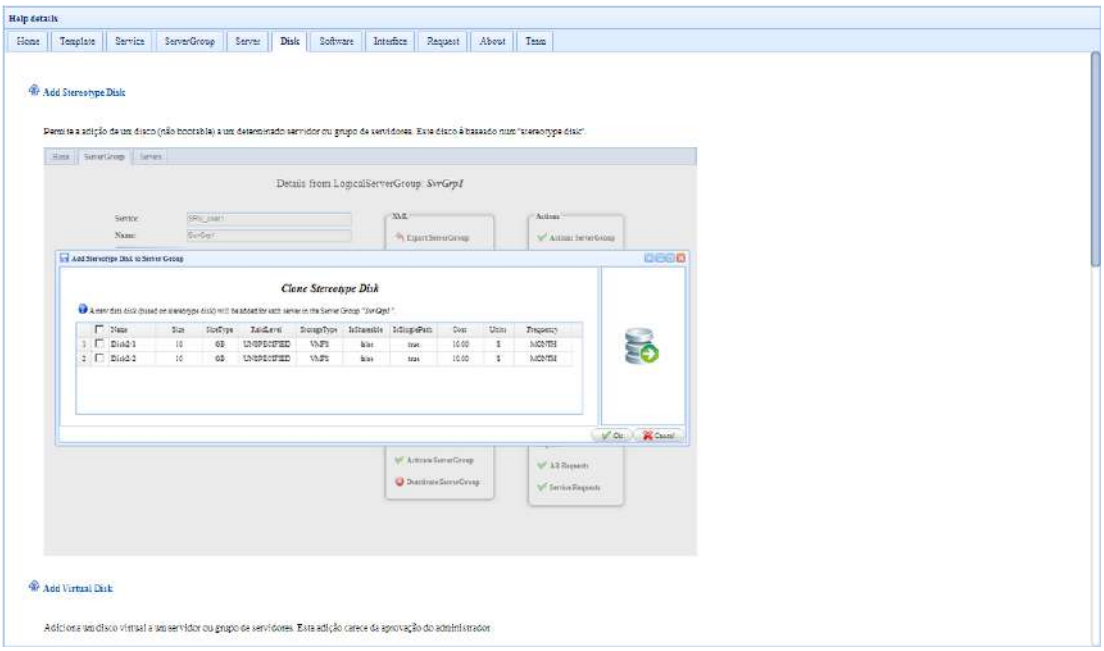
Um servidor é o núcleo de um serviço, igualmente com configurações específicas como memória, processadores, velocidade, discos, software, interfaces, etc.

Figura 109 - Server info Tab



N.5 Disk

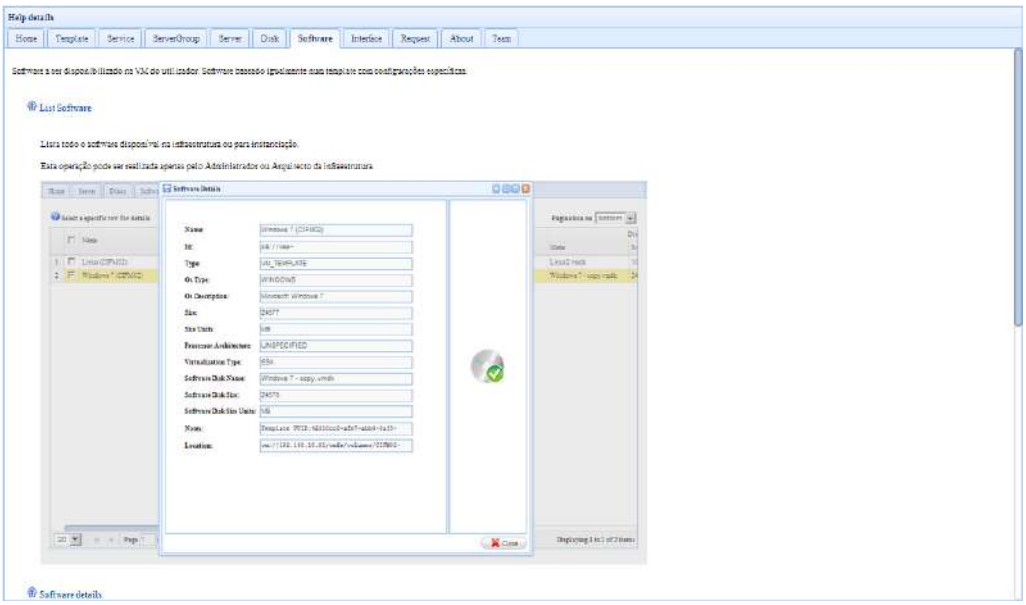
Figura 110 - Disk info Tab



N.6 Software

Software a ser disponibilizado na VM do utilizador. Software baseado igualmente num template com configurações específicas.

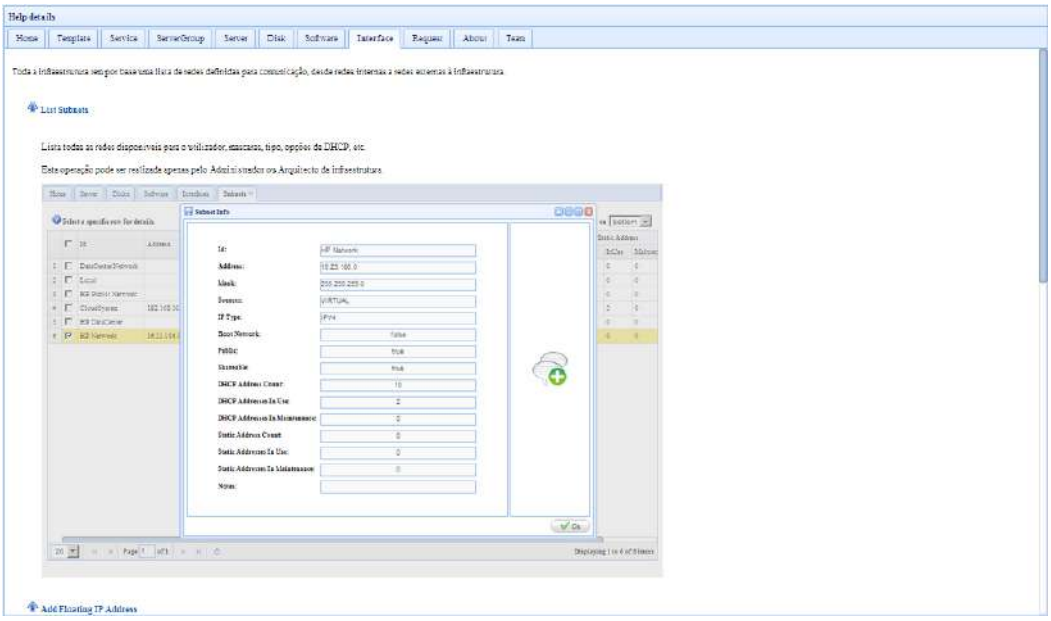
Figura 111 - Software info Tab



N.7 Interface

Toda a infra-estrutura tem por base uma lista de redes definidas para comunicação, desde redes internas a redes externas à infra-estrutura.

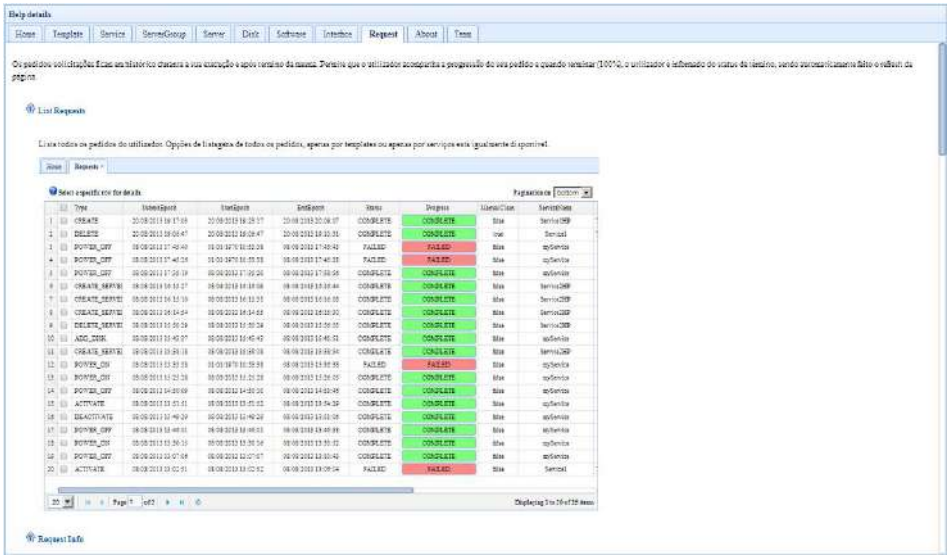
Figura 112 - Interface info Tab



N.8 Request

Os pedidos/solicitações ficam em histórico durante a sua execução e após término da mesma. Permite que o utilizador acompanhe a progressão do seu pedido e quando terminar (100%), o utilizador é informado do status de término, sendo automaticamente feito o refresh da página.

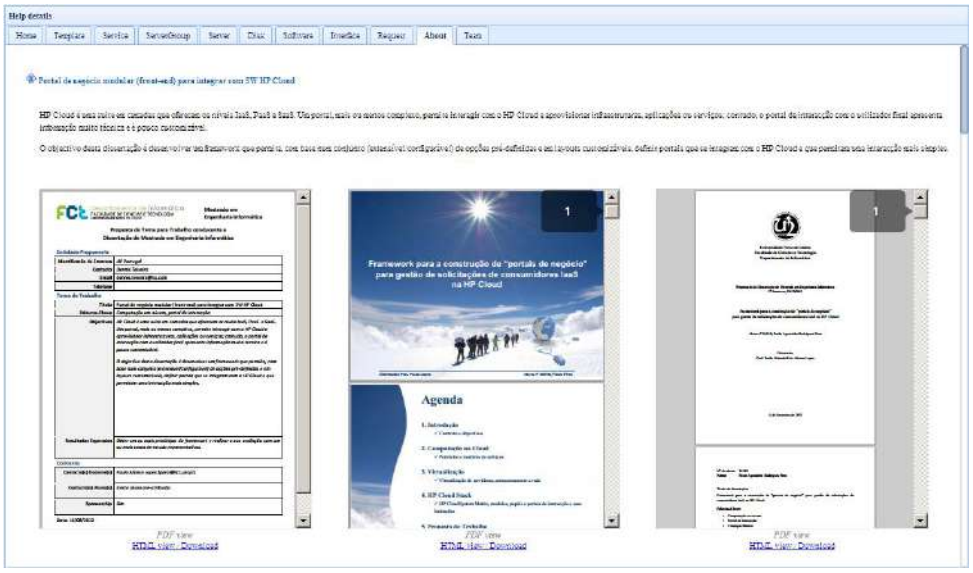
Figura 113 - Request info Tab



N.9 About

Um pequeno resumo desta Tese.

Figura 114 - About info Tab



N.10 Team

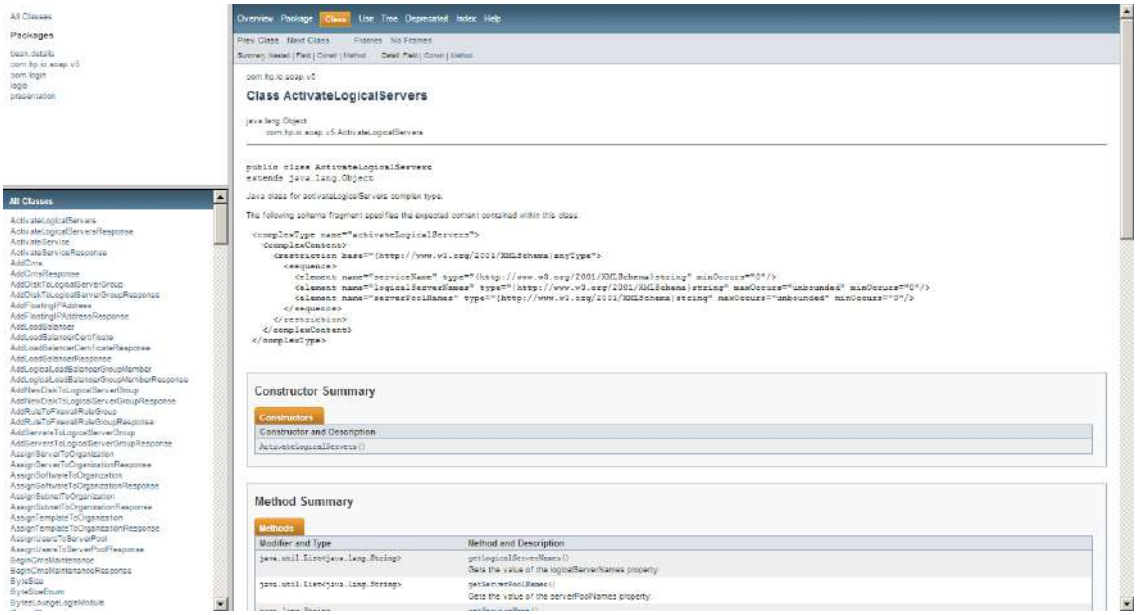
Intervenientes de todo o projecto.

Figura 115 - Team info Tab

N.11 Javadoc

Informação sobre todo o webservice, métodos, módulos de acesso, login, configurações, etc. Informação disponível apenas ao Administrador.

