

INSTITUTO SUPERIOR DE TECNOLOGIAS AVANÇADAS



VIRTUALIZAÇÃO E CLOUD COMPUTING

VIRTUALIZAÇÃO DE SERVIDORES E BIBLIOTECA VIRTUAL

LICENCIATURA INFORMÁTICA - LISBOA

2015/2016

Gonçalo Pimenta de Abreu

Nº 1946

Coordenador: Prof. Doutor. Pedro Brandão

Orientador: Prof. Doutor. Pedro Brandão



Dedicatória

Dedico este trabalho à minha mulher, pelo contínuo estímulo para continuar o caminho que iniciei em 2013 e pela extrema paciência e força que me deu para persistir perante os mais duros desafios.

Aos meus pais que me proporcionaram a minha formação e pelos valores transmitidos ao longo da minha vida, pelos princípios da humildade, trabalho e dedicação.

(Página intencionalmente deixada em branco)

Agradecimentos

Durante todo este meu percurso académico muitas foram as pessoas que me apoiaram e contribuíram para o meu sucesso. Por este motivo não posso deixar de agradecer a todos aqueles que me acompanharam ao longo destes últimos anos e permitiram que eu terminasse este Projeto Global da Licenciatura em Engenharia Informática.

Inicialmente queria agradecer ao meu orientador, Prof. Doutor Pedro Brandão e ao professor Eng. Pedro Crispim por me terem proporcionado as condições e os elementos chave que permitiram a realização do laboratório.

Um agradecimento especial ao colega de Licenciatura, Nuno Miguel Mendes, que deu uma ajuda imprescindível na programação e no desenvolvimento da aplicação.

Quero agradecer a todos os meus amigos e colegas desta Licenciatura, com eles a vida académica tornou-se muito mais interessante. Entre todos eles existem pessoas que especialmente se destacam nomeadamente, Ricardo Nunes, Duarte Bizarra, Ricardo Morais, Nuno Azevedo e Tiago Soveral - pelos 3 anos passados, em que me ajudaram bastante desde o primeiro dia de faculdade e prestaram uma grande ajuda na realização deste projeto.

Agradeço a todos os professores do ISTECS que contribuíram em muito para o aumento dos meus conhecimentos e competências.

Para último ficaram as pessoas mais importantes na minha vida, a minha família. Obviamente, tenho que destacar a minha mulher e os meus pais, obrigado por todos os sacrifícios feitos e por toda a dedicação para que eu pudesse chegar onde cheguei. Sempre me deram apoio e confiança e este projeto é a minha mostra de trabalho e retribuição pelos momentos difíceis passados para que eu pudesse iniciar este curso e terminá-lo.

Muito Obrigado!

(Página intencionalmente deixada em branco)

Resumo

Este documento descreve o trabalho realizado no âmbito da Unidade Curricular Projeto Global, na Licenciatura em Engenharia Informática do Instituto Superior de Tecnologias Avançadas de Lisboa.

Quando apareceram os primeiros sistemas de computação, as máquinas eram muito grandes e dispendiosas; eram projetadas para realizar um conjunto de operações, em que rapidamente vários sectores da sociedade passaram a desenvolver aplicações específicas para cada área e assim aproveitar o poder computacional. Devido à evolução tecnológica, passou a ser possível ter vários sistemas, simultaneamente, em funcionamento apenas numa máquina.

Atualmente, com a expansão das tecnologias de informação, muito do conhecimento humano passou a estar registado em suportes digitais, o que nos remete à utilização de intermediários para a leitura dessa informação: *hardware* e *software*. Na última década imperou o uso de bases de dados relacionais. Estas bases de dados desempenham um papel relevante no contexto global da informação digital. Os objetos digitais são estruturas complexas que implicam o uso de *hardware* e *software* específico para que os seres humanos os possam interpretar e compreender.

Este trabalho procura, de maneira concisa, expor o conceito de virtualização desde o seu desenvolvimento até à importância que assumiu nos dias de hoje, sendo explicada a sua arquitetura e técnicas existentes, bem como as tecnologias de implementação de virtualização. É também abordado o tema dos sistemas de bases de dados relacionais com implementação em SQL e o modo como os seus conteúdos podem ser disponibilizados numa aplicação através do ADO.NET.

Por fim, apresenta-se a implementação de uma solução de uma Biblioteca Virtual, que melhor se enquadra nos requisitos definidos no projeto, para a sua elaboração em ambiente de laboratório.

Palavras chave: Virtualização, SQL, ADO.NET, Biblioteca Virtual, SCVMM, Hyper-V, *Hypervisor*

(Página intencionalmente deixada em branco)

Abstract

This document describes the work carried out under the Curricular Unit Global Project in the Computer Engineering Degree of the Advanced Technology Institute of Lisbon.

When appeared the first computer systems, the machines were too large and expensive; they were designed to perform a number of operations in which various sectors of society quickly began to develop specific applications for each area and so take advantage of the computing power. Due to technological developments, it is now possible to have multiple systems simultaneously, operating only in a machine.

Currently, with the expansion of information technology, much of human knowledge has to be recorded in digital media, which brings us to the use of intermediaries to read this information: hardware and software. In the last decade ruled the use of relational databases. These databases play an important role in the global context of digital information. The digital objects are complex structures involving the use of specific hardware and software so that human beings can interpret and understand them.

This project seeks, in concise form expose the concept of virtualization since its development to the importance it has assumed nowadays, being explained their existing architecture and techniques as well as the implementation of virtualization technologies. It also addressed the subject of relational database systems implementation in SQL and how their content can be made available in an application through ADO.NET.

Finally, the implementation presents a solution to a Virtual Library that best fits the requirements defined in the project, for its preparation in the laboratory environment.

Key Words: Virtualization, SQL, ADO.NET, Virtual Library, SCVMM, Hyper-V, *Hypervisor*

(Página intencionalmente deixada em branco)

Abreviaturas e Símbolos

ADO	ActiveX Data Objects
AMD-V	AMD Virtualization
ANSI	American National Standards Institute
BD	Base de Dados
BIOS	Basic Input/Output System
CPU	Central Processing Unit
CSV	Comma Separated Value
DLL	Dynamic-Link Library
IBM	International Business Machines
INTEL-V	INTEL Virtualization
ISO	International Organization for Standardization
NoSQL	Not Only SQL
ODBC	Open Database Connectivity
OLE	Object Linking and Embedding
RAM	Random Access Memory
SGBD	Sistema de Base de Dados
SGBDR	Sistema de Base de Dados Relacional
SO	Sistema Operativo
SQL	Structured Query Language
TI	Tecnologias de Informação
UEFI	Unified Extensible Firmware Interface
VID	Virtual Infrastructure Driver
VMBus	Virtual Machine Bus
VMM	Virtual Machine Monitor
VSP	Virtualization Service Provider
WMI	Windows Management Instrumentation
WPF	Windows Presentation Foundation
XML	Extensible Markup Language

(Página intencionalmente deixada em branco)

Índice

Dedicatória	III
Agradecimentos	V
Resumo	VII
Abstract	IX
Abreviaturas e Símbolos	XI
Índice.....	XIII
Índice de Figuras.....	XVII
Índice de Tabelas	XXXI
1 Introdução	33
1.1 Enquadramento.....	33
1.2 Motivações	34
1.3 Objetivos	34
1.4 Estrutura do Documento.....	35
2 Estado de Arte.....	37
2.1 Virtualização	37
2.1.1 Classificação das Técnicas de Virtualização.....	41
2.2 Plataforma de Monitor de Máquinas Virtuais - <i>Hypervisor</i>	44
2.2.1 Hyper-V.....	46
2.3 Sistemas de Base de Dados	49
2.3.1 SQL	53
2.4 <i>Framework</i> .NET.....	54
2.4.1 ADO.NET	55
3 Contextualização.....	59
3.1 Tecnologia:	59
3.2 Base de Dados	60
3.3 Interface de acesso à BD	61
4 Implementação da Solução	63
4.1 Máquina Física	63
4.1.1 Configuração da Máquina Física.....	64

4.2	Servidores Virtuais	64
4.2.1	Visão Geral da Solução	64
4.2.2	Criação dos Servidores Virtuais	65
4.2.2.1	Servidor Controlador de Domínio	65
4.2.2.2	Servidor SQL.....	72
4.2.2.3	Servidor Virtual Machine Manager.....	78
4.2.3	Configuração dos Servidores Virtuais.....	83
4.2.3.1	Configuração das Interfaces de Rede	83
4.2.3.2	Controlador de Domínio.....	89
4.2.3.3	Servidor SQL.....	122
4.2.3.4	Servidor Virtual Machine Manager.....	129
4.3	Aplicação Biblioteca Virtual	181
4.3.1	Criar a Base de Dados da Aplicação	181
4.3.1.1	Criar Tabelas na Base de Dados.....	182
4.3.1.2	Diagrama e Correlação de Tabelas da Base de Dados	186
4.3.2	Autenticação na Aplicação através da <i>Active Directory</i> (AD DS).....	187
4.3.3	Estabelecer Ligação da Aplicação à Base de Dados	192
4.3.4	<i>Layout</i> das Interfaces Gráficas	194
4.3.4.1	Interface Gráfica Principal	194
4.3.4.2	Interface Gráfica Adicionar e Editar	200
4.3.5	Funcionalidades da Aplicação.....	208
4.3.5.1	Campo de Pesquisa.....	208
4.3.5.2	Eliminar Conteúdos nas Tabelas	214
4.3.5.3	Editar e Adicionar Conteúdos nas Tabelas	218
4.3.5.4	<i>ComboBox</i>	226
4.3.5.5	<i>Upload</i> e <i>Download</i> de Documentos em PDF	228
4.3.5.6	Validação de Campos Obrigatórios.....	236

4.3.5.7	Validação do Campo E-mail	238
4.3.5.8	Encriptação	239
4.3.5.9	Informações para o Utilizador na Interface Gráfica Principal	240
4.3.5.10	Criar Ficheiro de Instalação da Aplicação	243
5	Conclusão.....	247
	Bibliografia	249

(Página intencionalmente deixada em branco)

Índice de Figuras

Figura 1: Virtualização de Servidores, Fonte (Mombrea, 2013)	38
Figura 2: Virtualização de Desktop, Fonte (“Virtualization,” 2012).....	39
Figura 3: Virtualização de Aplicações, Fonte (Qureshi, 2007)	40
Figura 4: Virtualização de Redes, Fonte (Oswalt, 2013).....	41
Figura 5: Virtualização Total, Fonte (Ribeiro, 2009)	42
Figura 6: Paravirtualização, Fonte (Ribeiro, 2009)	43
Figura 7: Virtualização Assistida por Hardware, Fonte (Ribeiro, 2009).....	44
Figura 8: Hypervisor Tipo 1, Fonte (Jason & Basrai, 2015)	45
Figura 9: Hypervisor Tipo 2, Fonte (Jason & Basrai, 2015)	46
Figura 10: Arquitetura Hyper-V, Fonte (Microsoft Corporation, n.d.-d)	47
Figura 11: Plataforma de Gestão do Hyper-V, Fonte (Davis, 2013)	48
Figura 12: Arquitetura Framework .NET, Fonte (Microsoft Corporation, 2005)	54
Figura 13: Exemplo de um acesso a base de dados com ADO.NET, Fonte (Meier, Vasireddy, Babbar, & Mackman, 2004)	57
Figura 14: Características da máquina física, Fonte própria.....	63
Figura 15: VMware Workstation 12 Pro, Fonte própria.....	64
Figura 16: Estrutura da Rede Virtual, Fonte própria	65
Figura 17: VMware criação de nova máquina virtual, Fonte própria.....	65
Figura 18: Assistente de criação de máquinas virtuais VMware, Início, Fonte própria	66
Figura 19: Assistente de criação de máquinas virtuais VMware, SO, Fonte própria	66
Figura 20: Assistente de criação de máquinas virtuais VMware, Licenciamento, Fonte própria	67
Figura 21: Assistente de criação de máquinas virtuais VMware, Nome do servidor virtual, Fonte própria	67
Figura 22: Assistente de criação de máquinas virtuais VMware, Processador, Fonte próprio.....	68
Figura 23: Assistente de criação de máquinas virtuais VMware, Memória, Fonte própria.....	68
Figura 24: Assistente de criação de máquinas virtuais VMware, Interface de Rede, Fonte própria	69
Figura 25: Assistente de criação de máquinas virtuais VMware, Espaço de Armazenamento, Fonte própria	70

Figura 26: Assistente de criação de máquinas virtuais VMware, Resumo de Instalação, Fonte própria.....	70
Figura 27: VMware configuração de <i>hardware</i> , Interface de Rede 1, Fonte própria.....	71
Figura 28: VMware configuração de <i>hardware</i> , Interface de Rede 2, Fonte própria.....	71
Figura 29: VMware configuração de <i>hardware</i> , Interface de Rede 3, Fonte própria.....	72
Figura 30: VMware criação de nova máquina virtual 2, Fonte própria.....	72
Figura 31: Assistente de criação de máquinas virtuais VMware, Início 2, Fonte própria.....	73
Figura 32: Assistente de criação de máquinas virtuais VMware, SO 2, Fonte própria.....	73
Figura 33: Assistente de criação de máquinas virtuais VMware, Nome do servidor virtual 2, Fonte própria.....	74
Figura 34: Assistente de criação de máquinas virtuais VMware, Processador 2, Fonte próprio..	75
Figura 35: Assistente de criação de máquinas virtuais VMware, Memória 2, Fonte própria.....	75
Figura 36: Assistente de criação de máquinas virtuais VMware, Interface de Rede 2, Fonte própria.....	76
Figura 37: Assistente de criação de máquinas virtuais VMware, Espaço de Armazenamento 2, Fonte própria.....	77
Figura 38: Assistente de criação de máquinas virtuais VMware, Resumo de Instalação 2, Fonte própria.....	77
Figura 39: Assistente de criação de máquinas virtuais VMware, Nome do servidor virtual 3, Fonte própria.....	78
Figura 40: Assistente de criação de máquinas virtuais VMware, Processador 3, Fonte próprio..	79
Figura 41: Assistente de criação de máquinas virtuais VMware, Memória 3, Fonte própria.....	79
Figura 42: Assistente de criação de máquinas virtuais VMware, Resumo de Instalação 3, Fonte própria.....	80
Figura 43: VMware configuração de <i>hardware</i> , Processador, Fonte própria.....	81
Figura 44: Editar ficheiro (*.vmx), Fonte Própria.....	81
Figura 45: Editar ficheiro (*.vmx) 2, Fonte Própria.....	82
Figura 46: Abrir janela dos Interface de Rede, Fonte Própria.....	84
Figura 47: Janela dos Interfaces de Rede, Fonte Própria.....	84
Figura 48: Renomear Interfaces de Rede, Fonte Própria.....	84
Figura 49: Renomear Interfaces de Rede 2, Fonte Própria.....	85

Figura 50: Alterar Hierarquia dos Interfaces de Rede, Fonte Própria	85
Figura 51: Alterar Hierarquia dos Interfaces de Rede 2, Fonte Própria	86
Figura 52: Propriedades do Interface de Rede, Fonte Própria	86
Figura 53: Propriedades do Interface de Rede 2, Fonte Própria.....	87
Figura 54: Alterar Endereçamento IP do Interface de Rede, Fonte Própria	87
Figura 55: Alterar Endereçamento IP do Interface de Rede 2, Fonte Própria	88
Figura 56: Alterar Endereçamento IP do Interface de Rede 3, Fonte Própria	88
Figura 57: <i>Server Manager</i> , Fonte Própria.....	89
Figura 58: Altear Nome do Servidor, Fonte Própria.....	90
Figura 59: Altear Nome do Servidor 2, Fonte Própria.....	90
Figura 60: <i>Server Manager</i> 1, “Add Roles and Features”, Fonte Própria	91
Figura 61: Assistente de configuração “Add Roles and Features”, <i>Active Directory</i> 1, Fonte Própria	91
Figura 62: Assistente de configuração “Add Roles and Features”, <i>Active Directory</i> 2, Fonte Própria	92
Figura 63: Assistente de configuração “Add Roles and Features”, <i>Active Directory</i> 3, Fonte Própria	92
Figura 64: Assistente de configuração “Add Roles and Features”, <i>Active Directory</i> 4, Fonte Própria	93
Figura 65: Assistente de configuração “Add Roles and Features”, <i>Active Directory</i> 5, Fonte Própria	93
Figura 66: Promover servidor a controlador de domínio, Fonte Própria.....	94
Figura 67: Assistente de configuração do “Active Directory and Domain Services”, Fonte Própria	95
Figura 68: Assistente de configuração do “Active Directory and Domain Services” 2, Fonte Própria.....	95
Figura 69: Assistente de configuração do “Active Directory and Domain Services” 3, Fonte Própria.....	96
Figura 70: <i>Server Manager</i> 2, Fonte Própria	96
Figura 71: Login no Controlador de Domínio, Fonte Própria	97
Figura 72: <i>Server Manager</i> 2, “Add Roles and Features”, Fonte Própria	97

Figura 73: Assistente de configuração “Add Roles and Features”, DHCP Server 1, Fonte Própria	98
Figura 74: Assistente de configuração “Add Roles and Features”, DHCP Server 2, Fonte Própria	98
Figura 75: <i>Server Manager</i> , <i>Tools</i> DHCP, Fonte Própria	99
Figura 76: Windows DHCP, Adicionar nova range de IPs, Fonte Própria.....	100
Figura 77: Assistente de configuração “New Scope”, “IP Address Range”, Fonte Própria.....	100
Figura 78: Assistente de configuração “New Scope”, “Lease Duration”, Fonte Própria	101
Figura 79: Assistente de configuração “New Scope”, “Domain Name and DNS Servers”, Fonte Própria.....	101
Figura 80: Windows DHCP, Adicionada nova range de IPs, Fonte Própria	102
Figura 81: <i>Server Manager</i> , “Active Directory Users and Computers”, Fonte Própria	103
Figura 82: Criar nova Unidade Organizacional 1, Fonte Própria	103
Figura 83: Criar nova Unidade Organizacional 2, Fonte Própria	104
Figura 84: Unidades Organizacionais do Laboratório, Fonte Própria	104
Figura 85: Alterar contentor padrão de novas máquinas no domínio, Fonte Própria	105
Figura 86: Criar novo Grupo de Segurança 1, Fonte Própria	105
Figura 87: Criar novo Grupo de Segurança 2, Fonte Própria	106
Figura 88: Criar nova conta de utilizador 1, Fonte Própria	107
Figura 89: Criar nova conta de utilizador 2, Fonte Própria	107
Figura 90: Adicionar conta ao Grupo de Segurança 1, Fonte Própria	108
Figura 91: Adicionar conta ao Grupo de Segurança 2, Fonte Própria	108
Figura 92: Contas do Grupo de Segurança “SCVMMAdmins”, Fonte Própria	109
Figura 93: Contas do Grupo de Segurança “AppAdmin”, Fonte Própria.....	109
Figura 94: Contas do Grupo de Segurança “AppUser”, Fonte Própria	110
Figura 95: Conteúdo Unidade Organizacional “AppUsers”, Fonte Própria	110
Figura 96: Conteúdo Unidade Organizacional “SCVMM Accounts”, Fonte Própria.....	111
Figura 97: Conteúdo Unidade Organizacional “Servers”, Fonte Própria	111
Figura 98: Conteúdo Unidade Organizacional “SQL Server Service Accounts”, Fonte Própria	111
Figura 99: Conteúdo Unidade Organizacional “Workstations”, Fonte Própria.....	112
Figura 100: <i>Server Manager</i> , “Group Policy Management”, Fonte Própria	113

Figura 101: Criar <i>Group Policy Object</i> 1, Fonte Própria.....	113
Figura 102: Criar <i>Group Policy Object</i> 2, Fonte Própria.....	114
Figura 103: Editar <i>Group Policy Object</i> , Fonte Própria	114
Figura 104: Configurar <i>Group Policy Object</i> 1, Instalação de Aplicações, Fonte Própria	115
Figura 105: Configurar <i>Group Policy Object</i> 2, Instalação de Aplicações, Fonte Própria	115
Figura 106: Configurar <i>Group Policy Object</i> 3, Instalação de Aplicações, Fonte Própria	116
Figura 107: Configurar <i>Group Policy Object</i> 4, Instalação de Aplicações, Fonte Própria	116
Figura 108: Configurar <i>Group Policy Object</i> 1, <i>Wallpaper</i> , Fonte Própria	117
Figura 109: Configurar <i>Group Policy Object</i> 2, <i>Wallpaper</i> , Fonte Própria	117
Figura 110: Configurar <i>Group Policy Object</i> 1, <i>Firewall</i> , Fonte Própria	118
Figura 111: Configurar <i>Group Policy Object</i> 2, <i>Firewall</i> , Fonte Própria	119
Figura 112: Configurar <i>Group Policy Object</i> 2, <i>Firewall</i> , Fonte Própria	119
Figura 113: Configurar <i>Group Policy Object</i> 3, <i>Firewall</i> , Fonte Própria	120
Figura 114: Configurar <i>Group Policy Object</i> 1, <i>File</i> , Fonte Própria.....	120
Figura 115: Configurar <i>Group Policy Object</i> 2, <i>File</i> , Fonte Própria.....	121
Figura 116: Configurar <i>Group Policy Object</i> 3, <i>File</i> , Fonte Própria.....	121
Figura 117: Resumo de todas as <i>Group Policy Object</i> configuradas, Fonte Própria	122
Figura 118: VMware configuração de <i>hardware</i> , Adicionar Imagem, Fonte própria.....	123
Figura 119: “This PC”, SQLPG1946, Fonte Própria.....	123
Figura 120: Assistente de Instalação 1, SQL Server 2012, Fonte Própria.....	124
Figura 121: Assistente de Instalação 2, SQL Server 2012, Fonte Própria.....	124
Figura 122: Assistente de Instalação 3, SQL Server 2012, Fonte Própria.....	125
Figura 123: Assistente de Instalação 4, SQL Server 2012, Fonte Própria.....	125
Figura 124: Assistente de Instalação 5, SQL Server 2012, Fonte Própria.....	126
Figura 125: Assistente de Instalação 6, SQL Server 2012, Fonte Própria.....	126
Figura 126: <i>Server Manager</i> , “Windows Firewall with Advanced Security”, Fonte Própria	127
Figura 127: Windows Firewall 2, “Inbound Rules”, Fonte Própria	127
Figura 128: Assistente de Configuração Nova Regra da Firewall 1, Fonte Própria.....	128
Figura 129: Assistente de Configuração Nova Regra da Firewall 2, Fonte Própria.....	128
Figura 130: Assistente de Configuração Nova Regra da Firewall 3, Fonte Própria.....	129
Figura 131: Windows Firewall 2, “Inbound Rules”, Fonte Própria	129

Figura 132: <i>Server Manager</i> 3, “Add Roles and Features”, Fonte Própria	130
Figura 133: Assistente de configuração “Add Roles and Features”, Hyper-V 1, Fonte Própria	130
Figura 134: Assistente de configuração “Add Roles and Features”, Hyper-V 2, Fonte Própria	131
Figura 135: Assistente de configuração “Add Roles and Features”, Hyper-V 3, Fonte Própria	131
Figura 136: Assistente de configuração “Add Roles and Features”, Hyper-V 4, Fonte Própria	132
Figura 137: Assistente de configuração “Add Roles and Features”, Hyper-V 5, Fonte Própria	132
Figura 138: Assistente de configuração “Add Roles and Features”, Hyper-V 6, Fonte Própria	133
Figura 139: Assistente de configuração “Add Roles and Features”, Hyper-V 7, Fonte Própria	133
Figura 140: Windows Assessment and Deployment Kits (ADK), Fonte Própria	134
Figura 141: Instalação do Windows ADK 1, Windows 8.1, Fonte Própria.....	135
Figura 142: Instalação do Windows ADK 2, Windows 8.1, Fonte Própria.....	135
Figura 143: Instalação Microsoft SQL Server 2012 Native Client 1, Fonte Própria.....	136
Figura 144: Instalação Microsoft SQL Server 2012 Native Client 2, Fonte Própria.....	136
Figura 145: Instalação Microsoft SQL Server 2012 Command Line, Fonte Própria	137
Figura 146: <i>Server Manager</i> , ADSI Edit, Fonte Própria	137
Figura 147: ADSI Edit 1, Criar Contentor, Fonte Própria.....	138
Figura 148: ADSI Edit 2, Criar Contentor, Fonte Própria.....	138
Figura 149: ADSI Edit 3, Criar Contentor, Fonte Própria.....	139
Figura 150: ADSI Edit 4, Criar Contentor, Fonte Própria.....	139
Figura 151: ADSI Edit 4, Criar Contentor, Fonte Própria.....	140
Figura 152: ADSI Edit 4, Criar Contentor, Fonte Própria.....	140
Figura 153: ADSI Edit 5, Criar Contentor, Fonte Própria.....	141
Figura 154: ADSI Edit 6, Criar Contentor, Fonte Própria.....	141
Figura 155: ADSI Edit 7, Criar Contentor, Fonte Própria.....	142
Figura 156: ADSI Edit 8, Criar Contentor, Fonte Própria.....	142
Figura 157: ADSI Edit 9, Criar Contentor, Fonte Própria.....	143
Figura 158: ADSI Edit 10, Criar Contentor, Fonte Própria.....	143
Figura 159: Instalação do SCVMM 2012 R2 1, Fonte Própria	144
Figura 160: Instalação do SCVMM 2012 R2 2, Fonte Própria	145
Figura 161: Instalação do SCVMM 2012 R2 3, Fonte Própria	145
Figura 162: Instalação do SCVMM 2012 R2 4, Fonte Própria	146