Figura 116 - Javadoc info Tab



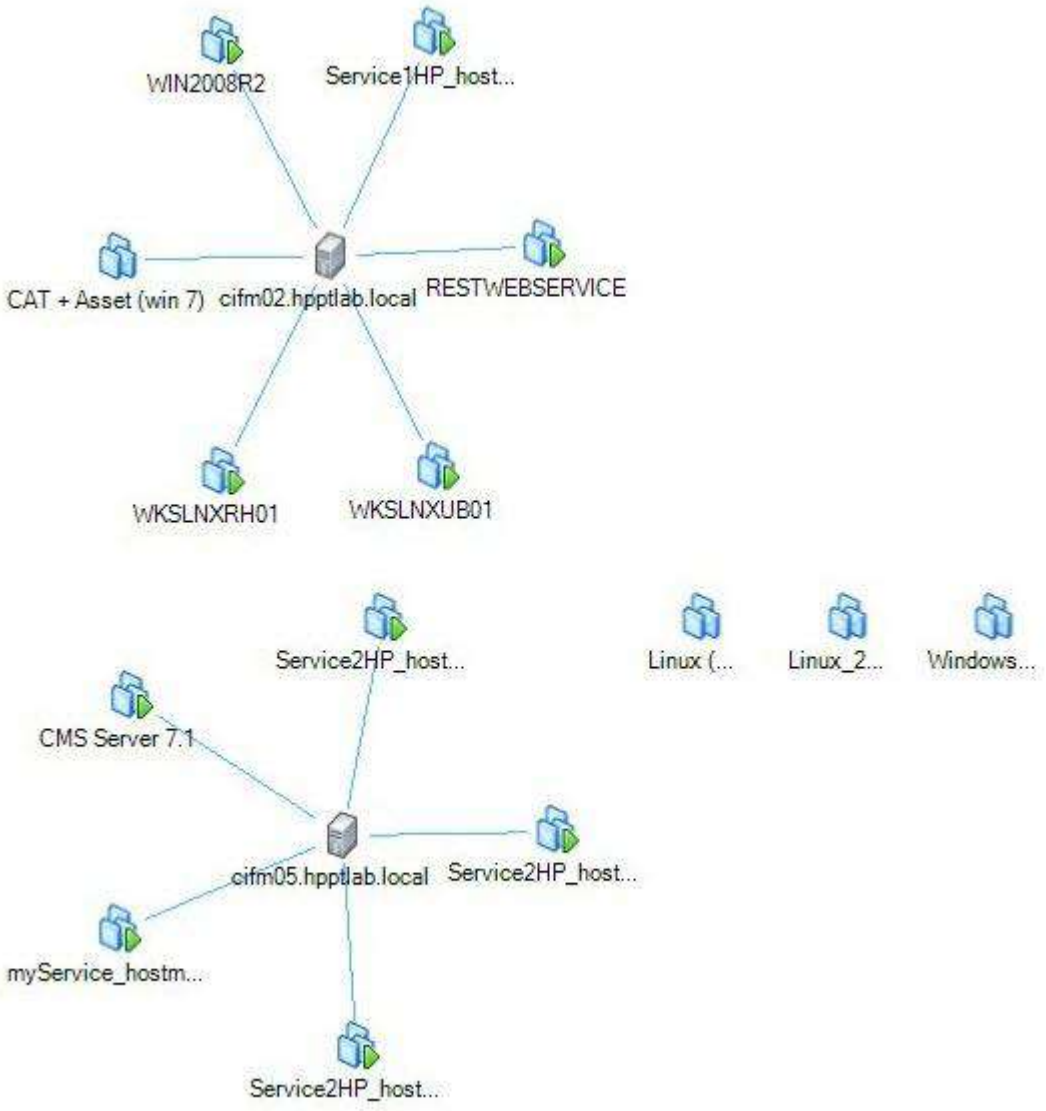
Anexo O

O.1 “Maps” – Host Options – Host to VM

Detalhes da relação Host/VM no final dos 3 serviços criados

- ✓ ***Host:*** cifm02.hpptlab.local / ***VM:*** Service1HP_hostonehost5
- ✓ ***Host:*** cifm05.hpptlab.local / ***VM:*** myService_hostmyhostr01, Service2HP_hostrhostr01, 02 e 03

Figura 117 - “Maps” da relação Host / VM

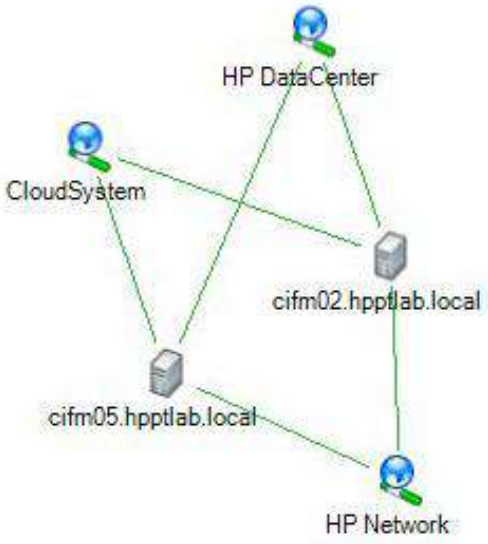


O.2 “Maps” – Host Options – Host to Network

Detalhes da relação Host/Network no final dos 3 serviços criados. Os 2 hosts (cifm02.hpptlab.local e cifm05.hpptlab.local) estão ligados às 3 redes existentes.

- ✓ ***CloudSystem:*** rede interna MOE (192.168.10…)
- ✓ ***HP Network:*** rede externa MOE e interna HP (16.23.186… e 16.23.187…)
- ✓ ***HP DataCenter:*** outra rede interna MOE

Figura 118 - “Maps” da relação Host / Network

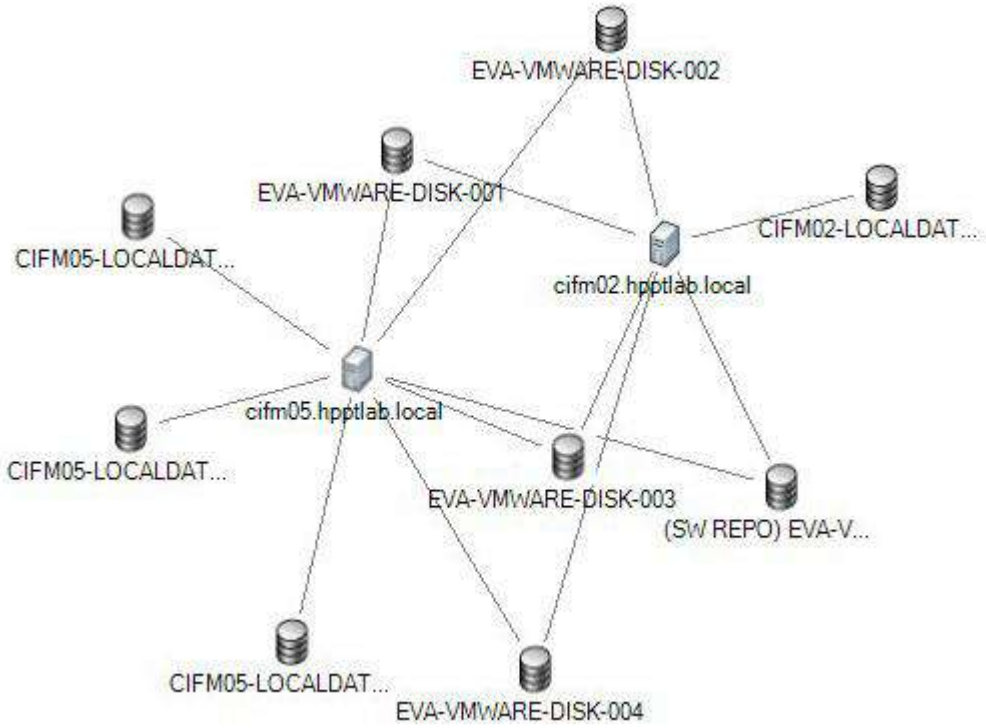


O.3 “Maps” – Host Options – Host to Datastore

Detalhes da relação Host/Datastore no final dos 3 serviços criados

- ✓ ***Host:*** cifm02.hpptlab.local / ***Datastore:*** CIF02-LOCALDATASTORE-001
- ✓ ***Host:*** cifm05.hpptlab.local / ***Datastore:*** CIF05-LOCALDATASTORE-001, 002, 003 e 004
- ✓ ***Datastore:*** EVA-VMware-DISK-001, 002, 003, 004 3 005

Figura 119 - “Maps” da relação Host / Datastore

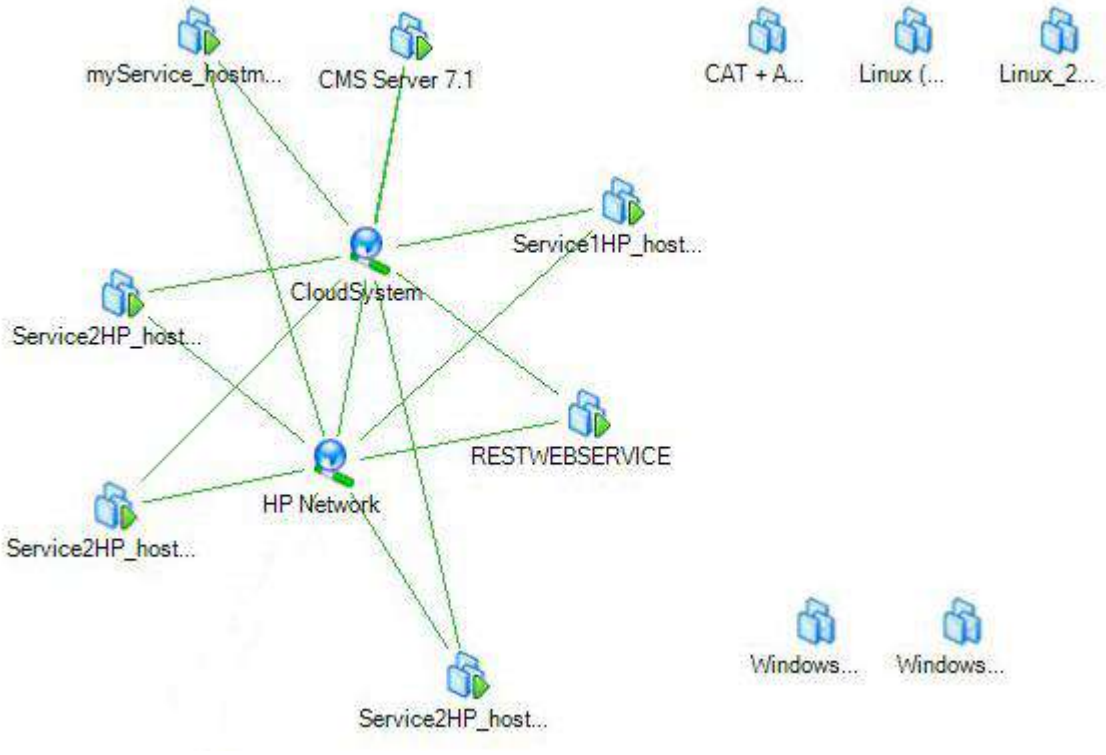


O.4 “Maps” – VM Options – VM to Network

Detalhes da relação VM/Network no final dos 3 serviços criados

- ✓ ***Network:*** CloudSystem / ***VM:*** todas as VMs na rede local MOE
- ✓ ***Network:*** HP Network / ***VM:*** todas as VMs na rede externa MOE