Figura 163: Instalação do SCVMM 2012 R2 5, Fonte Própria	147
Figura 164: Instalação do SCVMM 2012 R2 6, Fonte Própria	148
Figura 165: Instalação do SCVMM 2012 R2 7, Fonte Própria	148
Figura 166: Instalação do SCVMM 2012 R2 8, Fonte Própria	149
Figura 167: Instalação do SCVMM 2012 R2 9, Fonte Própria	149
Figura 168: Adicionar <i>host</i> Hyper-V no SCVMM 1, Fonte Própria	150
Figura 169: Adicionar <i>host</i> Hyper-V no SCVMM 2, Fonte Própria	150
Figura 170: Adicionar <i>host</i> Hyper-V no SCVMM 3, Fonte Própria	151
Figura 171: Adicionar <i>host</i> Hyper-V no SCVMM 4, Fonte Própria	152
Figura 172: Adicionar <i>host</i> Hyper-V no SCVMM 5, Fonte Própria	152
Figura 173: Adicionar <i>host</i> Hyper-V no SCVMM 6, Fonte Própria	153
Figura 174: Adicionar <i>host</i> Hyper-V no SCVMM 7, Fonte Própria	153
Figura 175: Adicionar <i>host</i> Hyper-V no SCVMM 8, Fonte Própria	154
Figura 176: Adicionar <i>host</i> Hyper-V no SCVMM 9, Fonte Própria	154
Figura 177: Adicionar <i>host</i> Hyper-V no SCVMM 10, Fonte Própria	155
Figura 178: Adicionar <i>host</i> Hyper-V no SCVMM 11, Fonte Própria	155
Figura 179: Configuração Rede do <i>Hypervisor</i> 1, Fonte Própria	156
Figura 180: Configuração Rede do <i>Hypervisor</i> 2, Fonte Própria	157
Figura 181: Configuração Rede do <i>Hypervisor</i> 3, Fonte Própria	157
Figura 182: Configuração Rede do <i>Hypervisor</i> 4, Fonte Própria	158
Figura 183: Organizar Biblioteca do SCVMM 1, Fonte Própria.....	159
Figura 184: Organizar Biblioteca do SCVMM 2, Fonte Própria.....	159
Figura 185: Organizar Biblioteca do SCVMM 3, Fonte Própria.....	159
Figura 186: Organizar Biblioteca do SCVMM 4, Fonte Própria.....	160
Figura 187: Criar Máquina Virtual para <i>Template</i> 1, Fonte Própria.....	161
Figura 188: Criar Máquina Virtual para <i>Template</i> 2, Fonte Própria.....	162
Figura 189: Criar Máquina Virtual para <i>Template</i> 3, Fonte Própria.....	162
Figura 190: Criar Máquina Virtual para <i>Template</i> 4, Fonte Própria.....	163
Figura 191: Criar Máquina Virtual para <i>Template</i> 5, Fonte Própria.....	163
Figura 192: Criar Máquina Virtual para <i>Template</i> 6, Fonte Própria.....	164
Figura 193: Criar Máquina Virtual para <i>Template</i> 7, Fonte Própria.....	164

Figura 194: Criar Máquina Virtual para <i>Template</i> 8, Fonte Própria.....	165
Figura 195: Criar Máquina Virtual para <i>Template</i> 9, Fonte Própria.....	165
Figura 196: Criar Máquina Virtual para <i>Template</i> 10, Fonte Própria.....	166
Figura 197: Criar Máquina Virtual para <i>Template</i> 11, Fonte Própria.....	166
Figura 198: Criar Máquina Virtual para <i>Template</i> 12, Fonte Própria.....	167
Figura 199: Criar Máquina Virtual para <i>Template</i> 13, Fonte Própria.....	167
Figura 200: Criar Máquina Virtual para <i>Template</i> 14, Fonte Própria.....	168
Figura 201: Criar Máquina Virtual para <i>Template</i> 15, Fonte Própria.....	168
Figura 202: Criar Máquina Virtual para <i>Template</i> 16, Fonte Própria.....	169
Figura 203: Criar um <i>Template</i> para uma Máquina Virtual 1, Fonte Própria.....	170
Figura 204: Criar um <i>Template</i> para uma Máquina Virtual 2, Fonte Própria.....	171
Figura 205: Criar um <i>Template</i> para uma Máquina Virtual 3, Fonte Própria.....	171
Figura 206: Criar um <i>Template</i> para uma Máquina Virtual 4, Fonte Própria.....	172
Figura 207: Criar um <i>Template</i> para uma Máquina Virtual 5, Fonte Própria.....	172
Figura 208: Criar um <i>Template</i> para uma Máquina Virtual 6, Fonte Própria.....	173
Figura 209: Criar um <i>Template</i> para uma Máquina Virtual 7, Fonte Própria.....	173
Figura 210: Criar um <i>Template</i> para uma Máquina Virtual 8, Fonte Própria.....	174
Figura 211: Criar um <i>Template</i> para uma Máquina Virtual 9, Fonte Própria.....	174
Figura 212: Criar um <i>Template</i> para uma Máquina Virtual 10, Fonte Própria.....	175
Figura 213: Criar um <i>Template</i> para uma Máquina Virtual 11, Fonte Própria.....	175
Figura 214: Criar Máquina Virtual a partir de um <i>Template</i> 1, Fonte Própria.....	176
Figura 215: Criar Máquina Virtual a partir de um <i>Template</i> 2, Fonte Própria.....	177
Figura 216: Criar Máquina Virtual a partir de um <i>Template</i> 3, Fonte Própria.....	177
Figura 217: Criar Máquina Virtual a partir de um <i>Template</i> 4, Fonte Própria.....	178
Figura 218: Criar Máquina Virtual a partir de um <i>Template</i> 5, Fonte Própria.....	178
Figura 219: Criar Máquina Virtual a partir de um <i>Template</i> 6, Fonte Própria.....	179
Figura 220: Criar Máquina Virtual a partir de um <i>Template</i> 7, Fonte Própria.....	179
Figura 221: <i>Desktop</i> da Máquina Cliente criada no SCVMM, Fonte Própria.....	180
Figura 222: Criar Base de Dados 1, Fonte Própria.....	181
Figura 223: Criar Base de Dados 2, Fonte Própria.....	182
Figura 224: Criar Base de Dados 3, Fonte Própria.....	182

Figura 225: Criar Tabelas na Base de Dados, Fonte Própria	183
Figura 226: Criar Tabela Documentos, Fonte Própria.....	184
Figura 227: Chaves Estrangeiras, Tabela Documentos, Fonte Própria	184
Figura 228: Criar Tabela Editoras, Fonte Própria.....	184
Figura 229: Criar Tabela Autores, Fonte Própria	185
Figura 230: Criar Tabela Tipos de Documentos, Fonte Própria.....	185
Figura 231: Resumo das Tabelas criadas, Fonte Própria.....	186
Figura 232: Diagrama das Tabelas da Base de Dados, Fonte Própria.....	186
Figura 233: Autenticação na Aplicação através da AD 1, Fonte Própria	187
Figura 234: Interface Gráfica Inicial 1, Fonte Própria.....	187
Figura 235: Autenticação na Aplicação através da AD 2, Fonte Própria	187
Figura 236: Autenticação na Aplicação através da AD 3, Fonte Própria	188
Figura 237: Autenticação na Aplicação através da AD 4, Fonte Própria	189
Figura 238: Autenticação na Aplicação através da AD 5, Fonte Própria	189
Figura 239: Autenticação na Aplicação através da AD 6, Fonte Própria	190
Figura 240: Autenticação na Aplicação através da AD 7, Fonte Própria	190
Figura 241: Autenticação na Aplicação através da AD 8, Fonte Própria	191
Figura 242: Autenticação na Aplicação através da AD 9, Fonte Própria	191
Figura 243: Autenticação na Aplicação através da AD 10, Fonte Própria	191
Figura 244: Interface Gráfica Inicial 2, Fonte Própria.....	192
Figura 245: Estabelecer Ligação com a Base de Dados 1, Fonte Própria	192
Figura 246: Estabelecer Ligação com a Base de Dados 2, Fonte Própria	193
Figura 247: Estabelecer Ligação com a Base de Dados 3, Fonte Própria	193
Figura 248: Estabelecer Ligação com a Base de Dados 4, Fonte Própria	194
Figura 249: Estabelecer Ligação com a Base de Dados 5, Fonte Própria	194
Figura 250: <i>Layout</i> Interface Gráfica Principal 1, Fonte Própria	195
Figura 251: <i>Layout</i> Interface Gráfica Principal 2, Fonte Própria	195
Figura 252: <i>Layout</i> Interface Gráfica Principal 3, Fonte Própria	196
Figura 253: <i>Layout</i> Interface Gráfica Principal 4, Fonte Própria	196
Figura 254: <i>Layout</i> Interface Gráfica Principal 5, Fonte Própria	197
Figura 255: <i>Layout</i> Interface Gráfica Principal 6, Fonte Própria	197

Figura 256: <i>Layout</i> Interface Gráfica Principal 7, Fonte Própria	197
Figura 257: <i>Layout</i> Interface Gráfica Principal 8, Fonte Própria	198
Figura 258: <i>Layout</i> Interface Gráfica Principal 9, Fonte Própria	198
Figura 259: <i>Layout</i> Interface Gráfica Principal 10, Fonte Própria	199
Figura 260: <i>Layout</i> Interface Gráfica Principal 11, Fonte Própria	200
Figura 261: Interface Gráfica Adicionar/Editar Documentos 1, Fonte Própria.....	201
Figura 262: Interface Gráfica Editar Documentos 1, Fonte Própria.....	201
Figura 263: Interface Gráfica Editar Documentos 2, Fonte Própria.....	202
Figura 264: Interface Gráfica Editar Documentos 3, Fonte Própria.....	202
Figura 265: Interface Gráfica Adicionar Documentos 1, Fonte Própria.....	203
Figura 266: Interface Gráfica Adicionar Documentos 2, Fonte Própria.....	203
Figura 267: Interface Gráfica Adicionar/Editar Documentos 2, Fonte Própria.....	204
Figura 268: Interface Gráfica Adicionar/Editar Documentos 3, Fonte Própria.....	204
Figura 269: Interface Gráfica Adicionar/Editar Documentos 4, Fonte Própria.....	205
Figura 270: Interface Gráfica Adicionar/Editar Documentos 5, Fonte Própria.....	205
Figura 271: Interface Gráfica Adicionar Editoras, Fonte Própria.....	205
Figura 272: Interface Gráfica Editar Editoras, Fonte Própria.....	206
Figura 273: Interface Gráfica Adicionar Autor, Fonte Própria.....	206
Figura 274: Interface Gráfica Editar Autor, Fonte Própria.....	206
Figura 275: Interface Gráfica Adicionar Tipo de Documento, Fonte Própria	207
Figura 276: Interface Gráfica Editar Tipo de Documento, Fonte Própria	207
Figura 277: Campo Pesquisa 1, Fonte Própria.....	208
Figura 278: Campo Pesquisa 2, Fonte Própria.....	208
Figura 279: Campo Pesquisa 3, Fonte Própria.....	208
Figura 280: Campo Pesquisa 4, Fonte Própria.....	209
Figura 281: Campo Pesquisa 5, Fonte Própria.....	209
Figura 282: Campo Pesquisa 6, Fonte Própria.....	209
Figura 283: Campo Pesquisa 7, Fonte Própria.....	210
Figura 284: Campo Pesquisa 8, Fonte Própria.....	210
Figura 285: Campo Pesquisa 9, Fonte Própria.....	210
Figura 286: Campo Pesquisa 10, Fonte Própria.....	211

Figura 287: Campo Pesquisa 11, Fonte Própria.....	211
Figura 288: Campo Pesquisa 12, Fonte Própria.....	212
Figura 289: Campo Pesquisa 13, Fonte Própria.....	212
Figura 290: Campo Pesquisa 14, Fonte Própria.....	213
Figura 291: Campo Pesquisa 15, Fonte Própria.....	213
Figura 292: Eliminar Conteúdos da Tabela Documentos 1, Fonte Própria	214
Figura 293: Eliminar Conteúdos da Tabela Documentos 2, Fonte Própria	214
Figura 294: Eliminar Conteúdos da Tabela Documentos, Fonte Própria	215
Figura 295: Eliminar Conteúdos da Tabela Documentos 4, Fonte Própria	215
Figura 296: Eliminar Conteúdos da Tabela Documentos 5, Fonte Própria	215
Figura 297: Eliminar Conteúdos da Tabela Documentos 6, Fonte Própria	216
Figura 298: Eliminar Conteúdos da Tabela Editora, Autor e Tipos de Documento 1, Fonte Própria	216
Figura 299: Eliminar Conteúdos da Tabela Editora, Autor e Tipos de Documento 2, Fonte Própria	217
Figura 300: Eliminar Conteúdos da Tabela Editora, Autor e Tipos de Documento 3, Fonte Própria	217
Figura 301: Eliminar Conteúdos da Tabela Editora, Autor e Tipos de Documento 4, Fonte Própria	217
Figura 302: Eliminar Conteúdos da Tabela Editora, Autor e Tipos de Documento 5, Fonte Própria	218
Figura 303: Editar Conteúdos das Tabelas 1, Fonte Própria	218
Figura 304: Editar Conteúdos das Tabelas 2, Fonte Própria	219
Figura 305: Editar Conteúdos das Tabelas 3, Fonte Própria	219
Figura 306: Editar Conteúdos das Tabelas 4, Fonte Própria	220
Figura 307: Editar Conteúdos das Tabelas 5, Fonte Própria	220
Figura 308: Editar Conteúdos das Tabelas 6, Fonte Própria	220
Figura 309: Editar Conteúdos das Tabelas 7, Fonte Própria	221
Figura 310: Editar Conteúdos das Tabelas 8, Fonte Própria	222
Figura 311: Editar Conteúdos das Tabelas 9, Fonte Própria	222
Figura 312: Editar Conteúdos das Tabelas 10, Fonte Própria	223