Figura 120 - “Maps” da relação VM / Network

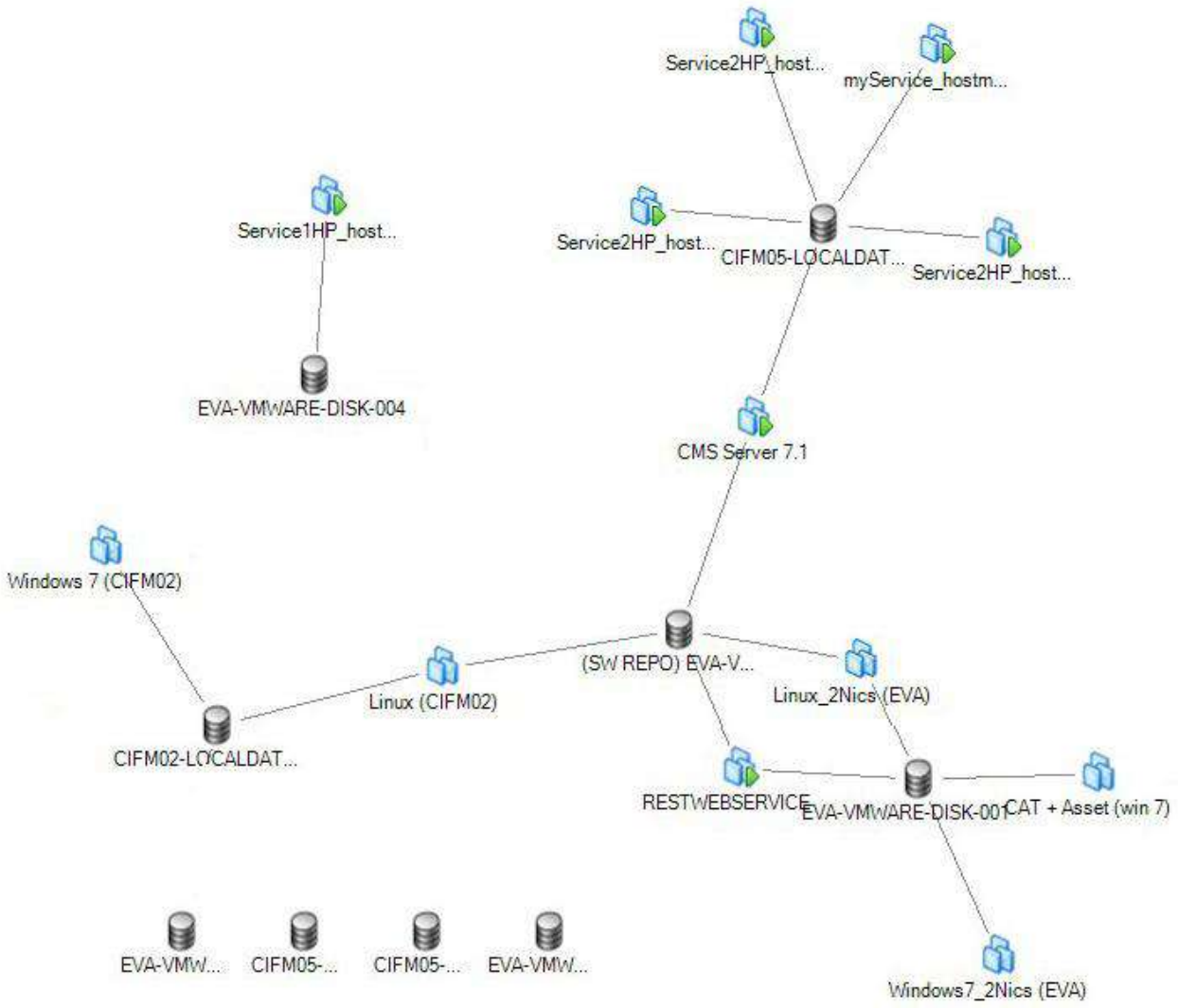


O.5 “Maps” – VM Options – VM to Datastore

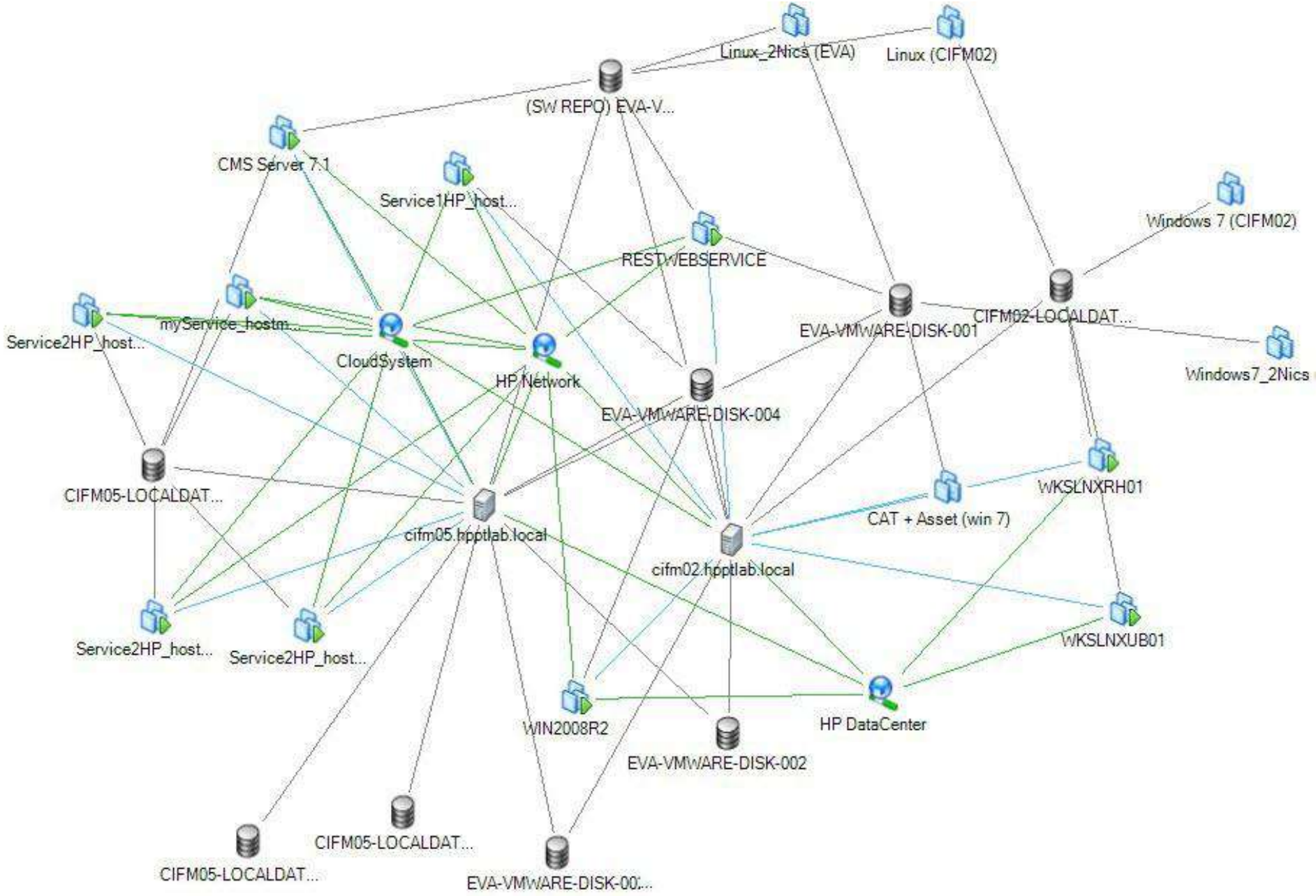
Detalhes da relação Host/Datastore no final dos 3 serviços criados

- ✓ ***VM:*** Service1HP_hostonehost5 / ***Datastore:*** EVA-VMware-DISK-004
- ✓ ***VM:*** myService_hostmyhostr01 / ***Datastore:*** CIF05-LOCALDATASTORE-004
- ✓ ***VM:*** Service2HP_hostrhostr01 / ***Datastore:*** CIF05-LOCALDATASTORE-004
- ✓ ***VM:*** Service2HP_hostrhostr02 / ***Datastore:*** CIF05-LOCALDATASTORE-004
- ✓ ***VM:*** Service2HP_hostrhostr03 / ***Datastore:*** CIF05-LOCALDATASTORE-004

Figura 121 - “Maps” da relação Host / Datastore



O.6 “Maps” – Host and VM Options – all relationships



Anexo P

P.1 Template

P.1.1 Informação

Figura 122 - Detalhes gerais do Template "Template_3srv"

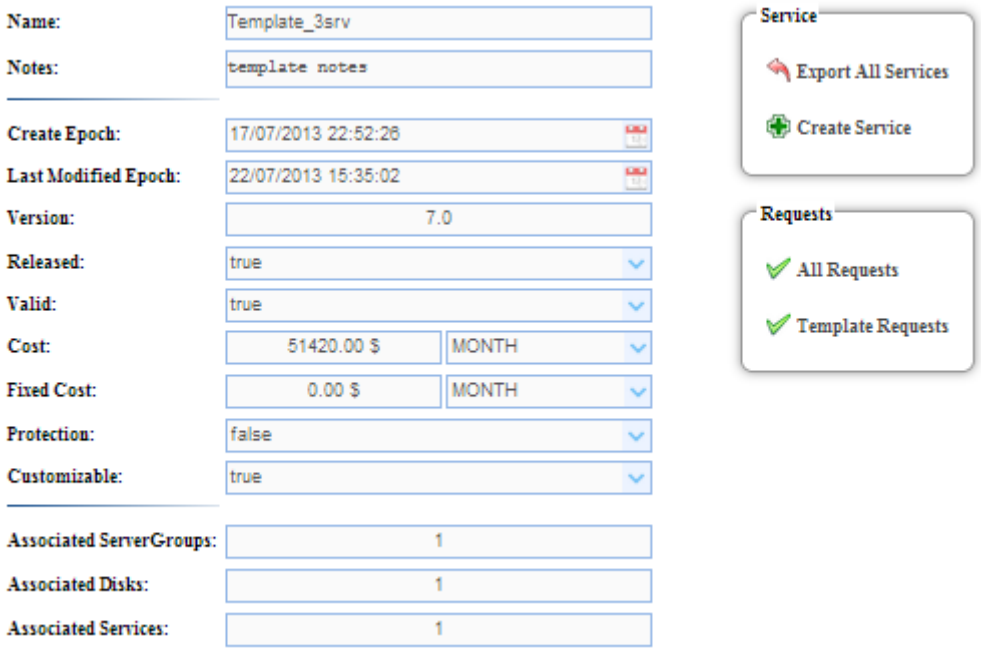


Figura 123 - Detalhes do ServerGroup "SvrGrp1" do Template "Template_3srv"

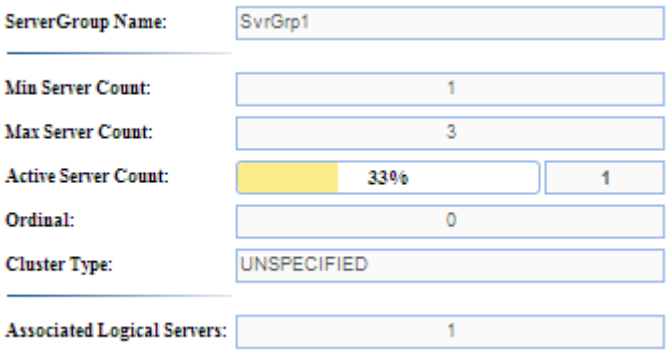


Figura 124 - Detalhes do Servidor “SvrGrp1-1”

Server Name:	SvrGrp1-1	
Hostname:	#hostr01	
Virtualization Type:	ESX	
Max Memory Size:	1024	MB
Memory Size:	50%	512
Max Processor Count:	3	
Processor Count:	33%	1
Processor Speed MHz:	0	
Virtual:	YES	
InCloud:	UNSPECIFIED	

Figura 125 - Detalhes dos Custos do Servidor “SvrGrp1-1”

Server Name:	SvrGrp1-1	
Total Cost per Server:	51400.00 $	MONTH
Base Cost per Server:	100.00 $	MONTH
Cost per Processor:	100.00 $	MONTH
Cost per Memory:	100.00 $	MONTH

Figura 126 - Detalhes do Disco “Disk1-1” do Servidor “SvrGrp1-1”

Name:	Disk1-1	
Storage Type:	VMFS	
Size:	24577	MB
Raid Level:	UNSPECIFIED	
Bootable:	YES	
Shareable:	NO	
Cost:	0.00 $	MONTH

Figura 127 - Detalhes do Software “Windows 7” do Servidor “SvrGrp1-1”

Name:	Windows 7 (CIFM02)
Notes:	Template UUID:42310cc0-afe7-abb4-9af5-2e02972c5eea
Type:	VM_TEMPLATE
OS Type:	WINDOWS
Virtualization Type:	ESX

Figura 128 - Detalhes da/s Interfaces “NIC0” e “NIC1” do Servidor “SvrGrp1-1”

Server Name:	SvrGrp1-1
Hostname:	#hostr01
Interface:	NIC0
Primary:	true
Redundancy Type:	UNSPECIFIED
Subnet Name:	HP Network
Subnet Public:	YES
Subnet Shareable:	YES
Subnet IPType:	IPV4
Addresses Name:	#hostr-1
Assignment Type:	DHCP
Cost:	10.00 $ MONTH

Server Name:	SvrGrp1-1
Hostname:	#hostr01
Interface:	NIC1
Primary:	false
Redundancy Type:	UNSPECIFIED
Subnet Name:	CloudSystem
Subnet Public:	NO
Subnet Shareable:	YES
Subnet IPType:	IPV4
Assignment Type:	AUTO
Cost:	10.00 $ MONTH

P.1.2 Execução

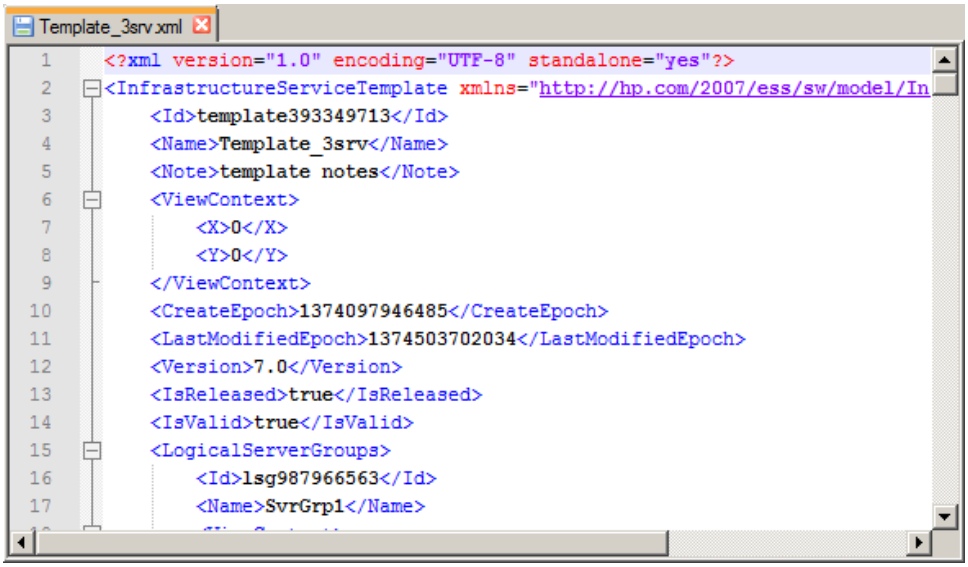
Figura 129 - Ficheiro “TemplateList.xml”

TemplateList.xml

```
1  <?xml version="1.0" encoding="UTF-8" standalone="yes"?><InfrastructureSe
2  <InfrastructureServiceTemplate xmlns="http://hp.com/2007/ess/sw/model/In
3      <Id>template270776809</Id>
4      <Name>Template_1ServerGroup_2NICs</Name>
5      <Note>Template with 1 Server Group and 2 NICs</Note>
6      <ViewContext>
7          <X>0</X>
8          <Y>0</Y>
9      </ViewContext>
10     <CreateEpoch>1374063825190</CreateEpoch>
11     <LastModifiedEpoch>1374503733057</LastModifiedEpoch>
12     <Version>7.0</Version>
13     <IsReleased>true</IsReleased>
14     <IsValid>true</IsValid>
15     <LogicalServerGroups>
16         <Id>template270776809#lsg602652593</Id>
17         <Name>SvrGrp1</Name>
```

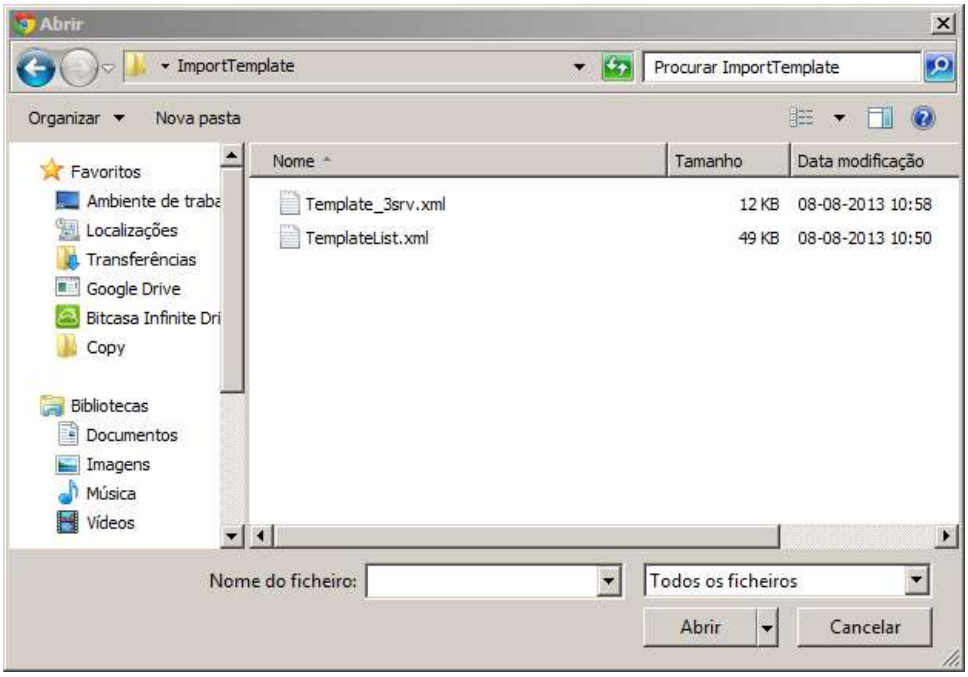
✓ *Export Template*

Figura 130 - Ficheiro “Template_3srv.xml”



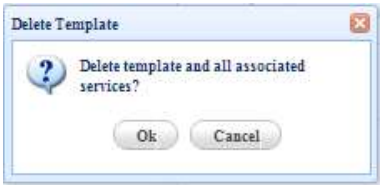
✓ *Import Template*

Figura 131 - Importação de um Template em formato XML



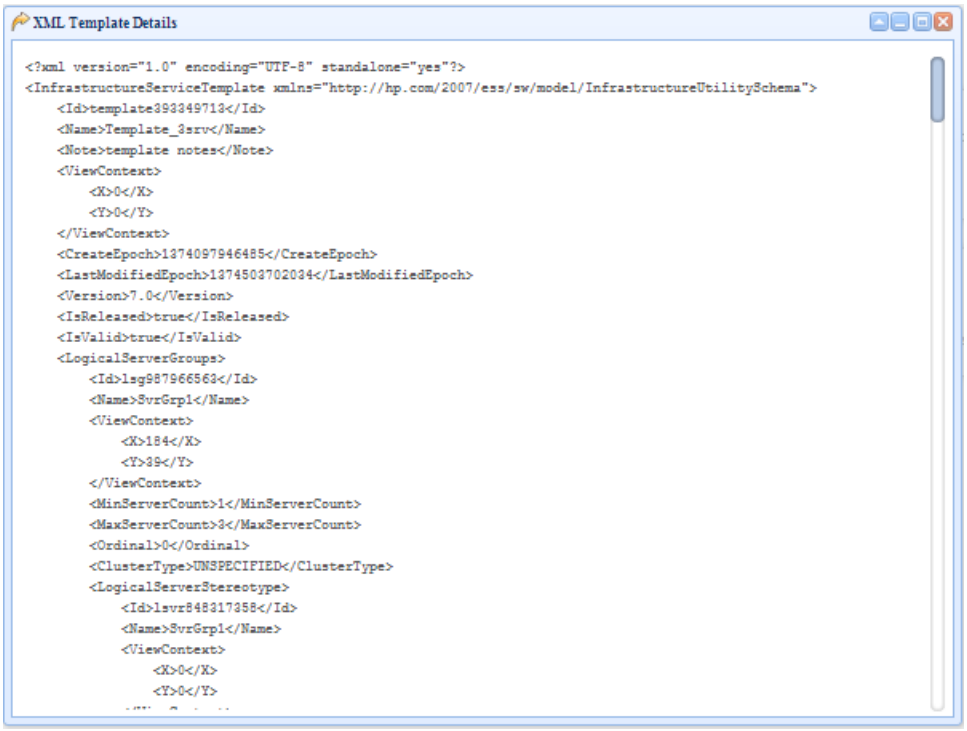
✓ *Delete Template*

Figura 132 - InfoWindow “Delete Template”



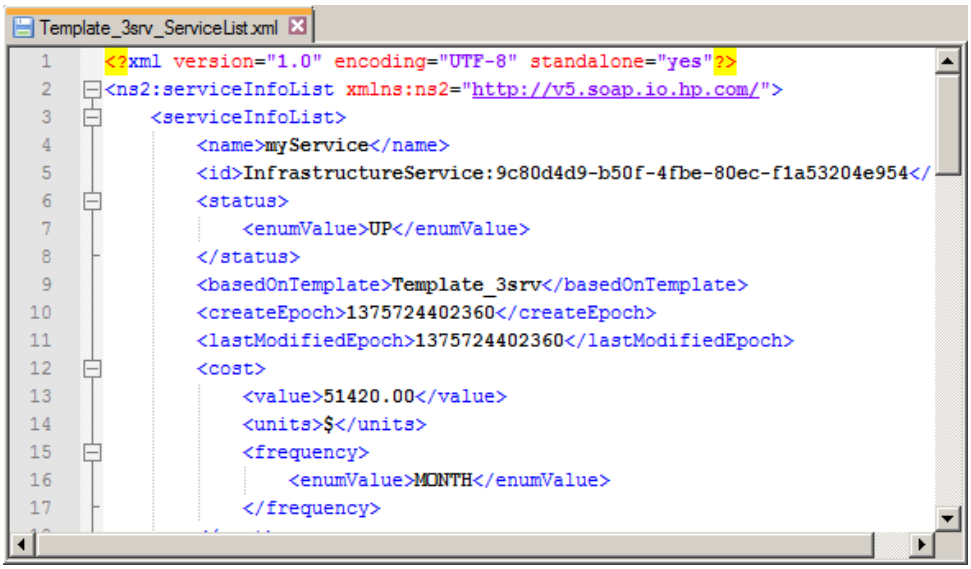
✓ *XML View*

Figura 133 - InfoWindow com a visualização do Template em XML



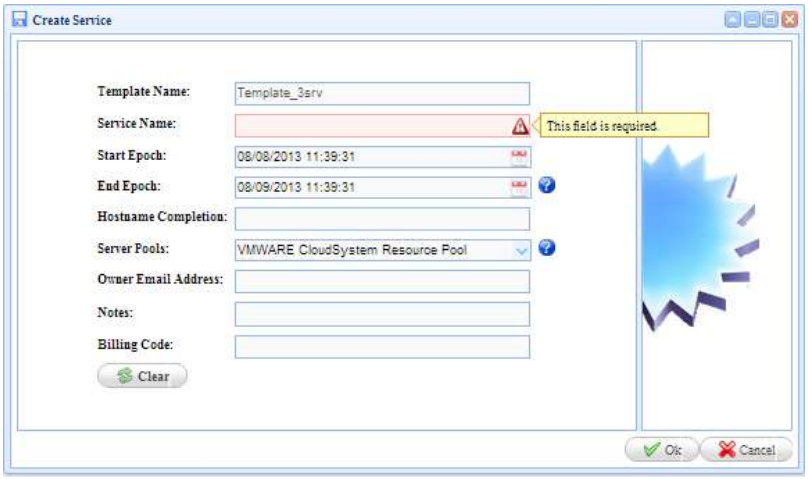
✓ *Export All Services*

Figura 134 - Ficheiro “Template_3srv_ServiceList.xml”



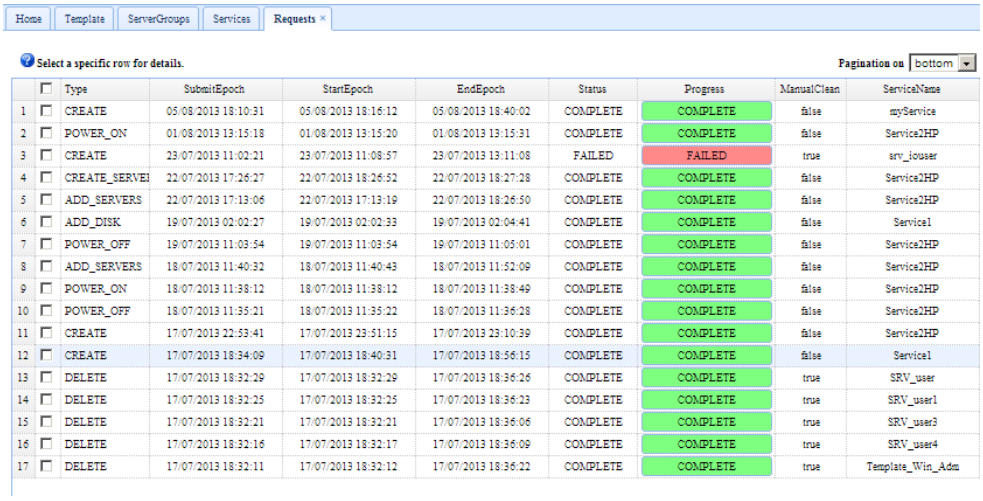
✓ *Create Service*

Figura 135 - Criação de um Serviço com base no Template "Template_3srvl"



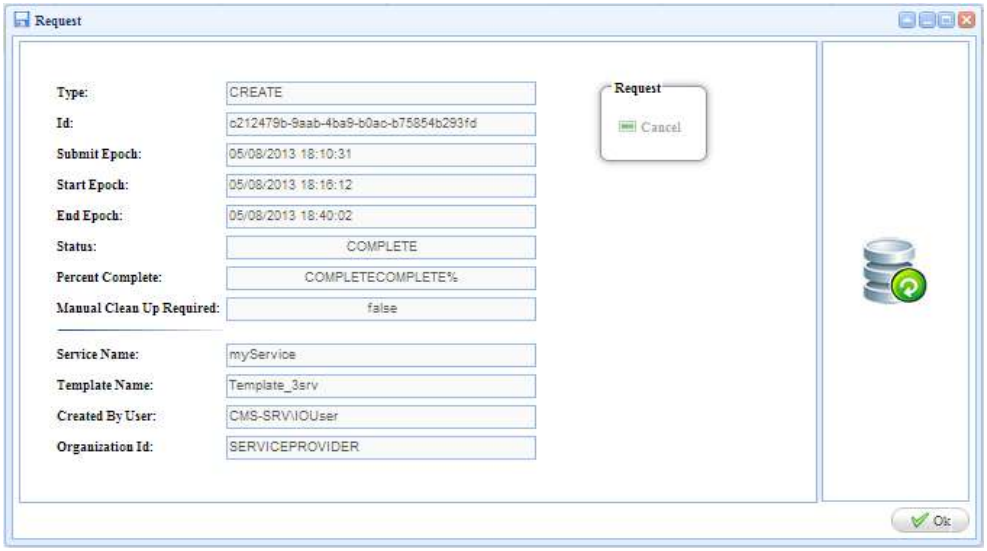
✓ *All Requests*

Figura 136 - Lista de todos os pedidos do utilizador



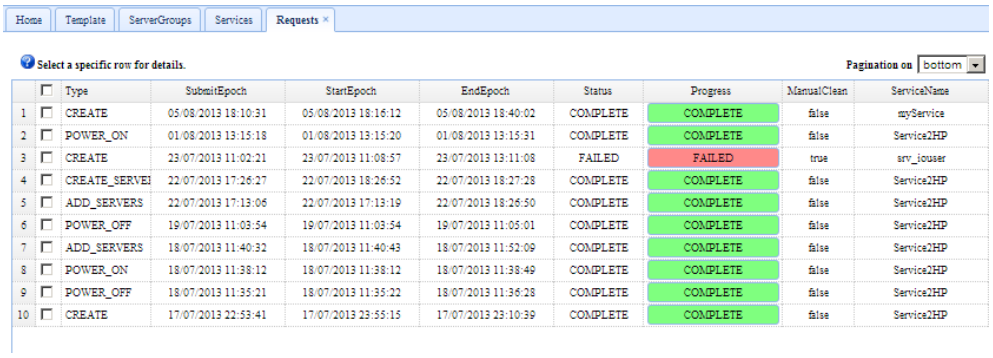
		Type	SubmitEpoch	StartEpoch	EndEpoch	Status	Progress	ManualClean	ServiceName
1	☐	CREATE	05/08/2013 18:10:31	05/08/2013 18:16:12	05/08/2013 18:40:02	COMPLETE	COMPLETE	false	myService
2	☐	POWER_ON	01/08/2013 13:15:18	01/08/2013 13:15:20	01/08/2013 13:15:31	COMPLETE	COMPLETE	false	Service2HP
3	☐	CREATE	23/07/2013 11:02:21	23/07/2013 11:08:57	23/07/2013 13:11:08	FAILED	FAILED	true	srv_iouser
4	☐	CREATE_SERVEI	22/07/2013 17:26:27	22/07/2013 18:26:52	22/07/2013 18:27:28	COMPLETE	COMPLETE	false	Service2HP
5	☐	ADD_SERVERS	22/07/2013 17:13:06	22/07/2013 17:13:19	22/07/2013 18:26:50	COMPLETE	COMPLETE	false	Service2HP
6	☐	ADD_DISK	19/07/2013 02:02:27	19/07/2013 02:02:33	19/07/2013 02:04:41	COMPLETE	COMPLETE	false	Service1
7	☐	POWER_OFF	19/07/2013 11:03:54	19/07/2013 11:03:54	19/07/2013 11:05:01	COMPLETE	COMPLETE	false	Service2HP
8	☐	ADD_SERVERS	18/07/2013 11:40:32	18/07/2013 11:40:43	18/07/2013 11:52:09	COMPLETE	COMPLETE	false	Service2HP
9	☐	POWER_ON	18/07/2013 11:38:12	18/07/2013 11:38:12	18/07/2013 11:38:49	COMPLETE	COMPLETE	false	Service2HP
10	☐	POWER_OFF	18/07/2013 11:35:21	18/07/2013 11:35:22	18/07/2013 11:36:28	COMPLETE	COMPLETE	false	Service2HP
11	☐	CREATE	17/07/2013 22:53:41	17/07/2013 23:51:15	17/07/2013 23:10:39	COMPLETE	COMPLETE	false	Service2HP
12	☐	CREATE	17/07/2013 18:34:09	17/07/2013 18:40:31	17/07/2013 18:56:15	COMPLETE	COMPLETE	false	Service1
13	☐	DELETE	17/07/2013 18:32:29	17/07/2013 18:32:29	17/07/2013 18:36:26	COMPLETE	COMPLETE	true	SRV_user
14	☐	DELETE	17/07/2013 18:32:25	17/07/2013 18:32:25	17/07/2013 18:36:23	COMPLETE	COMPLETE	true	SRV_user1
15	☐	DELETE	17/07/2013 18:32:21	17/07/2013 18:32:21	17/07/2013 18:36:06	COMPLETE	COMPLETE	true	SRV_user3
16	☐	DELETE	17/07/2013 18:32:16	17/07/2013 18:32:17	17/07/2013 18:36:09	COMPLETE	COMPLETE	true	SRV_user4
17	☐	DELETE	17/07/2013 18:32:11	17/07/2013 18:32:12	17/07/2013 18:36:22	COMPLETE	COMPLETE	true	Template_Win_Adm

Figura 137 - Eventual cancelamento de um pedido



✓ *Template Requests*

Figura 138 - Lista dos pedidos resultantes da relação Template/utilizador

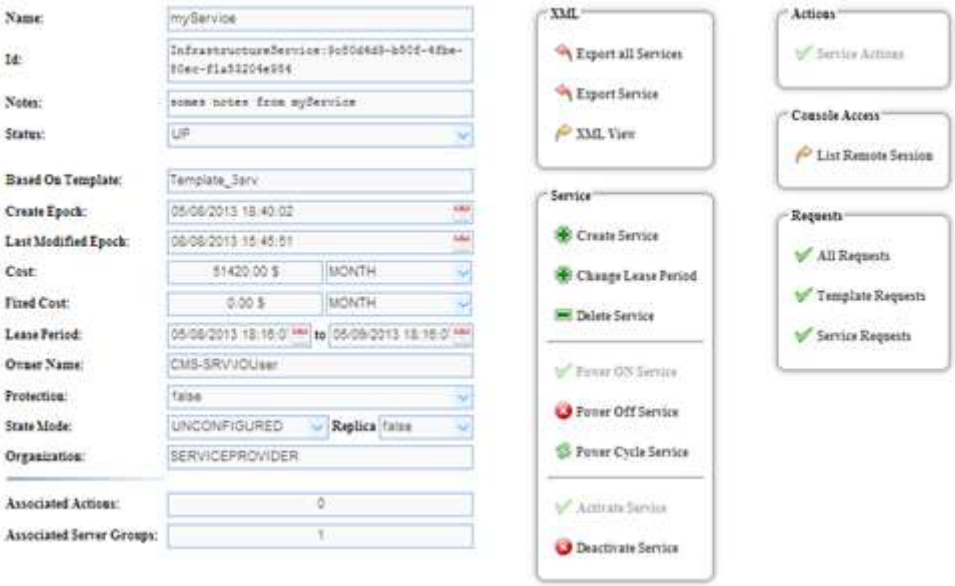


	Type	SubmitEpoch	StartEpoch	EndEpoch	Status	Progress	ManualClean	ServiceName
1	CREATE	05/08/2013 18:10:31	05/08/2013 18:16:12	05/08/2013 18:40:02	COMPLETE	COMPLETE	false	myService
2	POWER_ON	01/08/2013 13:15:18	01/08/2013 13:15:20	01/08/2013 13:15:31	COMPLETE	COMPLETE	false	Service2HP
3	CREATE	23/07/2013 11:02:21	23/07/2013 11:08:57	23/07/2013 13:11:08	FAILED	FAILED	true	srv_iouser
4	CREATE_SERVEI	22/07/2013 17:26:27	22/07/2013 18:26:52	22/07/2013 18:27:28	COMPLETE	COMPLETE	false	Service2HP
5	ADD_SERVERS	22/07/2013 17:13:06	22/07/2013 17:13:19	22/07/2013 18:26:50	COMPLETE	COMPLETE	false	Service2HP
6	POWER_OFF	19/07/2013 11:03:54	19/07/2013 11:03:54	19/07/2013 11:05:01	COMPLETE	COMPLETE	false	Service2HP
7	ADD_SERVERS	18/07/2013 11:40:32	18/07/2013 11:40:43	18/07/2013 11:52:09	COMPLETE	COMPLETE	false	Service2HP
8	POWER_ON	18/07/2013 11:38:12	18/07/2013 11:38:12	18/07/2013 11:38:49	COMPLETE	COMPLETE	false	Service2HP
9	POWER_OFF	18/07/2013 11:35:21	18/07/2013 11:35:22	18/07/2013 11:36:28	COMPLETE	COMPLETE	false	Service2HP
10	CREATE	17/07/2013 22:53:41	17/07/2013 23:55:15	17/07/2013 23:10:39	COMPLETE	COMPLETE	false	Service2HP