Figura 313: Editar Conteúdos das Tabelas 11, Fonte Própria	223
Figura 314: Editar Conteúdos das Tabelas 12, Fonte Própria	223
Figura 315: Adicionar Conteúdos nas Tabelas 1, Fonte Própria	224
Figura 316: Adicionar Conteúdos nas Tabelas 2, Fonte Própria	224
Figura 317: Adicionar Conteúdos nas Tabelas 3, Fonte Própria	225
Figura 318: Adicionar Conteúdos nas Tabelas 4, Fonte Própria	226
Figura 319: ComboBox 1, Fonte Própria.....	227
Figura 320: ComboBox 2, Fonte Própria.....	227
Figura 321: ComboBox 3, Fonte Própria.....	228
Figura 322: <i>Upload</i> de Documentos 1, Fonte Própria	228
Figura 323: <i>Upload</i> de Documentos 2, Fonte Própria	228
Figura 324: <i>Upload</i> de Documentos 3, Fonte Própria	229
Figura 325: <i>Upload</i> de Documentos 4, Fonte Própria	229
Figura 326: <i>Upload</i> de Documentos 5, Fonte Própria	230
Figura 327: <i>Upload</i> de Documentos 6, Fonte Própria	230
Figura 328: <i>Upload</i> de Documentos 7, Fonte Própria	231
Figura 329: <i>Upload</i> de Documentos 8, Fonte Própria	231
Figura 330: <i>Upload</i> de Documentos 9, Fonte Própria	232
Figura 331: <i>Dwonload</i> de Documentos 1, Fonte Própria	232
Figura 332: <i>Dwonload</i> de Documentos 2, Fonte Própria	233
Figura 333: <i>Dwonload</i> de Documentos 3, Fonte Própria	233
Figura 334: <i>Dwonload</i> de Documentos 4, Fonte Própria	234
Figura 335: <i>Dwonload</i> de Documentos 5, Fonte Própria	234
Figura 336: <i>Dwonload</i> de Documentos 6, Fonte Própria	234
Figura 337: <i>Dwonload</i> de Documentos 7, Fonte Própria	235
Figura 338: <i>Dwonload</i> de Documentos 8, Fonte Própria	235
Figura 339: <i>Dwonload</i> de Documentos 9, Fonte Própria	236
Figura 340: Validação de Campos Obrigatórios 1, Fonte Própria.....	236
Figura 341: Validação de Campos Obrigatórios 2, Fonte Própria.....	237
Figura 342: Validação de Campos Obrigatórios 3, Fonte Própria.....	237
Figura 343: Validação de Campos Obrigatórios 4, Fonte Própria.....	238

Figura 344: Validação de E-mail 1, Fonte Própria	238
Figura 345: Validação de E-mail 2, Fonte Própria	239
Figura 346: Encriptação, Fonte Própria	240
Figura 347: ToolTip Editora/Autor na Interface Principal 1, Fonte Própria	241
Figura 348: ToolTip Editora/Autor na Interface Principal 2, Fonte Própria	241
Figura 349: Ampliação da Capa na Interface Gráfica Principal 1, Fonte Própria	242
Figura 350: Ampliação da Capa na Interface Gráfica Principal 2, Fonte Própria	242
Figura 351: Criar Ficheiro de Instalação da Aplicação 1, Fonte Própria.....	243
Figura 352: Criar Ficheiro de Instalação da Aplicação 2, Fonte Própria.....	244
Figura 353: Criar Ficheiro de Instalação da Aplicação 3, Fonte Própria.....	244
Figura 354: Criar Ficheiro de Instalação da Aplicação 4, Fonte Própria.....	245

(Página intencionalmente deixada em branco)

Índice de Tabelas

Tabela 1: Classes ADO.NET de uma Data Provider, Fonte (Corporation, n.d.)	56
Tabela 2: Máquinas Virtuais, Fonte própria	64

(Página intencionalmente deixada em branco)

1 Introdução

Um projeto, por definição, é uma atividade temporária, com data de início e de fim, definidas claramente, com um conjunto de objetivos, condições e responsabilidades estabelecidas, com um determinado orçamento e um planeamento definido, envolvendo várias partes interessadas no resultado final do projeto.

O presente relatório descreve o projeto desenvolvido pelo aluno Gonçalo Manuel Cota Charrua Pimenta de Abreu, no âmbito da Unidade Curricular Projeto Global, no Instituto Superior de Tecnologias Avançadas de Lisboa.

Este primeiro capítulo expõe o enquadramento, as motivações, os objetivos e a organização do projeto de dissertação.

1.1 Enquadramento

Embora o conceito de virtualização, não seja um conceito novo, o uso desta tecnologia vem crescendo exponencialmente nas infraestruturas de Tecnologias da Informação. Este crescimento deve-se sobretudo aos benefícios que esta tecnologia nos trás, tais como a rentabilização das capacidades dos atuais processadores, economia na gestão de equipamentos, energia e espaço. Com a virtualização é possível armazenar vários sistemas operativos e vários ambientes num mesmo *hardware*, podendo os seus recursos físicos ser partilhados pelas máquinas virtuais instaladas, sobre esse mesmo *hardware*.

A complexidade crescente, quer dos SGBDs quer das aplicações, conduziu a que os SGBDs relacionais se impusessem no mercado pela sua elegância, simplicidade e facilidade de utilização. A clareza e a sólida fundamentação dos seus conceitos essenciais permitiram que todos os intervenientes, fabricantes, analistas e utilizadores, se entendessem em torno de um modelo comum. A divulgação dos sistemas relacionais foi acompanhada de uma progressiva adesão das organizações à construção de sistemas de informação assentes em bases de dados.

1.2 Motivações

A principal motivação que levou ao desenvolvimento deste projeto, foi a oportunidade de aprendizagem de competências profissionais numa tecnologia muito utilizada em todas as grandes empresas, não só pela sua importância mas também devido a todos os custos que implica ter diversas máquinas a funcionar separadamente.

Desta forma, foi possível explorar este tipo de tecnologia, abordada ao longo da licenciatura de Informática e da Pós-Graduação em Virtualização e Cloud Computing, tendo uma visão mais profunda sobre ela, não só o seu conceito base mas toda a sua arquitetura e tecnologias existentes para assim se poder adotar a melhor solução para cada caso. Outra das principais razões pelo qual optei por este tema, foi o facto de cada vez mais as empresas estarem a recrutar engenheiros para administração de sistemas sendo esta uma das áreas de requisito para esse tipo de serviço e que me desperta bastante interesse.

1.3 Objetivos

O projeto consistiu em estudar, desenhar e implementar uma solução de virtualização com System Center Virtual Machine Manager, que permite configurar e gerir o *host* de virtualização, as redes e os recursos de armazenamento, de modo a criar e implementar máquinas virtuais e serviços numa rede virtual, com a finalidade de facilitar todo o processo de gestão e manutenção de um Administrador de Sistemas.

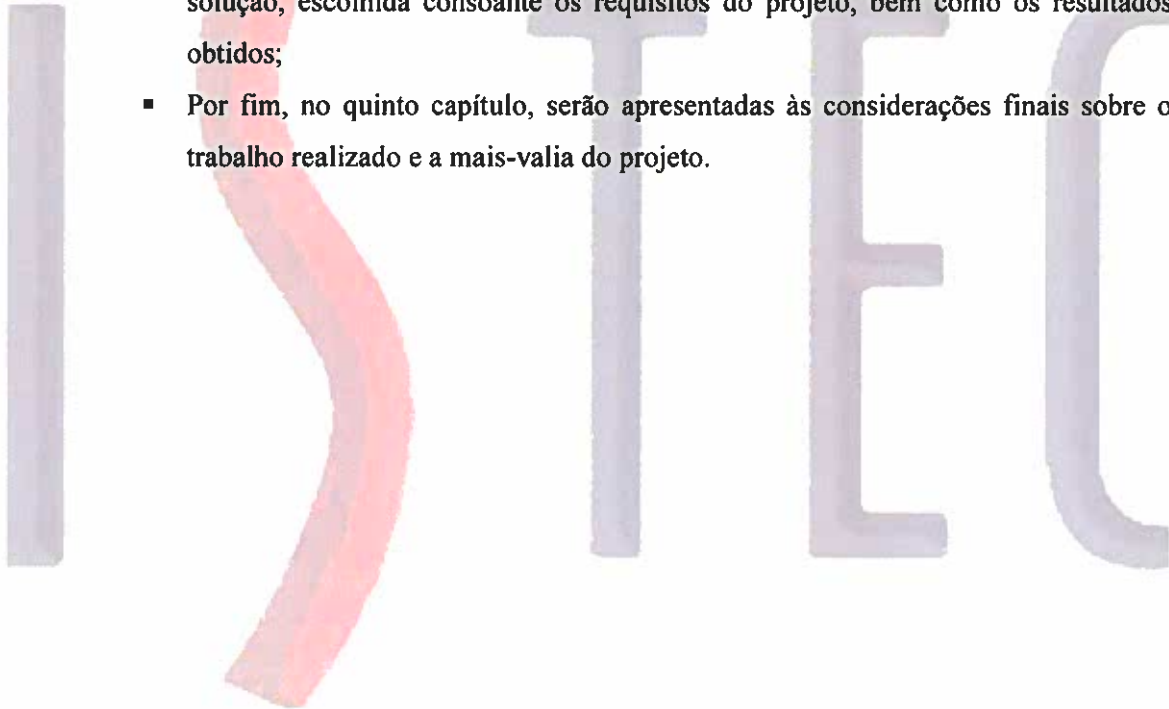
Neste projeto foi também estudada e desenvolvida uma aplicação em C#, com a interface para o utilizador em WPF, para uma Biblioteca Virtual, esta aplicação que será instalada automaticamente em cada máquina virtual cliente, criada a partir do *host* de virtualização.

Neste âmbito foi criado um sistema de base de dados relacional e alojado num servidor SQL, pelo que iremos demonstrar a forma de ligação da aplicação desenvolvida à base de dados através do ADO.NET.

1.4 Estrutura do Documento

Este documento compreende dois capítulos para além do presente capítulo, organizando-se da seguinte forma:

- No segundo capítulo é descrito o Estado da Arte: sendo apresentado o conceito de virtualização, a sua arquitetura, tecnologias e ainda alguns dos *softwares* que permitem a sua implementação e gestão;
- No terceiro capítulo é descrita a Contextualização, onde são apresentadas as metodologias utilizadas no projeto;
- No quarto capítulo é descrita toda a Implementação da Solução, pela apresentação detalhada de todos os passos e requisitos necessários para a implementação da solução, escolhida consoante os requisitos do projeto, bem como os resultados obtidos;
- Por fim, no quinto capítulo, serão apresentadas às considerações finais sobre o trabalho realizado e a mais-valia do projeto.



ISTEC

(Página intencionalmente deixada em branco)

2 Estado de Arte

O Estado da Arte é uma das partes mais importantes de todo trabalho científico, pois é o momento no qual se faz referência a produtos ou serviços que utilizam ferramentas e/ou tecnologias próximas, similares ou dentro da mesma área daquela que pretendemos desenvolver. Deste modo consegue-se fazer uma gestão daquilo que se pretende para o nosso projeto, bem como agilizar o processo desconstrução e inovação do mesmo.

No Estado da Arte do presente trabalho, é feita a contextualização desta dissertação, através da apresentação de conceitos fundamentais. Assim, na secção 1.1 apresentamos o estado de arte da virtualização com enfoque nos conceitos fundamentais; na secção 1.1.1 será realizada uma classificação a nível das técnicas de virtualização; na secção 1.2 abordaremos o tema sobre as plataformas de VMM, na secção 1.2.1 fala sobre a plataforma de VMM Hyper-V da Microsoft; na secção 1.3 descreve os modelos de implantação de Sistemas de Bases de Dados; na secção 1.3.1 abordamos o SQL; na secção 1.4 retrata a *framework .NET* e por fim na secção 1.4.1 fala de um dos componentes da *framework .NET* da Microsoft o *ADO.NET*.

2.1 Virtualização

O conceito de virtualização não é novo; se recuarmos aos anos 60 do século passado, verifica-se que pela primeira vez foi realizada uma implementação de um sistema de virtualização. Atualmente, existe um acréscimo no interesse sobre o estudo e o uso de técnicas de virtualização em infraestruturas de TI devido às suas inúmeras vantagens, tais como (Ruest & Ruest, 2009):

- Necessidade de espaço, desta forma diminui-se os equipamentos físicos, permitindo uma melhor gestão, bem como ter uma maior eficiência energética (Ruest & Ruest, 2009);
- Adaptação a diferentes ambientes – possibilidade de alterar os recursos físicos atribuídos inicialmente a uma máquina virtual, consoante as necessidades da mesma (Ruest & Ruest, 2009);
- Balanceamento de carga e maior facilidade para garantir redundância nos serviços (Ruest & Ruest, 2009).