P.2 Serviço

P.2.1 Informação

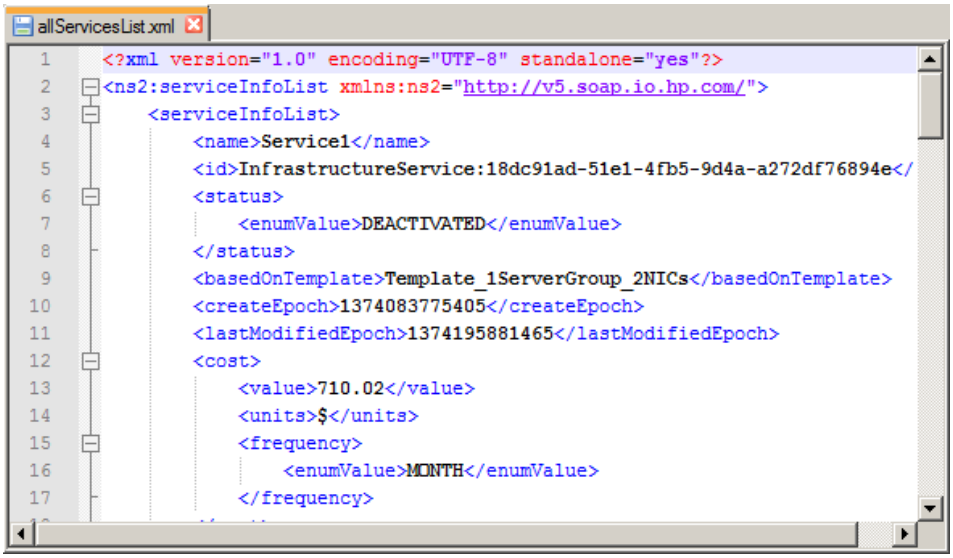
Figura 139 - Detalhes do Serviço “myService” do Template “Template_3srv”



P.2.2 Execução

- ✓ *Export All Services*

Figura 140 - Ficheiro “allServicesList.xml”



```
allServicesList.xml
1  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2  <ns2:serviceInfoList xmlns:ns2="http://v5.soap.io.hp.com/">
3      <serviceInfoList>
4          <name>Service1</name>
5          <id>InfrastructureService:18dc91ad-51e1-4fb5-9d4a-a272df76894e</
6          <status>
7              <enumValue>DEACTIVATED</enumValue>
8          </status>
9          <basedOnTemplate>Template_1ServerGroup_2NICs</basedOnTemplate>
10         <createEpoch>1374083775405</createEpoch>
11         <lastModifiedEpoch>1374195881465</lastModifiedEpoch>
12         <cost>
13             <value>710.02</value>
14             <units>$</units>
15             <frequency>
16                 <enumValue>MONTH</enumValue>
17             </frequency>
```

✓ *Export Service*

Figura 141 - Ficheiro “myService.xml”


```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:serviceDetails xmlns:ns2="http://v5.soap.io.hp.com/">
    <name>myService</name>
    <id>InfrastructureService:9c80d4d9-b50f-4fbe-80ec-f1a53204e954</id>
    <status>
        <enumValue>UP</enumValue>
    </status>
    <basedOnTemplate>Template_3srv</basedOnTemplate>
    <createEpoch>1375724402360</createEpoch>
    <lastModifiedEpoch>1375724402360</lastModifiedEpoch>
    <cost>
        <value>51420.00</value>
        <units>$</units>
        <frequency>
            <enumValue>MONTH</enumValue>
        </frequency>
    </cost>
    <leasePeriod>
```


✓ *XML View*

Figura 142 - InfoWindow com a visualização do Serviço em XML

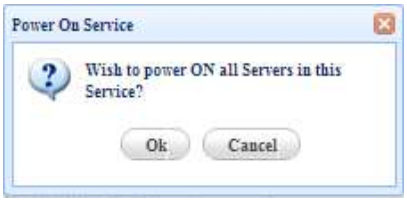

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:serviceDetails xmlns:ns2="http://v5.soap.io.hp.com/">
    <name>myService</name>
    <id>InfrastructureService:9c80d4d9-b50f-4fbe-80ec-f1a53204e954</id>
    <status>
        <enumValue>UP</enumValue>
    </status>
    <basedOnTemplate>Template_3srv</basedOnTemplate>
    <createEpoch>1375724402360</createEpoch>
    <lastModifiedEpoch>1375724402360</lastModifiedEpoch>
    <cost>
        <value>51420.00</value>
        <units>$</units>
        <frequency>
            <enumValue>MONTH</enumValue>
        </frequency>
    </cost>
    <leasePeriod>
        <startEpoch>1375722967000</startEpoch>
        <endEpoch>1378401367000</endEpoch>
    </leasePeriod>
    <ownerName>CMS-SRV\IOUser</ownerName>
    <notes>somes notes from myService</notes>
    <billingCode></billingCode>
    <drPolicy>
        <drProtection>false</drProtection>
    </drPolicy>
    <drState>
        <drMode>
            <enumValue>UNCONFIGURED</enumValue>
        </drMode>
```


✓ *Delete Service*

Figura 143 - Remoção de um Serviço

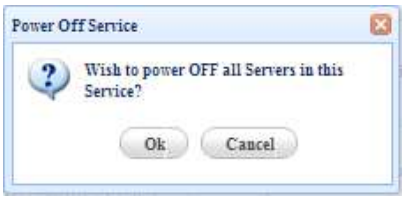
✓ *Power ON Service*

Figura 144 - Power ON de um Serviço



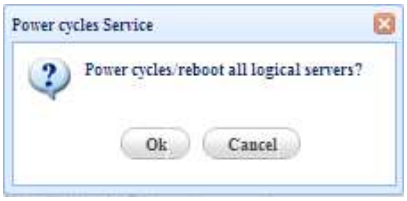
✓ *Power OFF Service*

Figura 145 - Power OFF de um Serviço



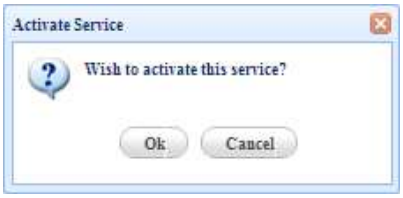
✓ *Power Cycle Service*

Figura 146 - Power Cycle de um Serviço



✓ *Activate Service*

Figura 147 - Activação de um Serviço



✓ *Deactivate Service*

Figura 148 - Desactivação de um Serviço

✓ *Service Actions*

Figura 149 - Lista dos Service Actions

Home | Service | ServerGroups | Actions ×

Select a specific row for details. Pagination on bottom

		TemplateName	ServiceName	Actions					
				Type	Trigger	Size	Name	Ordinal	Path
1		TemplateWin	TemplateWin_SRV	CREATE	SERVICE_BEGIN	1	05_CheckServerHasBooted_server_end	0	/Library/H
2		TemplateWin	TemplateWin_SRV	CREATE_REPLICA	SERVICE_END	1	05_CheckServerHasBooted_server_end	0	/Library/H
3		TemplateWin	TemplateWin_SRV	ADD_SERVERS	SERVICE_END	1	05_CheckServerHasBooted_server_end	0	/Library/H
4		TemplateWin	TemplateWin_SRV	REMOVE_SERVERS	SERVICE_END	1	05_CheckServerHasBooted_server_end	0	/Library/H
5		TemplateWin	TemplateWin_SRV	CHANGE_LEASE_PERIOD	SERVICE_BEGIN	1	05_CheckServerHasBooted_server_end	0	/Library/H
6		TemplateWin	TemplateWin_SRV	POWER_OFF	SERVICE_BEGIN	1	05_CheckServerHasBooted_server_end	0	/Library/H
7		TemplateWin	TemplateWin_SRV	DELETE_REPLICA	SERVICE_END	1	05_CheckServerHasBooted_server_end	0	/Library/H
8		TemplateWin	TemplateWin_SRV	DELETE_SERVER_SNAPSHOT	SERVICE_BEGIN	1	05_CheckServerHasBooted_server_end	0	/Library/H

✓ *List Remote Session*

Figura 150 - Lista dos Acessos Remotos do um Serviço

Home | Service | ServerGroups | RemoteSessions ×

Select a specific row for details. Pagination on bottom

		Name	Hostname	Software	Virtual	Cloud	Memory		Processor		Status	Address		VMAccess			
							Used	Max	Used	Max		Ip	Subnet	RDP	RDPfile	Telnet	VMRC
1		SvrGrp1-1	host/hostr01	WINDOWS	true	false	512 (MB)	1024 (MB)	1	3	UP	16.23.187.31	HP Network	RDP	RDPfile	Telnet	VMRC
2		SvrGrp1-1	host/hostr01	WINDOWS	true	false	512 (MB)	1024 (MB)	1	3	UP	192.168.10.200	CloudSystem	RDP	RDPfile	Telnet	VMRC
3		SvrGrp1-2	host/hostr02	WINDOWS	true	false	512 (MB)	1024 (MB)	1	3	UP	16.23.187.32	HP Network	RDP	RDPfile	Telnet	VMRC
4		SvrGrp1-2	host/hostr02	WINDOWS	true	false	512 (MB)	1024 (MB)	1	3	UP	192.168.10.201	CloudSystem	RDP	RDPfile	Telnet	VMRC
5		SvrGrp1-3	host/hostr03	WINDOWS	true	false	512 (MB)	1024 (MB)	1	3	UP	16.23.187.1	HP Network	RDP	RDPfile	Telnet	VMRC
6		SvrGrp1-3	host/hostr03	WINDOWS	true	false	512 (MB)	1024 (MB)	1	3	UP	192.168.10.204	CloudSystem	RDP	RDPfile	Telnet	VMRC

Figura 151 - Acesso remoto por RDP

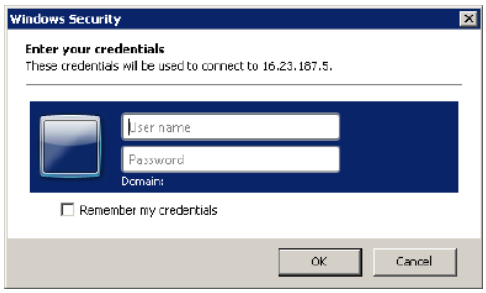


Figura 152 - Acesso remoto por Telnet

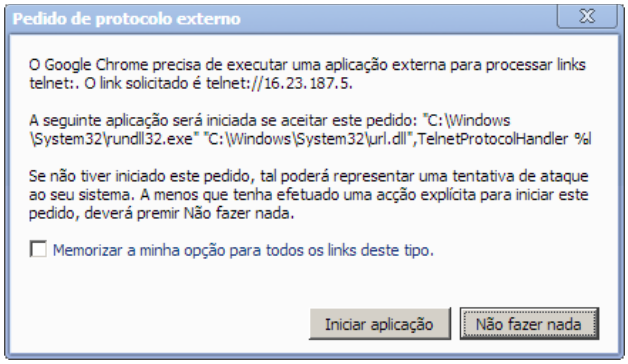
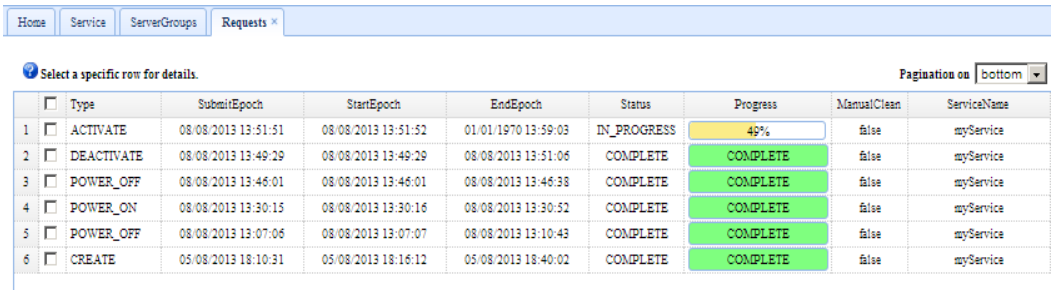


Figura 153 - Acesso remoto por VMRC

✓ *Service Requests*

Figura 154 - Listagem dos pedidos por Serviço



		Type	SubmitEpoch	StartEpoch	EndEpoch	Status	Progress	ManualClean	ServiceName
1	☐	ACTIVATE	08/08/2013 13:51:51	08/08/2013 13:51:52	01/01/1970 13:59:03	IN_PROGRESS	49%	false	myService
2	☐	DEACTIVATE	08/08/2013 13:49:29	08/08/2013 13:49:29	08/08/2013 13:51:06	COMPLETE	COMPLETE	false	myService
3	☐	POWER_OFF	08/08/2013 13:46:01	08/08/2013 13:46:01	08/08/2013 13:46:38	COMPLETE	COMPLETE	false	myService
4	☐	POWER_ON	08/08/2013 13:30:15	08/08/2013 13:30:16	08/08/2013 13:30:52	COMPLETE	COMPLETE	false	myService
5	☐	POWER_OFF	08/08/2013 13:07:06	08/08/2013 13:07:07	08/08/2013 13:10:43	COMPLETE	COMPLETE	false	myService
6	☐	CREATE	05/08/2013 18:10:31	05/08/2013 18:16:12	05/08/2013 18:40:02	COMPLETE	COMPLETE	false	myService

P.3 Grupo de Servidores

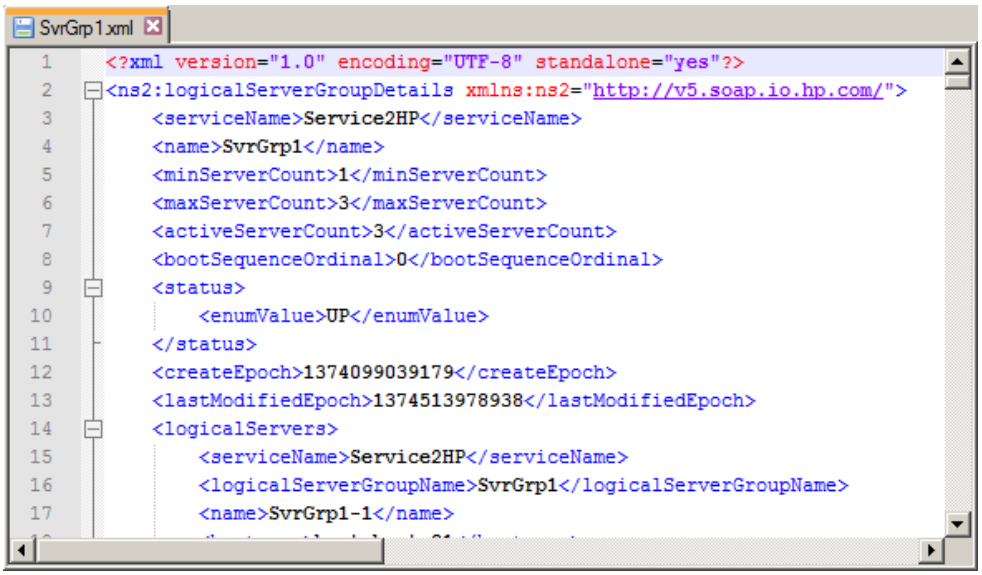
P.3.1 Informação

Figura 155 - Detalhes do Grupo de Servidores “SvrGrp1” do Serviço “myService”

P.3.2 Execução

✓ *Export Server Group*

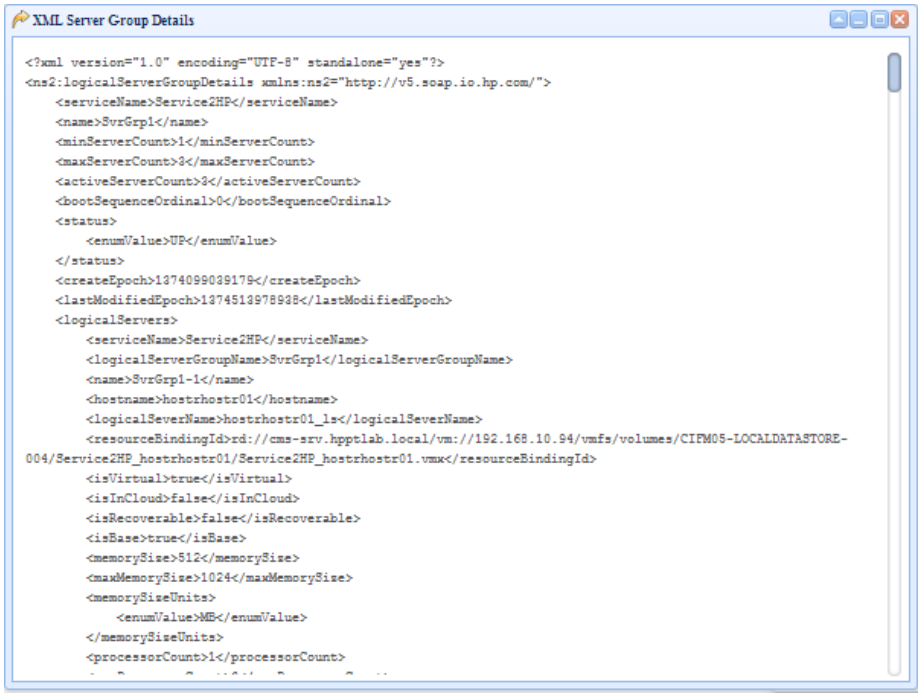
Figura 156 - Ficheiro “SvrGrp1.xml”