O conceito de virtualização consiste na técnica de “montagem” de ambientes isolados, capazes de executarem sistemas operativos e aplicações separadamente, tornando a sua infraestruturas mais simples e eficiente e aproveitando ao máximo a capacidade do *hardware*.

Deste modo, os sistemas operativos e aplicações tiram melhor proveito da máquina física uma vez que são executados de forma independente, fazendo com que o desempenho e a disponibilidade sejam maiores e as operações cada vez mais mecanizadas (Steil, 2006).

Teoricamente não é possível afirmar qual a definição correta para o termo virtualização.

Esta tecnologia tem diversas soluções possíveis onde se pode implementar, dependendo apenas daquilo que se pretende, são elas (VMware, n.d.):

- **Virtualização de Servidores** – a arquitetura dos servidores x86¹ apenas permite executar um sistema operativo de cada vez; contudo, esta técnica possibilita que esse mesmo servidor, possa executar várias máquinas virtuais usando o *hardware* do servidor físico, subdividindo-o pelas diversas máquinas. Normalmente, não é aproveitado todo o potencial do servidor físico isolado, desta forma não só é utilizado na maioria dos recursos (CPU, RAM, discos rígidos, etc.), mas também é possível ter uma segurança isolada e poupança de energia, evitando custos desnecessários (VMware, n.d.);

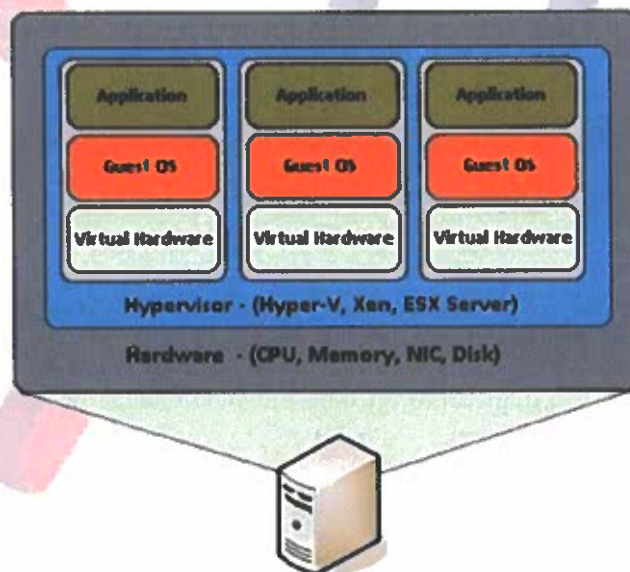


Figura 1: Virtualização de Servidores, Fonte (Mombrea, 2013)

- **Virtualização de Desktop** – a técnica usada nesta solução é idêntica à de virtualização em servidores, ou seja, fornecer várias máquinas virtuais usando o *hardware* do

¹ Nome genérico dada à arquitetura de processadores baseados no Intel 8086

desktop, onde cada utilizador pode aceder à sua área de trabalho virtual, a partir de qualquer local ou qualquer computador, tendo sempre as suas aplicações disponíveis sem perda de desempenho. A utilização desta técnica simplifica a gestão aos administradores de sistemas, uma vez que é centralizada, reduzindo o tempo de manutenção das máquinas e os custos para as instituições (Paper & Analyst, 2006; Virtue IT, 2015b; VMware, n.d.);

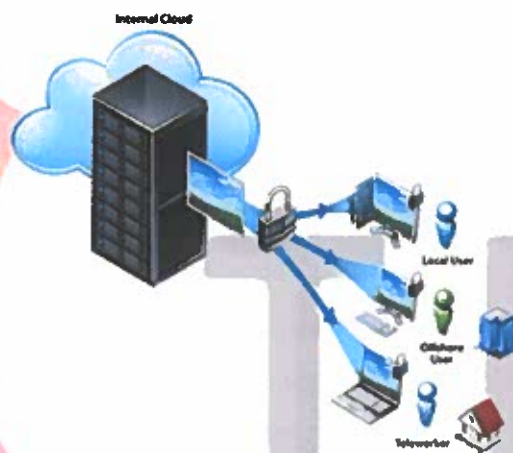


Figura 2: Virtualização de Desktop, Fonte (“Virtualization,” 2012)

- **Virtualização de Aplicações** – esta solução de virtualização consiste em isolar a aplicação, possibilitando assim executar qualquer *software* sem que este tenha sido instalado previamente. A não instalação do *software* evita a criação de ficheiros, DLLs e chaves de registo, o que muitas das vezes é prejudicial para o computador, uma vez que, por vezes, mesmo depois da sua desinstalação prevalecem, deixando os sistemas operativos mais lentos (Paper & Analyst, 2006; Virtue IT, 2015a, 2015b);

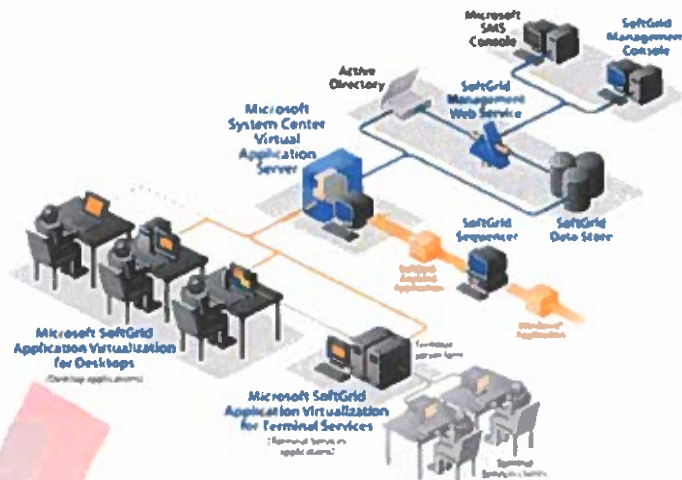


Figura 3: Virtualização de Aplicações, Fonte (Qureshi, 2007)

- **Virtualização de Redes** – a virtualização de rede consiste num ambiente de rede individual para cada grupo ou organização. Os ambientes são criados apenas numa infraestrutura, que é partilhada pela rede, em que cada grupo de utilizadores pertence a uma rede virtualizada, com diversos serviços, sendo semelhante a uma rede não virtualizada. Na perspetiva do utilizador final, está a aceder aos serviços de uma forma normal, com políticas de segurança independentes e recursos dedicados. Desta forma, a virtualização de rede engloba todos os dispositivos e serviços de rede numa estrutura de rede comum, centralizando diversos equipamentos e serviços, sendo possível aos grupos de utilizadores usufruírem com uma maior flexibilidade de outros serviços partilhados na rede. As redes virtualizadas são capazes de fornecer os mesmos recursos e garantias de uma rede física, bem como independência de *hardware* de virtualização: visão total da rede, manutenção mecanizada, balanceamento de carga, implementação contínua e suporte para novas aplicações que são executadas exatamente da mesma forma que numa rede física (Natário, 2011).

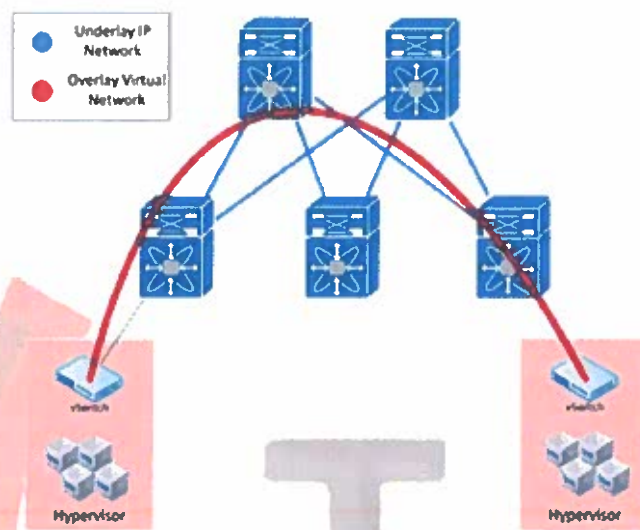


Figura 4: Virtualização de Redes, Fonte (Oswalt, 2013)

Em 1972, Robert P. Goldberg cientista americano de computação, iniciou a base teórica da arquitetura para sistemas computacionais de virtualização na sua dissertação na Universidade de Harvard. No mesmo ano, a empresa americana IBM lançou um *mainframe*² capaz de executar simultaneamente diferentes sistemas operativos sob a supervisão de um monitor de máquina virtual – *hypervisor*³ (Goldberg, 1974).

2.1.1 Classificação das Técnicas de Virtualização

Existem níveis de privilégios para que o sistema operativo controle as interações entre as aplicações e o *hardware*. Esses níveis de privilégio, chamam-se *rings* ou anéis, e estão organizados numa forma hierárquica que começa no anel 0, e que corresponde ao *kernel*⁴ do sistema operativo e como tal corresponde ao nível mais privilegiado. Os dois níveis seguintes correspondem aos componentes mais críticos do sistema operativo, como o caso de drivers. Sendo o último nível, aquele que engloba as aplicações que o utilizador interage, como o caso de navegadores de internet e processadores de texto.

² Computador de grande porte, dedicado normalmente ao processamento de um grande volume de informação.

³ Plataforma que permite aplicar diversas técnicas de controle de virtualização para utilizar, ao mesmo tempo, diferentes sistemas operativos no mesmo computador.

⁴ Componente central do sistema operativo dos computadores, serve de ponte entre os aplicativos e o processamento real de dados feito a nível de hardware.

Dentro das técnicas de virtualização existentes, os três tipos mais utilizados são a Virtualização Total (*Full Virtualization*), a Paravirtualização (*OS-Layer* ou *Paravirtualization*) e a Virtualização Assistida por *Hardware* (*Hardware-Layer Virtualization*). Cada um destes tipos de Virtualização funciona num conjunto de anéis diferentes.

Estes três tipos de virtualização são explicados mais detalhadamente (VMware, 2007):

- **Virtualização Total:** Permite ao sistema operativo convidado aceder diretamente ao processador e disco. Cada convidado é independente entre si, não tendo conhecimento da existência de outros convidados. O anfitrião é responsável por gerir todo o sistema, incluindo alocar recursos para cada um dos sistemas operativos convidados. Sistema de virtualização opera no anel 0, já o sistema operativo convidado opera no anel 1. Este sistema é conhecido como *Hypervisor*. Existem várias plataformas que aplicam este tipo de virtualização, como é o caso do *VirtualBox*⁵ e do *Microsoft Server 2012*⁶ (VMware, 2007). Demonstrado na Figura 5.

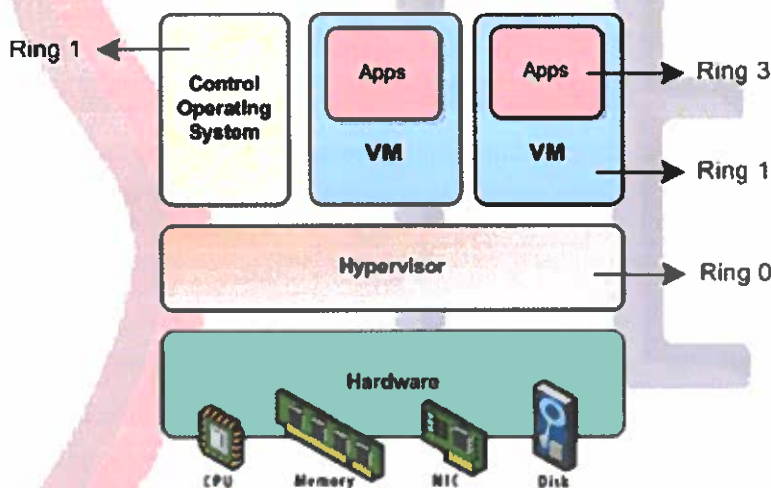


Figura 5: Virtualização Total, Fonte (Ribeiro, 2009)

- **Paravirtualização:** Relativamente semelhante com a virtualização completa, mas neste caso os sistemas operativos convidados necessitam de modificações ao nível do sistema operativo, para que as operações não virtualizáveis possam ser chamadas a partir da interface do *hypervisor*. O anfitrião tem uma menor carga computacional, uma vez que cada convidado tem acesso total ao processador. Como são necessárias

⁵ <https://www.virtualbox.org/manual/ch10.html>

⁶ <https://www.microsoft.com/en-us/server-cloud/products/windows-server-2012-r2/>

modificações ao núcleo do sistema operativo, esta técnica sofre de portabilidade e impossibilita que seja o utilizador a escolher livremente qual o desktop a usar. O sistema operativo convidado opera no anel 0. O *Xen Project*⁷ é um exemplo de Paravirtualização (VMware, 2007). Podemos ver esta técnica na Figura 6.

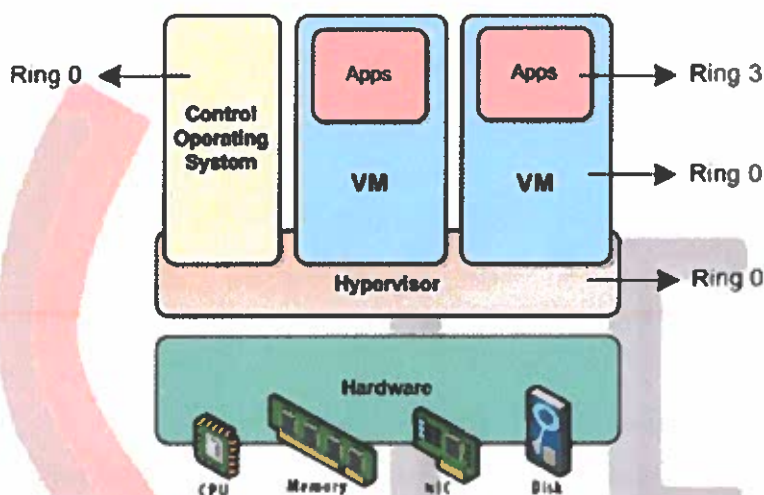


Figura 6: Paravirtualização, Fonte (Ribeiro, 2009)

- **Virtualização Assistida por Hardware:** Neste contexto não existe conceito de *Hypervisor*. Existe um *kernel* comum e partilhado por todas as instâncias em vez de cada instância ser independente. Este método também é conhecido por *containers virtualization*. Cada *container*⁸ equivale a uma instância do sistema operativo base, não sendo assim possível ter um sistema operativa da família *Windows*⁹ tendo como base um *Linux kernel*¹⁰. Esta limitação proíbe o uso deste modelo para o objetivo proposto, uma vez que é pretendido usar desktops de famílias distintas, e independentes do sistema operativo base. Sistema Operativo convidado situa-se no anel 0. Este tipo de virtualização é implementado por vários produtos no mercado, como por exemplo o *VMware Workstation*¹¹ (VMware, 2007). Ver a Figura 7.

⁷ <http://www.xenproject.org/users/why-the-xen-project.html>

⁸ Delimitador abstrato, um objeto que contém outros objetos que podem ser incluídos ou removidos dinamicamente

⁹ <https://www.microsoft.com/pt-pt/windows/>

¹⁰ <http://www.linux.com/news/software/linux-kernel>

¹¹ <https://www.vmware.com/products/workstation>

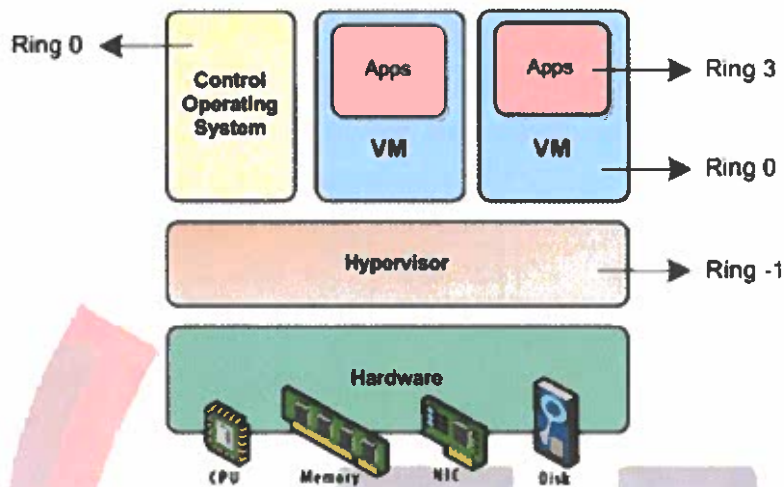


Figura 7: Virtualização Assistida por Hardware, Fonte (Ribeiro, 2009)

2.2 Plataforma de Monitor de Máquinas Virtuais - Hypervisor

Os sistemas de máquinas virtuais são capazes de virtualizar todo um conjunto de *hardware*, incluindo memórias, processador/es, discos e dispositivos periféricos. Um Monitor de Máquina Virtual, em inglês *Virtual Machine Monitor (VMM)* ou *Hypervisor*, é um modo ou camada de *software* que permite ter controlo total sobre o CPU e ainda sobre a virtualização utilizada nos diferentes sistemas operativos, ou seja, substitui o *kernel mode*¹² (modo de supervisão). Assim, é feita a ativação desta camada, o *kernel mode* deixa de ter controlo sobre o CPU e embora a interface entre o *kernel mode* e o modo de utilizador seja a mesma, passa a existir uma nova interface entre ambos: todas as instruções privilegiadas executadas pelo *kernel*, em *kernel mode*, são passadas através do modo *hypervisor*, sendo que o VMM pode desta forma emular o comportamento desejado. Este método é também conhecido por “imitar e emular” (“trap and emulate”). Através deste método, todos os erros de sistema (crashes das máquinas virtuais), acessos dos periféricos I/O, tarefas, etc, são sempre tratadas desta forma.

A empresa americana IBM foi a primeira a implementar o modo *hypervisor* num processador, PowerPC 970 ou G5¹³. Nos dias de hoje, a empresa *Intel*¹⁴ e *AMD*¹⁵ já têm

¹² Código de execução que tem acesso completo e irrestrito ao hardware subjacente.

¹³ <http://www.ibm.com/developerworks/systems/library/es-apple.html>

¹⁴ <http://www.intel.com/content/www/us/en/homepage.html>

¹⁵ <http://www.amd.com/en-gb>

processadores com arquitetura semelhantes e destinados a virtualização, *Intel VT-X*¹⁶ e *AMD-V*¹⁷ respectivamente (Goldberg, 1974; Steil, 2006).

Segundo Goldberg e Popek, um Monitor de Máquina Virtual deve apresentar três propriedades fundamentais, que permitem determinar se a arquitetura do computador suporta uma virtualização concisa / eficiente e capaz de fornecer todos os requisitos aos sistemas virtualizados (Popek & Goldberg, 1973):

- **Fidelidade / Equivalência:** o sistema operativo implementado numa VMM deve ter um comportamento idêntico ao quando executado diretamente no *hardware* equivalente;
- **Segurança e Controlo de Recursos:** O VMM deve conter os mesmos recursos de segurança e controlo total sobre os sistemas;
- **Desempenho e Eficiência:** A maioria das instruções dos sistemas devem ser executados pelo *hardware* sem necessitarem de intervenção do VMM.

A camada *Hypervisor* pode ser classificado em dois tipos (Jason & Basrai, 2015):

- **Tipo 1:** Virtualização nativa ou também chamada *bare-metal*¹⁸:

Este tipo de virtualização é o mais usado em *datacenters*¹⁹, porque é uma técnica onde a camada do *hypervisor* assenta diretamente sobre o *hardware* e todos os sistemas operativos virtualizados são executados sobre ela. A tarefa desta camada é gerir a alocação da memória e dos recursos para as máquinas virtuais, fornecendo muitas das vezes interfaces de administração e monitorização. (Jason & Basrai, 2015)

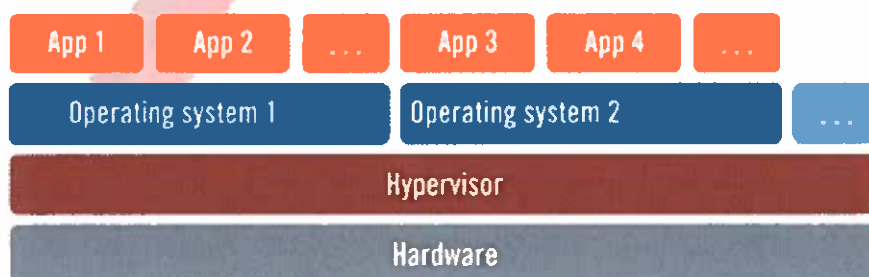


Figura 8: Hypervisor Tipo 1, Fonte (Jason & Basrai, 2015)

¹⁶ <http://ark.intel.com/Products/VirtualizationTechnology>

¹⁷ <http://www.amd.com/en-us/solutions/servers/virtualization>

¹⁸ Computador sem Sistema Operativo

¹⁹ Local onde são concentrados os equipamentos de processamento e armazenamento de dados de uma empresa ou organização

Hypervisors deste tipo: *Microsoft Hyper-V*²⁰, *VMware ESXi*²¹, *Xen*²²

▪ **Tipo 2:** Virtualização com sistema operativo ou também chamado de *hosted*:

Este tipo, tal como o nome indica, é uma virtualização implementada através de um sistema operativo, ou seja, um sistema operativo corre sobre o *hardware* e por sua vez a camada *hypervisor* corre em cima deste SO, sendo esta um *software hosted*²³. Neste tipo, o VMM garante todos os serviços do sistema físico (memórias, dispositivos, BIOS, etc) a cada máquina virtual, fazendo com que cada uma “pense” que está a ser executada num *hardware* físico sem qualquer camada. Desta maneira, o próprio SO fornece uma “camada” parecida ao *hypervisor* que permite virtualizar outros sistemas operativos no seu interior, criando assim máquinas virtuais. (Jason & Basrai, 2015)

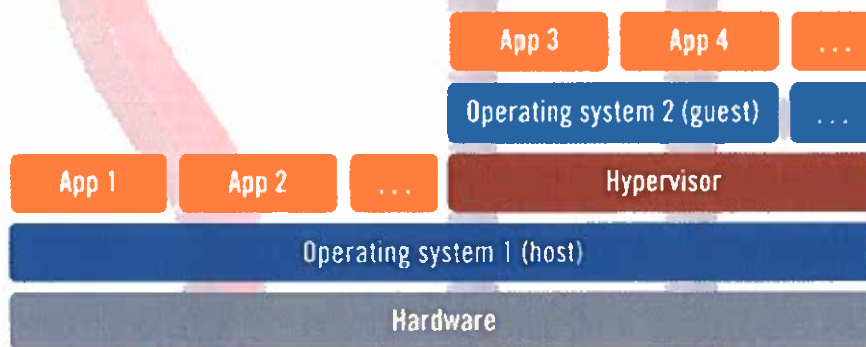


Figura 9: Hypervisor Tipo 2, Fonte (Jason & Basrai, 2015)

Hypervisors deste tipo: *VMWare Player*²⁴, *RedHat KVM*²⁵, *Oracle VirtualBox*²⁶.

2.2.1 Hyper-V

A tecnologia *Hyper-V* virtualiza o *hardware* de modo a disponibilizar um ambiente no qual é possível executar vários sistemas operativos em simultâneo num único computador físico.

²⁰ <https://technet.microsoft.com/en-us/library/mt169373.aspx>

²¹ <https://www.vmware.com/br/products/esxi-and-esx/overview>

²² <http://xenserver.org/about-xenserver-open-source.html>

²³ Software que é instalado, hospedado e acedido inteiramente a partir de um servidor remoto ou local.

²⁴ <https://www.vmware.com/products/player>

²⁵ <https://www.redhat.com/en/resources/kvm-%E2%80%93-kernel-based-virtual-machine>

²⁶ <https://www.virtualbox.org/>

O *Hyper-V* permite criar e gerir as máquinas virtuais e os recursos correspondentes. Cada máquina virtual representa um sistema informático virtualizado isolado apto a executar o seu próprio sistema operativo. O sistema operativo executado numa máquina virtual é designado por sistema operativo convidado.

Tem como principal função criar e gerir um ambiente de virtualização usando a tecnologia *Hyper-V* presente no *Windows Server 2012*, ou seja, requer a instalação deste sistema operativo e só depois é possível ativar esta ferramenta (Figura 11) (Microsoft Corporation, n.d.-c).

Com a ativação do *Hyper-V* são instalados alguns componentes necessários para o uso da ferramenta, são eles (Microsoft Corporation, n.d.-c):

- Ferramentas de gestão de máquinas virtuais;
- Componente de barramento *VMbus*²⁷ (*Virtual Machine Bus*);
- Provedor WMI²⁸ (*Windows Management Instrumentation*) de virtualização;
- VID²⁹ (*Virtual Infrastructure Driver*);
- VSP³⁰ (*Virtualization Service Provider*).

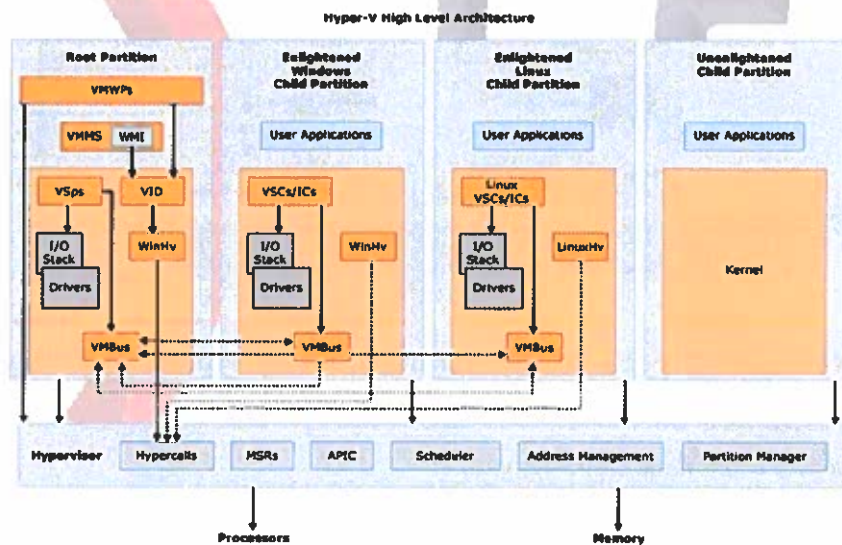


Figura 10: Arquitetura Hyper-V, Fonte (Microsoft Corporation, n.d.-d)

²⁷ Canal de comunicação lógica inter-partição (Partição anfitrião e convidado).