```
SvrGrp1.xml
1  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2  <ns2:logicalServerGroupDetails xmlns:ns2="http://v5.soap.io.hp.com/">
3      <serviceName>Service2HP</serviceName>
4      <name>SvrGrp1</name>
5      <minServerCount>1</minServerCount>
6      <maxServerCount>3</maxServerCount>
7      <activeServerCount>3</activeServerCount>
8      <bootSequenceOrdinal>0</bootSequenceOrdinal>
9      <status>
10         <enumValue>UP</enumValue>
11     </status>
12     <createEpoch>1374099039179</createEpoch>
13     <lastModifiedEpoch>1374513978938</lastModifiedEpoch>
14     <logicalServers>
15         <serviceName>Service2HP</serviceName>
16         <logicalServerGroupName>SvrGrp1</logicalServerGroupName>
17         <name>SvrGrp1-1</name>
```

✓ *XML View*

Figura 157 - InfoWindow com a visualização do Grupo de Servidores em XML

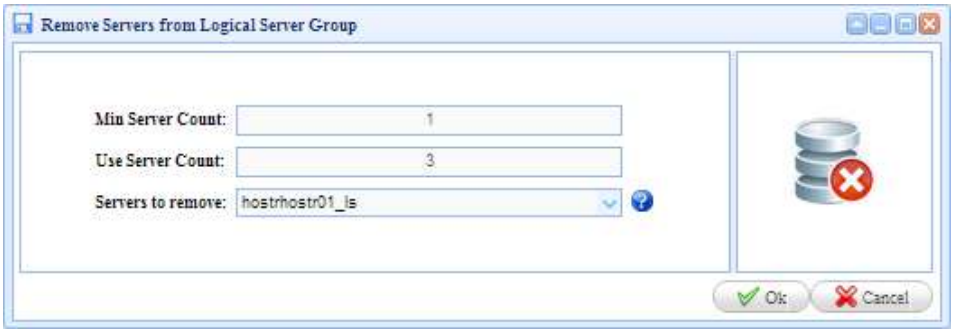


✓ *Add Servers (UP)*

Figura 158 - Adição de Servidores ao Grupo de Servidores

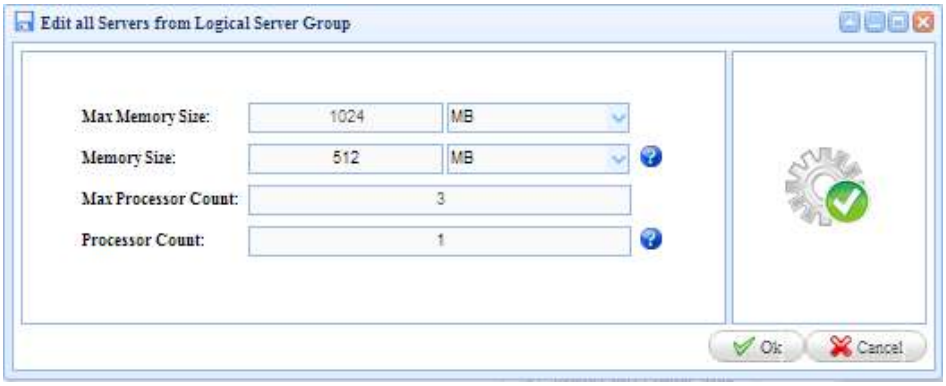
✓ *Remove Servers (DOWN)*

Figura 159 - Remoção de Servidores ao Grupo de Servidores



✓ *Edit ServerGroup*

Figura 160 - Edição de Servidores ao Grupo de Servidores



✓ *Power ON ServerGroup*

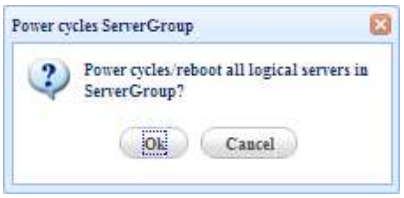
Figura 161 - Power ON de um ServerGroup

✓ *Power OFF ServerGroup*

Figura 162 - Power OFF de um ServerGroup

✓ *Power Cycle ServerGroup*

Figura 163 - Power Cycle de um ServerGroup



✓ *Activate ServerGroup*

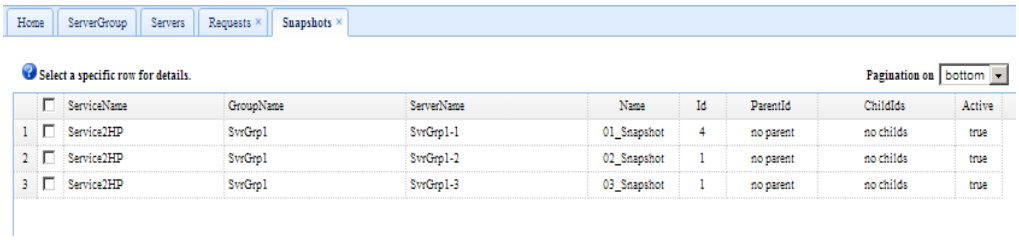
Figura 164 - Activação de um ServerGroup

✓ *Deactivate ServerGroup*

Figura 165 - Desactivação de um ServerGroup

✓ *List Snapshots*

Figura 166 - Listagem de todos os Snapshots existentes no Grupo de Servidores



		ServiceName	GroupName	ServerName	Name	Id	ParentId	ChildIds	Active
1	☐	Service2HP	SvrGrp1	SvrGrp1-1	01_Snapshot	4	no parent	no childs	true
2	☐	Service2HP	SvrGrp1	SvrGrp1-2	02_Snapshot	1	no parent	no childs	true
3	☐	Service2HP	SvrGrp1	SvrGrp1-3	03_Snapshot	1	no parent	no childs	true

Figura 167 - Acções Snapshot/Servidor

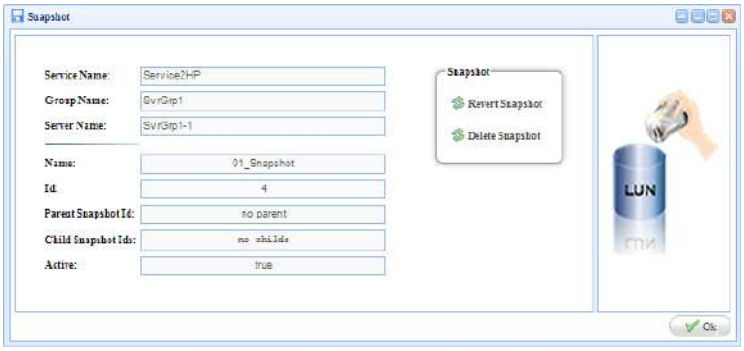


Figura 168 - Revert Snapshot

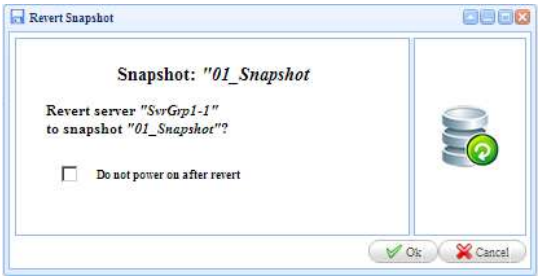
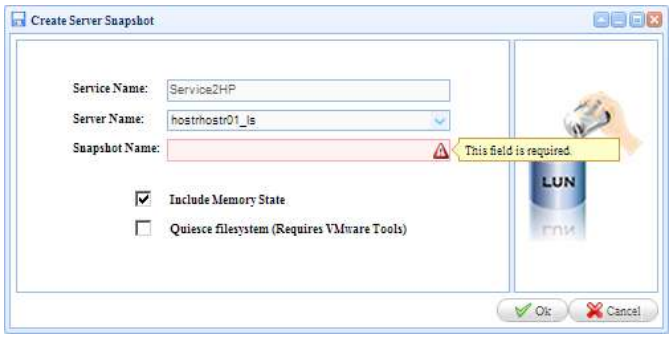


Figura 169 - Delete Snapshot

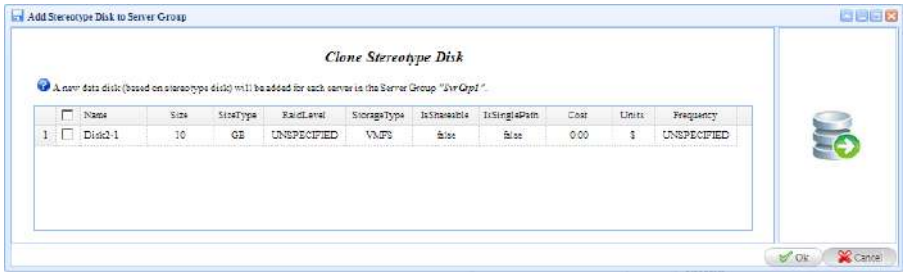
✓ *Create Snapshot*

Figura 170 - Create Snapshot



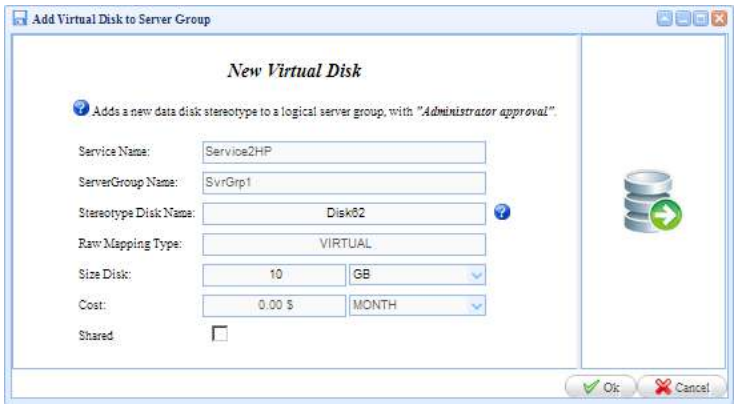
✓ *Add Stereotype Disks*

Figura 171 - Lista de "Stereotype Disks"



✓ *Add Virtual Disks*

Figura 172 - Adicionar um disco virtual "Disk62" ao Grupo de Servidores



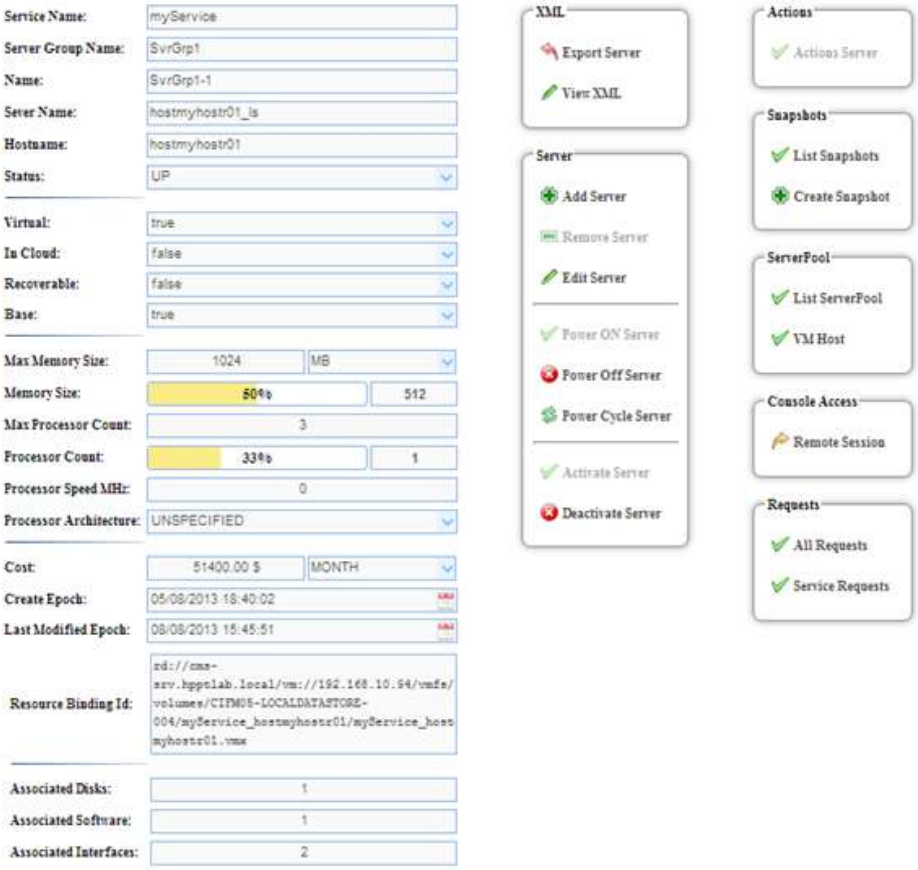
✓ *Add Physical Disks*

Figura 173 - Adicionar um disco físico “Disk788” ao Grupo de Servidores

P.4 Servidores

P.4.1 Informação

Figura 174 - Detalhes do Servidor “SvrGrp1-1” do Serviço “myService”



P.4.2 Execução

✓ *Export Server*

Figura 175 - Ficheiro “hostmygostr01_ls.xml”