²⁸ Conjunto de extensões para o Windows Driver Model que fornece uma interface de sistema operacional através da qual componentes instrumentados fornecem informação e notificação.

²⁹ Fornece serviços de gestão de partições, de processadores virtuais e de memória para partições.

³⁰ Encontra-se na partição de raiz e fornece suporte sintético para as partições filho sobre o VMbus.

As ferramentas de gestão são baseadas em interface gráfica para facilitar a interação por parte do utilizador e permitem (Microsoft Corporation, n.d.-c):

- Gerir o *Hyper-V*;
- Possui uma consola de gestão da *Microsoft*³¹ (*Microsoft Management Console*);
- Permitem uma ligação “gráfica” às máquinas virtuais (*Virtual Machine Connection*) para desta forma ser possível controlá-las e interagir;

*Cmdlets*³² específicos do *Hyper-V* para a *PowerShell*³³ do *Windows*.

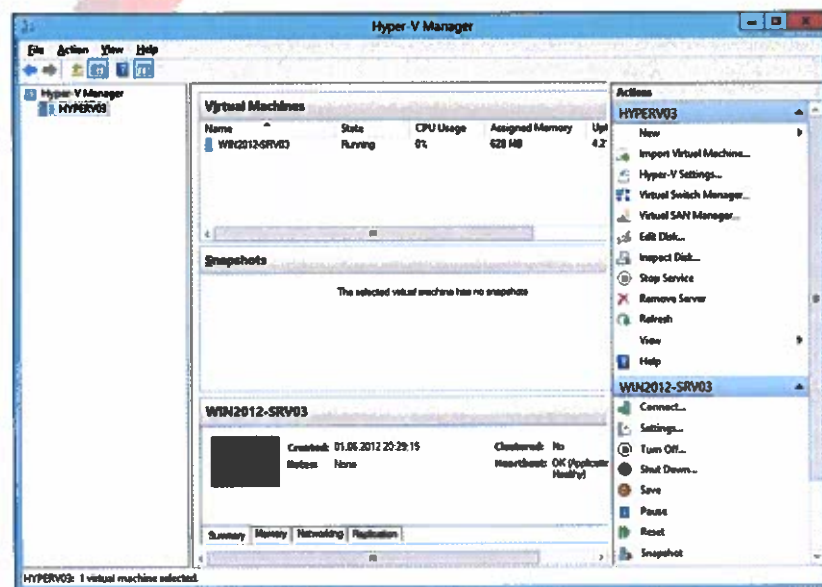


Figura 11: Plataforma de Gestão do Hyper-V, Fonte (Davis, 2013)

No que concerne aos benefícios desta tecnologia, estes passam por (Advantages & Server, 2013; Microsoft Corporation, n.d.-i):

- Eficiência económica e operacional;
- Conjunto de ferramentas unificado: seja para ambientes físicos / servidores, *cloud*, ou virtuais, tudo numa única plataforma;
- Mais económica que as soluções *VMware*: a gestão de todo o *datacenter* é ilimitado em termos de licenças, com base em processadores.

³¹ <https://www.microsoft.com/en-us/>

³² Comandos leves usados no ambiente Windows PowerShell, são invocados por este em contexto de scripts automáticos através da linha de comandos

³³ <https://msdn.microsoft.com/en-us/powershell/mt173057.aspx>

Algumas das características que o *Microsoft Hyper-V* apresenta são (Advantages & Server, 2013; Microsoft Corporation, 2015):

- Disco rígido virtual partilhado;
- Gestão simplificada;
- Rede *Hyper-V*;
- *Clustering*³⁴ de *Failover* e *Hyper-V*, fazendo proteção da placa de rede virtual e do armazenamento da máquina virtual;
- Migrações dinâmicas;
- Ativação automático de uma máquina virtual;
- Duplicar o *Hyper-V*;
- Serviços de integração;
- *Virtual Machine Generation*, é determinando o *hardware* virtual e a funcionalidade que está atribuída à máquina virtual, suporte para UEFI, inicialização segura, etc.

2.3 Sistemas de Base de Dados

Um SGBD tem como principal função gerir todo o processo de manutenção de informação dos sistemas existentes. No entanto, as funcionalidades e o tipo de tratamento da informação dependem do tipo de SGBD utilizado, como veremos de seguida (Date, 2004).

Sistemas de Base de Dados Relacional

O modelo relacional para bases de dados é dos mais utilizados e amplamente popular a nível global. Edgar Frank Codd foi o matemático que teorizou este modelo aquando do seu trabalho de pesquisa na IBM. Tendo por base a matemática o modelo assenta as suas propriedades e características na teoria dos conjuntos (Codd, 1970).

Um SGDB relacional como o *Microsoft SQL Server*³⁵ ou o *MySQL*³⁶, atualmente utilizados na Janela Digital, têm como principal diferença relativamente aos restantes tipos de SGBD o facto de estruturarem os seus dados em tabelas e de se basearem em álgebra relacional, o que facilita a sua compreensão. Além disso, um SGDB relacional tem as seguintes funções (Date, 2004):

³⁴ Tarefa de agrupar um conjunto de dados ou objetos segundo seu grau de semelhança.

³⁵ <https://www.microsoft.com/pt-pt/server-cloud/products/sql-server/>

³⁶ <https://www.mysql.com/>

- Manipulação de dados;
- Validação e otimização;
- Integridade e segurança dos dados;
- Concorrência e recuperação de dados; • Criação e manutenção dos metadados³⁷;
- Desempenho.

Como podemos verificar, estes efetuam múltiplas e variadas funções, facilitando a sua utilização e manutenção por parte do utilizador, uma vez que é tudo tratado pelo SGDB (Date, 2004).

Muitas destas funções são mantidas pelo conjunto de propriedades destinadas a assegurar a confiança nas transações, conhecidas como ACID (Atomicidade, Consistência, Isolamento, Durabilidade).

- **Atomicidade** - cada transação é atômica, se uma parte da transação falha toda a transação falha;
- **Consistência** - as transações respeitam as regras impostas pelos metadados;
- **Isolamento** - uma informação só pode ser alterada ou acedida se não houver mais nenhuma transação a utilizar a mesma informação;
- **Durabilidade** - o efeito de uma transação com sucesso é permanente;

As propriedades enunciadas permitem garantir a fiabilidade e consistência de uma BD. Nestes SGBD o isolamento funciona através do bloqueio da informação a ser processada. A transação marca a informação a ser acedida e o SGDB não permite que outras transações acedam à mesma informação até que esta termine. Este funcionamento pode levar à sobrecarga do sistema e ao adiamento de outras transações, (se uma transação for demasiado complexa) levando a uma diminuição do desempenho do sistema.

Uma das transações mais utilizadas mas, ao mesmo tempo, dispendiosa em termos de processamento, é a operação *JOIN*³⁸. Devido à sua função de junção de tabelas, é utilizada para relacionar a informação contida em várias tabelas. A operação *JOIN* é caracterizada por ser um cruzamento de todos os registos das tabelas seleccionadas, seguido de seleções e projeções. O resultado deste cruzamento de informação é normalmente muito maior do que o resultado final do *JOIN* e por essa razão não se pode desprezar a carga envolvida com a utilização desta operação.

³⁷ Dados sobre outros dados

³⁸ Clausula SQL que combina registos de duas ou mais tabelas de uma SGBDR

Com esta estrutura compreendem-se as dificuldades dos SGBDR em manipular com elevada eficiência enormes quantidades de dados. O facto de a operação efetuar o cruzamento de todos os dados das tabelas seleccionadas, juntando ao facto de estas tabelas conterem um elevado número de registos é uma preocupação, pois aumenta exponencialmente o número de informação cruzada e trabalhada (Date, 2004; Gray, 1981; Ramakrishnan & Gehrke, 2003).

A sua estrutura faz com seja necessário normalizar os dados de modo a evitar redundâncias e incongruências. Assim sendo, é necessário efetuar um estudo e uma preparação prévia para otimizar a BD e obter o melhor desempenho possível. Por outro lado, esta estrutura rígida faz com qualquer alteração seja de elevada dificuldade para a manutenção do mesmo desempenho. Muitas vezes será necessário reestruturar a BD novamente com os custos e tempo inerentes à mesma.

O facto de as BD relacionais serem a referência na área e utilizadas nos mais diversos ambientes faz também com que sejam muito testadas e com que tenham um maior suporte através da comunidade existente, bem como da empresa responsável.

As BD relacionais não foram desenhadas para serem distribuídas, sendo esta a chave para a escalabilidade³⁹, uma vez que os dispositivos de armazenamento se encontram incorporados num CPU comum e não distribuídos pelos nós do sistema (Özsu & Valduriez, 2011). Devido a esta arquitetura, a escalabilidade é um problema para sistemas em constante crescimento, pois a alteração centra-se na adição de recursos (CPU ou memória) a um nó do sistema. Isto advém do facto de haver vários nós com funções completamente distintas. Separadamente, existem os nós de escrita - reconhecidos por serem os pontos de rutura do sistema - e os nós de leitura - associados ao espaço físico utilizado (Burlison, n.d.; Rouse, 2013).

Sistemas de Base de Dados Não Relacional

Estes SGBD são conhecidos por *NoSQL*, que significa “Not Only SQL”, salientando o facto de não conterem um esquema fixo, principal característica das BD relacionais. Estas não se relacionam por ligações nas tabelas, (porque o esquema deixa de ser em tabelas) mas continuam a existir relações entre as entidades⁴⁰, embora não explícitas e truncadas como nos SGBDR. Estas

³⁹ Característica desejável em todo o sistema, em uma rede ou em um processo. Indica sua capacidade de manipular uma porção crescente de trabalho de forma uniforme, ou estar preparado para crescer.

⁴⁰ Qualquer coisa concreta ou abstrata, abstraída do mundo real e modelada em forma de tabela que guardarão informações na base de dados.

relações são constituídas de outra forma e não explicitamente na base de dados como nas relacionais.

Nos anos de 2009 e 2010, com o ressurgimento das BD não-relacionais em grande escala, aumentou a discussão sobre as vantagens e desvantagens destas, em contraposição com as BD relacionais.

Antes de qualquer tipo de comparação, convém referir que os SGBD não-relacionais contemplam, independentemente da plataforma utilizada, as seguintes vantagens:

- Alta escalabilidade;
- Alta disponibilidade;
- Alto desempenho (mesmo com enormes quantidades de dados);
- Flexibilidade;
- *Software* livre (sem custos).

Convém salientar o facto de se indicar o *software* livre e sem custos como uma vantagem dos SGDB não-relacionais, mas a característica não é exclusiva destas. Existem vários SGBDR que são reconhecidas pela sua qualidade e que também não contemplam custos para o utilizador, como o *PostgreSQL*⁴¹ e o *MySQL*. Por não se colocar essa questão, não haverá referências ao custo da sua utilização, contrariamente às SGBDR.

O principal fundamento para estas vantagens - independentemente do SGDB - é o facto da lógica de validação, do controlo de acessos, do mapeamento de dados indexados, do corelacionamento dos dados, da resolução de conflitos e dos procedimentos desencadeados por *triggers*⁴² passarem da camada de base de dados, para a camada da aplicação. Desta forma, a camada da BD é deixada unicamente com preocupações de desempenho e escalabilidade (Zyp, 2010).

Por outro lado, o facto de todos os controlos serem feitos na camada de aplicação e não pelo SGDB, coloca em necessidade o seguinte:

- Atenção especial para a inconsistência nos dados
- Gestão de conflitos

⁴¹ <http://www.postgresql.org/>

⁴² Tipo especial de procedimento armazenado, que é executado sempre que há uma tentativa de modificar os dados de uma tabela que é protegida por ele.

O facto da validação e resolução de conflitos passar para a camada da aplicação leva a uma preocupação extra essencialmente por parte de quem desenvolve a aplicação. Com a implementação de uma SGBD não-relacional é necessário garantir que estas características são asseguradas ou então existe o risco de a BD ser inconsistente e incongruente, não sendo a mesma fiável.

Existem vários tipos de modelos de armazenamento de dados utilizados pelas BD não-relacionais: o armazenamento por valores das chaves, por documentos e por colunas.

Enquanto que o armazenamento por valores das chaves surge como um modelo extremamente rápido e eficiente pela sua simplicidade, o armazenamento de colunas providencia um modelo flexível e extremamente estruturável. Por sua vez, o armazenamento por documentos combina a simplicidade das BD não-relacionais com uma maior estruturação (Leavitt, 2010).

2.3.1 SQL

A Structured Query Language (SQL) encontra-se normalizada e segue as normas ISO⁴³ e ANSI⁴⁴, contudo existem algumas extensões adicionadas individualmente por diferentes proprietários de SGBD. A primeira versão apareceu no início dos anos 70, pela mão de Donald D. Chamberlin e Raymond F. Boyce na IBM (Chamberlin & Boyce, 1974). Esta linguagem foi criada com o intuito de lidar com um sistema proprietário da IBM denominado de *System R*⁴⁵, baseado no modelo relacional (Chamberlin et al., 1981).