```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:logicalServerDetails xmlns:ns2="http://v5.soap.io.hp.com/">
    <serviceName>myService</serviceName>
    <logicalServerGroupName>SvrGrp1</logicalServerGroupName>
    <name>SvrGrp1-1</name>
    <hostname>hostmyhostr01</hostname>
    <logicalSeverName>hostmyhostr01_ls</logicalSeverName>
    <resourceBindingId>rd://cms-srv.hpptlab.local/vm://192.168.10.94/vmf
    <isVirtual>true</isVirtual>
    <isInCloud>false</isInCloud>
    <isRecoverable>false</isRecoverable>
    <isBase>true</isBase>
    <memorySize>512</memorySize>
    <maxMemorySize>1024</maxMemorySize>
    <memorySizeUnits>
        <enumValue>MB</enumValue>
    </memorySizeUnits>
```


✓ *XML View*

Figura 176 - InfoWindow com a visualização do Servidor em XML


```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:logicalServerDetails xmlns:ns2="http://v5.soap.io.hp.com/">
    <serviceName>myService</serviceName>
    <logicalServerGroupName>SvrGrp1</logicalServerGroupName>
    <name>SvrGrp1-1</name>
    <hostname>hostmyhostr01</hostname>
    <logicalSeverName>hostmyhostr01_ls</logicalSeverName>
    <resourceBindingId>rd://cms-srv.hpptlab.local/vm://192.168.10.94/vmfs/volumes/CIFM05-LOCALDATASTORE-
004/myService_hostmyhostr01/myService_hostmyhostr01.vmx</resourceBindingId>
    <isVirtual>true</isVirtual>
    <isInCloud>false</isInCloud>
    <isRecoverable>false</isRecoverable>
    <isBase>true</isBase>
    <memorySize>512</memorySize>
    <maxMemorySize>1024</maxMemorySize>
    <memorySizeUnits>
        <enumValue>MB</enumValue>
    </memorySizeUnits>
    <processorCount>1</processorCount>
    <maxProcessorCount>3</maxProcessorCount>
    <processorSpeedMHz>0</processorSpeedMHz>
    <processorArchitecture>
        <enumValue>UNSPECIFIED</enumValue>
    </processorArchitecture>
    <status>
        <enumValue>UP</enumValue>
    </status>
    <cost>
        <value>51400.00</value>
        <units>$</units>
        <frequency>
```

✓ *Remove Server*

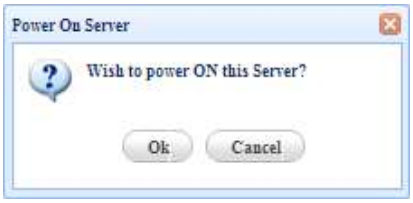
Figura 177 - Remoção de um Servidor

✓ *Edit Server*

Figura 178 - Edição de um Servidor

✓ *Power ON Server*

Figura 179 - Power ON de um Servidor

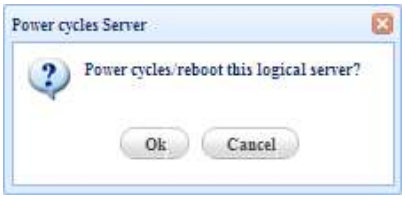


✓ *Power OFF Server*

Figura 180 - Power OFF de um Servidor

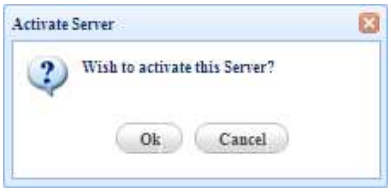
✓ *Power Cycle Server*

Figura 181 - Power Cycle de um Servidor



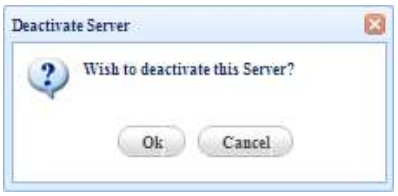
✓ *Activate Server*

Figura 182 - Activação de um Servidor



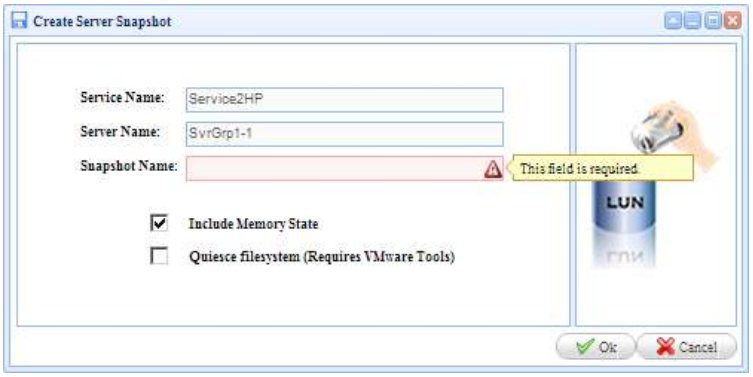
✓ *Deactivate Server*

Figura 183 - Desactivação de um Servidor



✓ *Create Snapshot*

Figura 184 - Create Snapshot



✓ *List ServerPool*

Figura 185 - Pools de Servidores

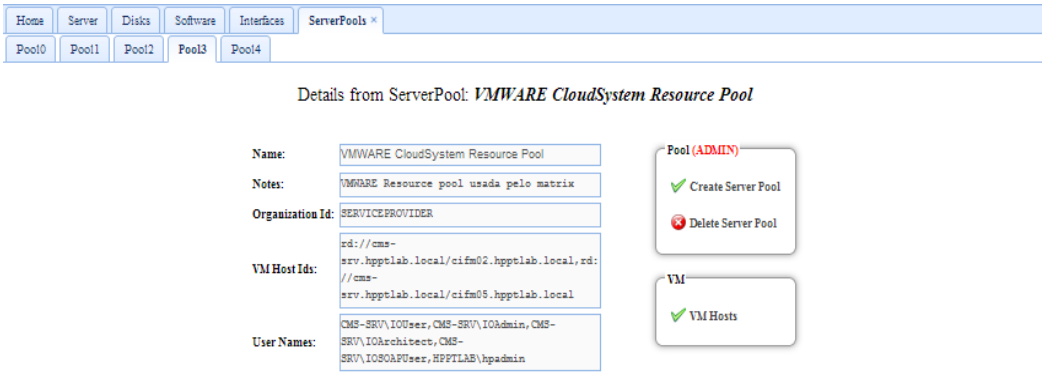


Figura 186 - Criar uma Pools de Servidores

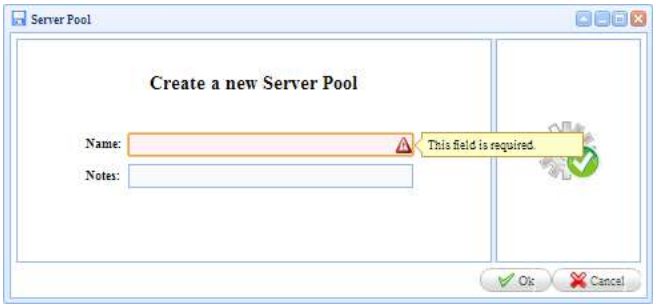


Figura 187 - Remoção de uma Pools de Servidores

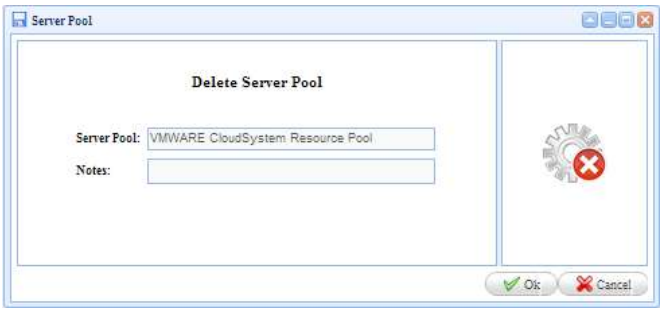
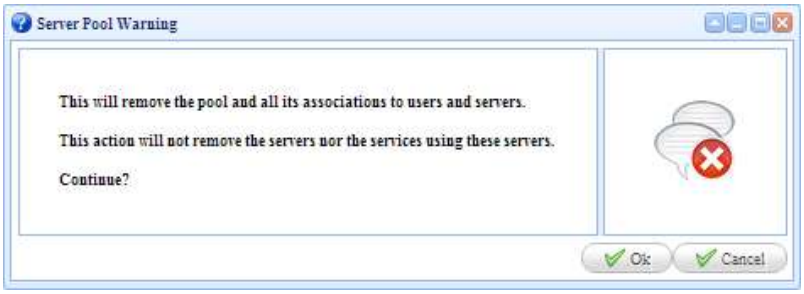
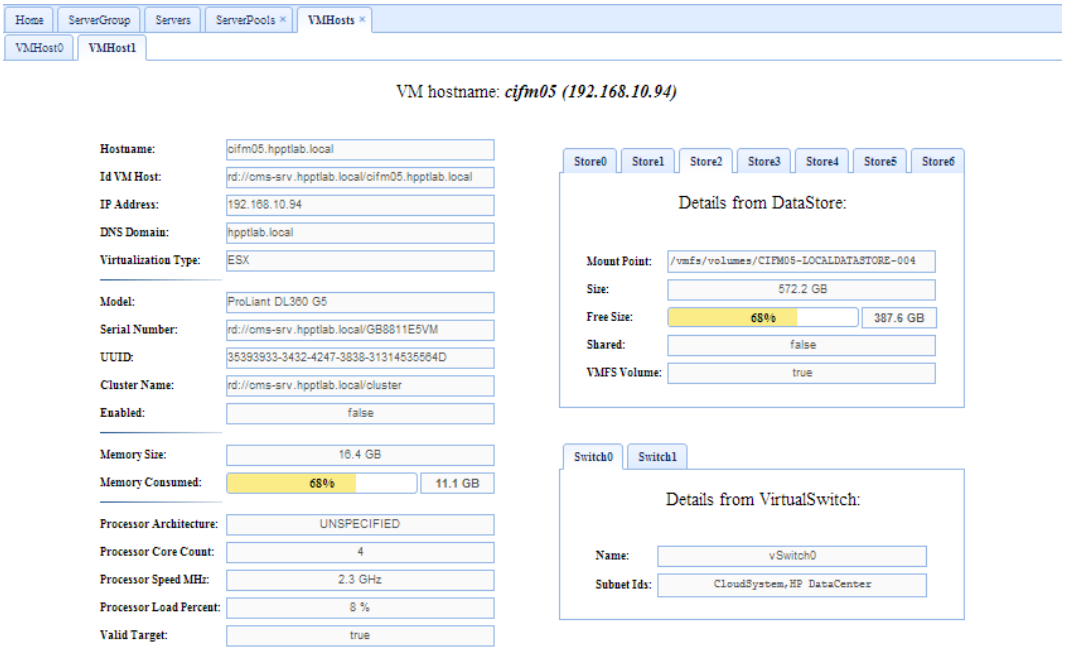


Figura 188 - InfoWindo de aviso ao administrador



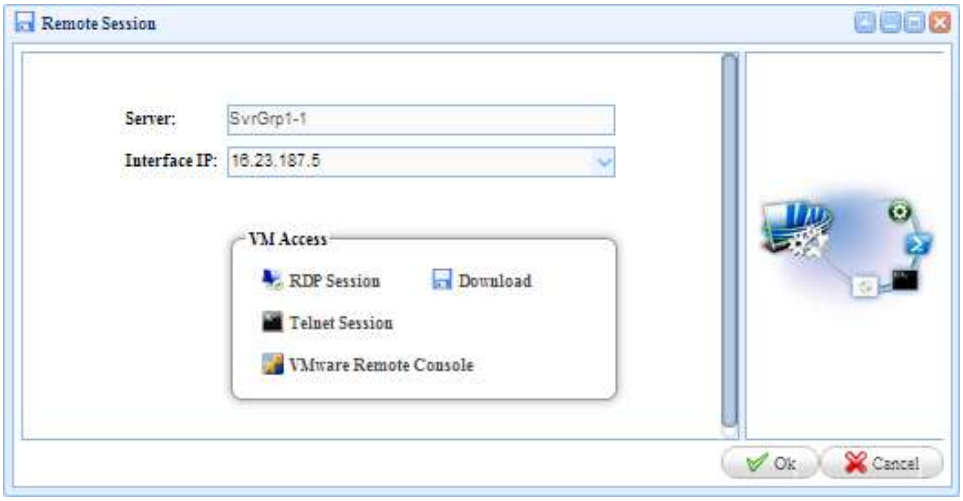
✓ *VM Host*

Figura 189 - Informação dos hosts físicos



✓ *Remote Session*

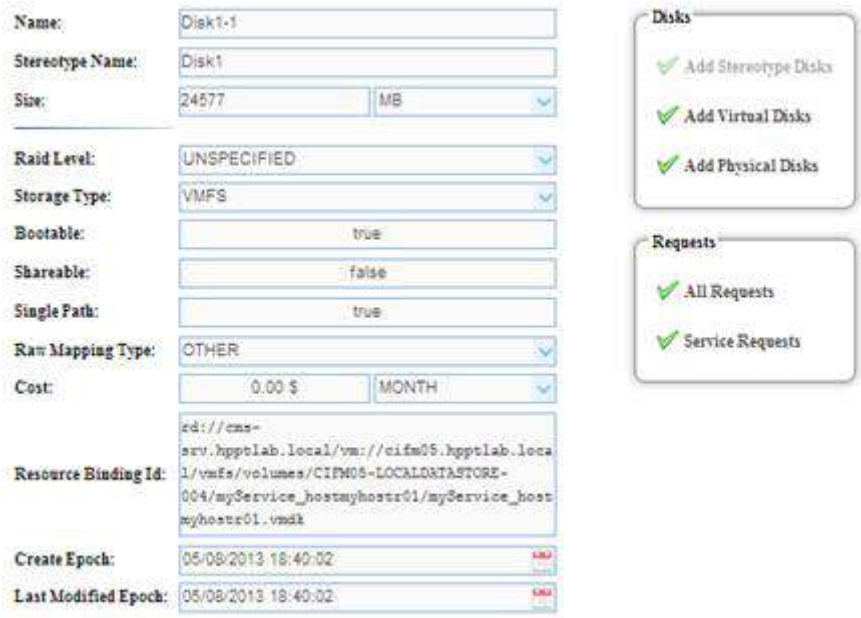
Figura 190 - Sessões Remotas num Servidor



P.5 Discos

P.5.1 Informação

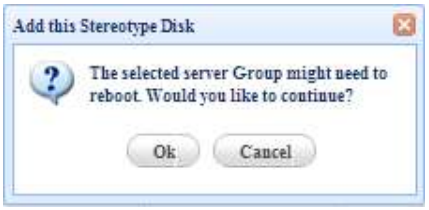
Figura 191 - Detalhes do Disco “Disk1-1” do Servidor “SvrGrp1-1”



P.5.2 Execução

- ✓ *Add Stereotype Disks*

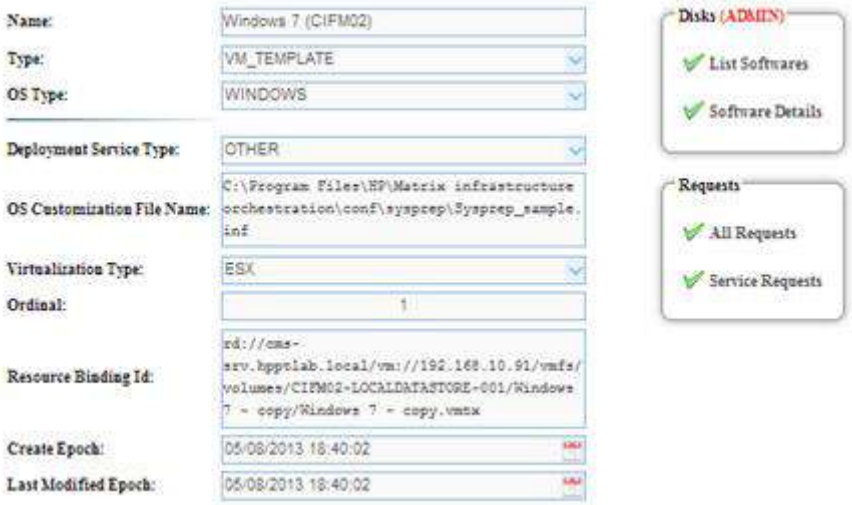
Figura 192 - Criação de um “Stereotype Disks”



P.6 Software

P.6.1 Informação

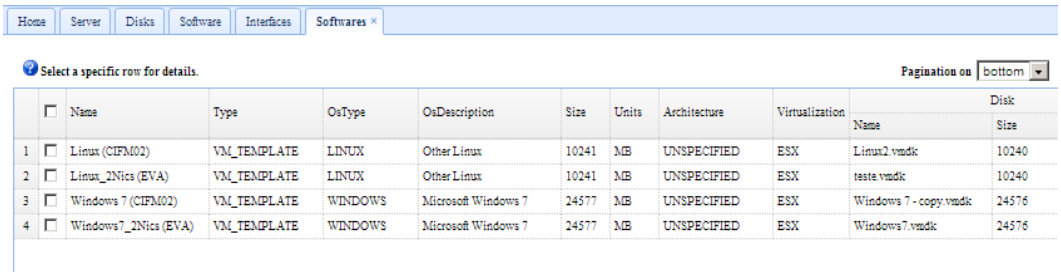
Figura 193 - Detalhes do Software “Windows 7” do Servidor “SvrGrp1-1”



P.6.2 Execução

- ✓ *List Softwares*

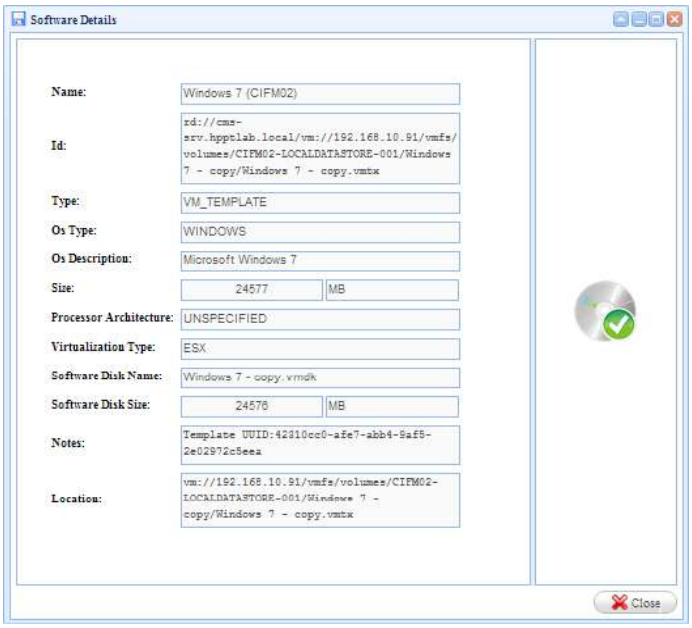
Figura 194 - Lista de Software



		Name	Type	OsType	OsDescription	Size	Units	Architecture	Virtualization	Disk Name	Disk Size
1	☐	Linux (CIFM02)	VM_TEMPLATE	LINUX	Other Linux	10241	MB	UNSPECIFIED	ESX	Linux2.vmdk	10240
2	☐	Linux_2Nics (EVA)	VM_TEMPLATE	LINUX	Other Linux	10241	MB	UNSPECIFIED	ESX	teste.vmdk	10240
3	☐	Windows 7 (CIFM02)	VM_TEMPLATE	WINDOWS	Microsoft Windows 7	24577	MB	UNSPECIFIED	ESX	Windows 7 - copy.vmdk	24576
4	☐	Windows7_2Nics (EVA)	VM_TEMPLATE	WINDOWS	Microsoft Windows 7	24577	MB	UNSPECIFIED	ESX	Windows7.vmdk	24576

✓ *Software Detals*

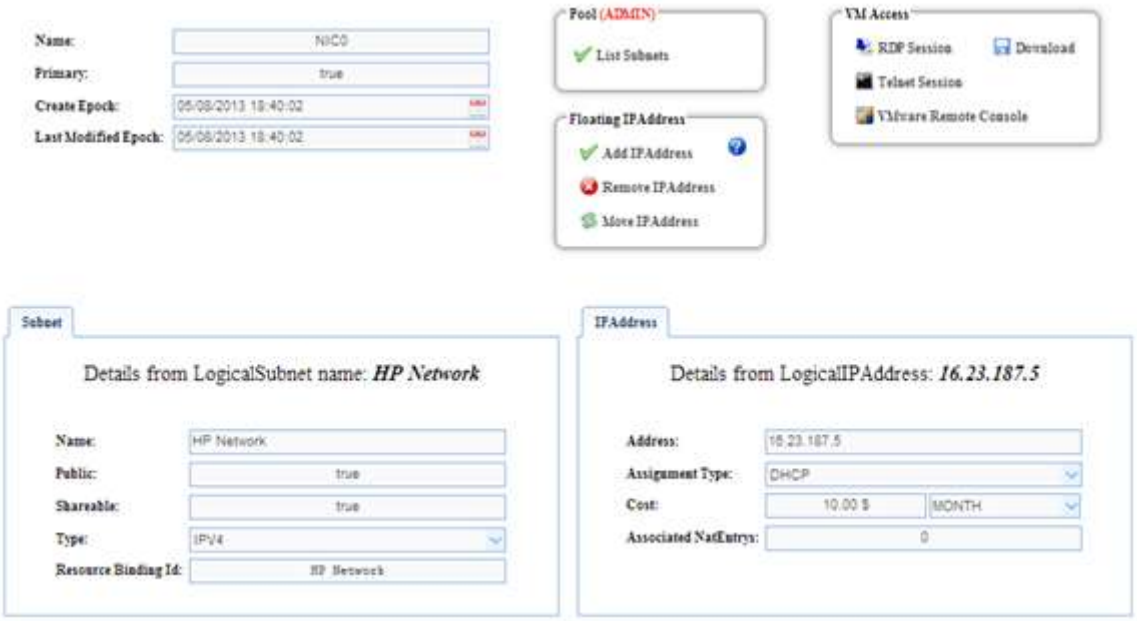
Figura 195 - Detalhes do Software



P.7 Interface

P.7.1 Informação

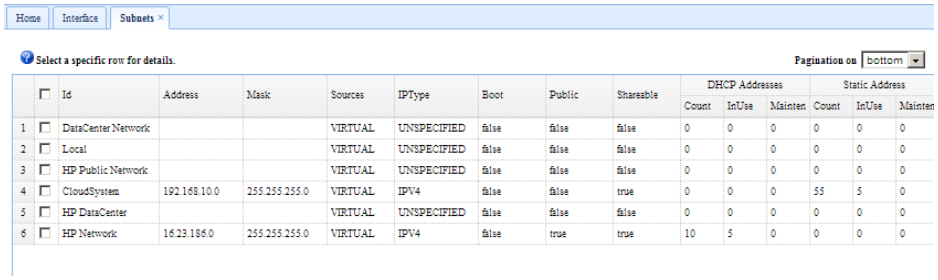
Figura 196 - Detalhes do Interface NIC0" do Servidor "SvrGrp1-1"



P.7.2 Execução

✓ *List Subnets*

Figura 197 - Listagem de todas as redes



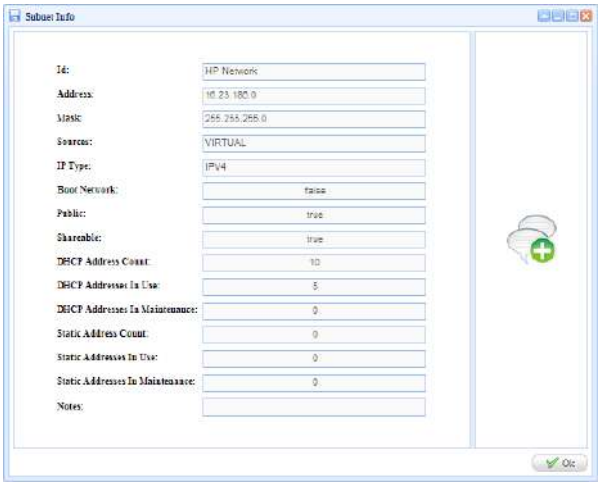
Home | Interface | Subnets

Select a specific row for details.

Pagination on bottom

		Id	Address	Mask	Sources	IPType	Boot	Public	Shareable	DHCP Addresses			Static Address		
										Count	InUse	Mainten	Count	InUse	Mainten
1	☐	DataCenter Network			VIRTUAL	UNSPECIFIED	false	false	false	0	0	0	0	0	0
2	☐	Local			VIRTUAL	UNSPECIFIED	false	false	false	0	0	0	0	0	0
3	☐	HP Public Network			VIRTUAL	UNSPECIFIED	false	false	false	0	0	0	0	0	0
4	☐	CloudSystem	192.168.10.0	255.255.255.0	VIRTUAL	IPV4	false	false	true	0	0	0	55	5	0
5	☐	HP DataCenter			VIRTUAL	UNSPECIFIED	false	false	false	0	0	0	0	0	0
6	☐	HP Network	16.23.186.0	255.255.255.0	VIRTUAL	IPV4	false	true	true	10	5	0	0	0	0

Figura 198 - Detalhe da rede



Subnet Info

Id:	HP Network
Address:	16.23.186.0
Mask:	255.255.255.0
Sources:	VIRTUAL
IP Type:	IPV4
Boot Network:	false
Public:	true
Shareable:	true
DHCP Address Count:	10
DHCP Addresses In Use:	5
DHCP Addresses In Maintenance:	0
Static Address Count:	0
Static Addresses In Use:	0
Static Addresses In Maintenance:	0
Notes:	

Ok

✓ *Add IPAddress*

Figura 199 - Adição de um IP flutuante à Interface “NIC0”

Add IP Address

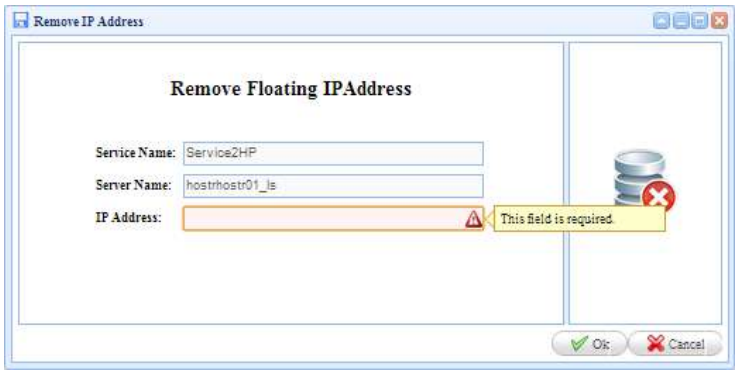
Add Floating IPAddress

Service Name:	Service2HP
Server Name:	hostrhostr01_is
NIC Name:	NIC0

Ok Cancel

✓ *Remove IPAddress*

Figura 200 - Remoção de um IP flutuante pertencente à Interface “NIC0”



✓ *Move IPAddress*

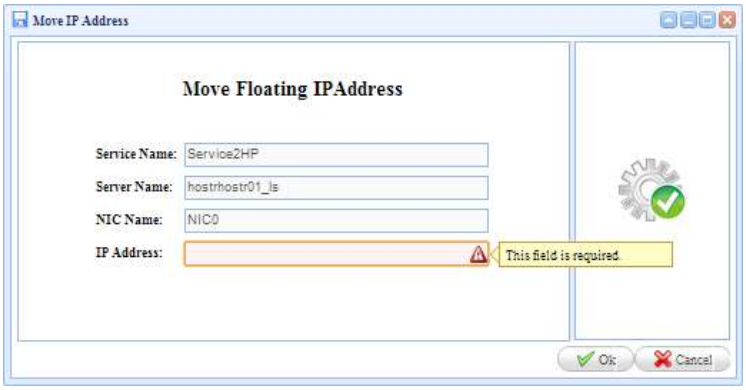



Figura 201 - Mover um IP flutuante para a Interface “NIC0”