É uma linguagem utilizada na pesquisa e criação de base de dados. Em comparação com outros tipos de linguagem utilizada para os mesmos fins, podemos referenciar que o SQL é um aplicativo simples e relativamente fácil de utilizar o que atualmente o torna um dos mais usados no mundo. As suas principais características prendem-se com (Ramakrishnan & Gehrke, 2003):

- Integridade dos dados;
- Otimização das consultas;
- Dados independentes a nível físico e lógico;
- *Backup*⁴⁶ de dados;

⁴³ Lista de normas internacionais da Organização Internacional para Padronização (ISO).

⁴⁴ Organização particular dos Estados Unidos sem fins lucrativos que tem por objetivo facilitar a padronização dos trabalhos dos seus membros.

⁴⁵ Sistema de base de dados construído pela IBM no seu Centro de Pesquisas de Almaden

⁴⁶ Cópia de dados de um dispositivo de armazenamento a outro para que possam ser restaurados em caso da perda dos dados originais

- Segurança.

Existe um vasto número de palavras-chave utilizados pelo SQL que permitem a manipulação, controle e definição de dados (Petković, 2008).

2.4 Framework .NET

A *framework .NET*⁴⁷ é uma iniciativa da *Microsoft* que visa disponibilizar uma plataforma única para o desenvolvimento e execução de sistemas de aplicações. O código gerado para *framework .NET* pode ser executado em qualquer dispositivo que possua esta plataforma. Deste modo, o programador deixa de escrever código para um sistema ou dispositivo específico, e passa a escrever para a plataforma *.NET*. (Microsoft Corporation, n.d.-e) No capítulo 1.4.1 é apresentado um dos componentes desta plataforma, em analogia com a Figura 12, o *ADO.NET*. (Microsoft Corporation, 2005)



Figura 12: Arquitetura Framework .NET, Fonte (Microsoft Corporation, 2005)

⁴⁷ <https://www.microsoft.com/net>

2.4.1 ADO.NET

O *ADO.NET*, *ActiveX Data Objects*, é um dos componentes da *framework* .NET que disponibiliza um conjunto de classes que providenciam vários serviços para acesso a dados, sejam eles relacionais, XML⁴⁸, OLE⁴⁹ ou ODBC⁵⁰. Através do *ADO.NET* as aplicações podem facilmente estabelecer ligações a este tipo de fontes de dados de modo a recuperar, armazenar ou atualizar dados.

As principais características do *ADO.NET* são: (Microsoft Corporation, n.d.-a)

- Desenhado para ser utilizado em ambientes desconectados
- Modelo de programação com suporte avançado para XML
- Conjunto de classes, interfaces, estruturas e enumerações que permitem gerir o acesso a dados dentro das aplicações desenvolvidas em ambiente .NET

Como já foi referido, o *ADO.NET* suporta vários tipos de dados, entre eles:

- Dados não estruturados;
- Estruturados, não hierárquicos;
- Ficheiros CSV⁵¹ (*Comma Separated Value*);
- Documentos XML;
- Dados de Bases de Dados Relacionais.

As classes do *ADO.NET* estão assim organizadas por *data provider*⁵² (fornecedor de dados). O ponto-chave da arquitetura definida pela *Microsoft* para o *ADO.NET* é que cada classe é abstrata e, conseqüentemente, cada *data provider* implementa a sua própria versão das várias classes. Por exemplo, existe um *data provider* para as bases de dados *Microsoft SQL Server* e existe outro para base de dados *Oracle*; conseqüentemente, existe uma classe *SQLConnection* definida no *data provider SQL Server* e uma classe *OracleConnection* definida no *data provider Oracle*. (Mariani, Meier, Vasireddy, Babbar, & Mackman, 2004)

⁴⁸ Recomendação da W3C para gerar linguagens de marcação para necessidades especiais.

⁴⁹ Representa uma biblioteca de interface COM que permite o acesso universal a diversas fontes de dados.

⁵⁰ Padrão para acesso a sistemas gerenciadores de bases de dados (SGBD)

⁵¹ Arquivos de formato regulamentado pelo RFC 4180 que faz uma ordenação de bytes ou um formato de terminador de linha

⁵² Componente do ADO.NET que interage com uma fonte de dados.

Tabela 1: Classes ADO.NET de uma Data Provider, Fonte (Corporation, n.d.)

Object	Descrição
Connection	Estabelece uma conexão com uma fonte de dados específica.A classe base para todos os objetos de Connection é a classe DbConnection .
command	Executa um comando em uma fonte de dados.Expõe Parameters e pode ser executado no escopo de uma Transaction de Connection .A classe base para todos os objetos de Command é a classe DbCommand .
DataReader	Ler um fluxo de dados apenas de encaminhamento e somente leitura de uma fonte de dados.A classe base para todos os objetos de DataReader é a classe DbDataReader .
DataAdapter	Preenche um DataSet e resolve atualizações com a fonte de dados.A classe base para todos os objetos de DataAdapter é a classe DbDataAdapter .

Os dados retornados pelas execuções de comandos *ADO.NET* podem ser processados diretamente ou armazenados num objeto *ADO.NET DataSet*⁵³. Este objeto possibilita a realização de um vasto conjunto de operações como combinação de dados de diversas fontes, estabelecimento de relações entre tabelas, manipulação conjunta de informação, entre outras. As classes do *ADO.NET* estão localizadas na biblioteca *System.Data*⁵⁴.*[data provider]* onde *[data provider]* especifica o fornecedor de dados que se pretende utilizar, por exemplo, *SqlClient*.

Esta abordagem do *ADO.NET* proporciona uma vantagem clara: cada *data provider* pode otimizar internamente as suas classes, tendo em conta o seu modo de funcionamento interno (Mariani et al., 2004).

Na Figura 13 é apresentado um exemplo de uma execução utilizando as classes *ADO.NET* para uma base de dados *SQL Server*.

⁵³ Coleção de dados normalmente tabulados

⁵⁴ Coleção organizada de símbolos e processos que podem ser usados para funcionar em tais símbolos.

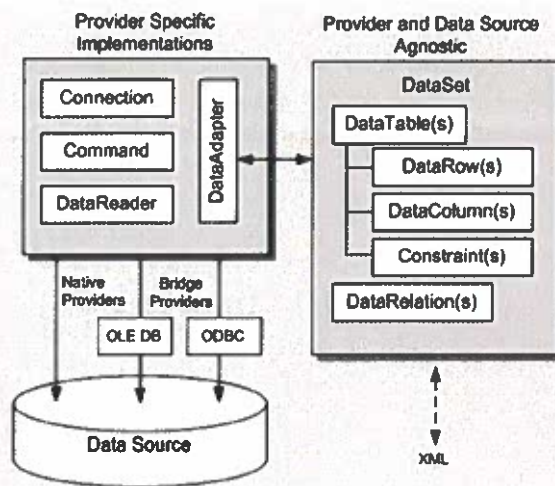


Figura 13: Exemplo de um acesso a base de dados com ADO.NET, Fonte (Meier, Vasireddy, Babbar, & Mackman, 2004)

A chamada à primeira classe *Connection* estabelece a ligação ao servidor SQL e à base de dados a utilizar. É depois executado o *Command* que contém uma instrução SQL que indica à base de dados o que fazer; o *DataReader* é um cursor de leitura rápido que vai ler os dados seleccionados pela instrução SQL. O *DataGrid* vai exibir os registos e despoleta eventos quando são adicionados, removidos ou editados registos. Estes eventos criam novas instruções SQL que alteram a base de dados e mostram os registos (Tracy, 2003).

ISTEC

(Página intencionalmente deixada em branco)

3 Contextualização

Para a realização deste projeto foi necessário efetuar um estudo e análise dos diversos sistemas de virtualização existentes no mercado. Numa segunda fase, foi necessário verificar o *hardware* disponível para os vários tipos de ferramentas, de modo a cumprir os requisitos de cada ferramenta. Por último, foi desenhada toda a estrutura da solução.

3.1 Tecnologia:

O cenário de laboratório teve como base uma biblioteca virtual. Este laboratório irá decorrer dentro de um domínio e será estruturado da seguinte forma:

- Um servidor físico com o Microsoft Windows 10 Profissional e com um *hypervisor*, neste caso o VMware Workstation 12 Profissional, que é simulado por um computador portátil,
- No VMware estarão disponíveis os seguintes servidores virtuais com as seguintes funções:
 - Servidor com o Microsoft Windows Server 2012 R2 Standard: irá desempenhar a função de Controlador de Domínio, onde serão adicionadas as seguintes roles:
 - Active Directory Domain Services,
 - DHCP Server
 - DNS Server
 - Servidor com o Microsoft Windows Server 2012 R2 Datacenter: irá ter a função de Servidor de SQL, no qual será instalado o Microsoft SQL Server 2012 Enterprise, onde ficarão as seguintes bases de dados:
 - Base de dados do SCVMM
 - Base de dados da Biblioteca Virtual
 - Servidor com o Microsoft Windows Server 2012 R2 Datacenter: irá desempenhar a função de Virtual Machine Manager, onde será ativada a role do Microsoft Hyper-V, será também instalado o Microsoft System Center Virtual Machine Manager 2012, e a partir deste serão disponibilizadas as seguintes máquinas virtuais clientes através de um *template*, pelo administrador do domínio:

- Máquina Virtual Cliente com o Microsoft Windows 8.1 Professional, para acesso por parte dos utilizadores à base de dados a fim de consulta,
- Máquina Virtual Cliente com o Microsoft Windows 8.1 Professional, para acesso por parte dos administradores da aplicação da Biblioteca Virtual, à base de dados a fim de edição da BD.

3.2 Base de Dados

- Construída uma base de dados para a Biblioteca Virtual no Microsoft SQL Server 2012 Enterprise,
- A base de dados criada as seguintes tabelas:
 - Documentos, com os seguintes conteúdos:
 - Capa, com as seguintes atributos:
 - Imagem da capa do documento, ou da 1ª página quando não for um livro,
 - Título,
 - Autor,
 - Editora,
 - Data, com os seguintes atributos:
 - Edição ou produção do documento,
 - Tipo, com os seguintes atributos:
 - Livro, Jornal, Manuscrito, Imagem, etc,
 - Resumo, com os seguintes atributos:
 - Resumo do documento em texto,
 - Documento, com os seguintes atributos:
 - Em versão PDF para Download
 - Editoras, com os seguintes conteúdos:
 - Nome,
 - Morada,
 - Email,
 - Telefone

- Autores, com os seguintes conteúdos:
 - Nome,
 - Nacionalidade,
 - Data de Nascimento
- Tipos de Documento, com os seguintes conteúdos:
 - Descrição

3.3 Interface de acesso à BD

- Produzir uma aplicação em C#, com interface para o utilizador em WPF e ligação à base de dados através do ADO.NET,
- Existirá um *icon* no *desktop* dos computadores clientes, que executará a aplicação para acesso à mesma,
- A aplicação terá dois interfaces um para os utilizadores, apenas para consulta e outro para administradores, para consulta e edição,
- Os documentos consultados no interface da aplicação estarão disponíveis para Download em formato PDF.

ISTEC

(Página intencionalmente deixada em branco)

4 Implementação da Solução

Depois do estudo realizado no Capítulo 2, Estado de Arte, onde foram abordadas técnicas, apresentados alguns *softwares* de virtualização, entre outros, foi necessário escolher um que cumprisse os requisitos do projeto de forma a implementar essa solução. Ao longo deste capítulo é apresentado a forma de como foi implementado laboratório da Biblioteca Virtual, numa máquina física, descrevendo as características da máquina. Em seguida serão descritas as características dos servidores virtuais e todos os passos necessários para a implementação da solução, bem como todas as configurações necessárias. Posteriormente iremos demonstrar como foi construída uma base de dados de modo a armazenar todos os conteúdos da Biblioteca. Por último, iremos descrever a criação e as características de uma aplicação para servir de interface com a base de dados.

4.1 Máquina Física

A solução da Biblioteca Virtual foi implementada sobre uma máquina física, que neste caso foi o computador portátil do aluno.

Tem então como características principais:

- Toshiba Tecra Z40-A-17R
- Processador: Intel Core i7-4600U vPro (CPU 2.10GHz-2.70GHz)
- Memória RAM: 16GB
- Armazenamento: Disco SSD 512GB
- Sistema Operativo: Microsoft Windows 10 Profissional

Windows edition

Windows 10 Pro

© 2015 Microsoft Corporation. All rights reserved.



System

Processor:	Intel(R) Core(TM) i7-4600U CPU @ 2.10GHz 2.70 GHz
Installed memory (RAM):	16.0 GB
System type:	64-bit Operating System, x64-based processor
Pen and Touch:	No Pen or Touch Input is available for this Display

Figura 14: Características da máquina física, Fonte própria

4.1.1 Configuração da Máquina Física

De modo a preparar o servidor para a instalação da solução escolhida, foi necessário primeiro efetuar algumas configurações no mesmo.

Iremos utilizar uma solução de virtualização em *hypervisor* do tipo 2, VMware Workstation 12 Professional, instalado sobre o Sistema Operativo da máquina física, o que permitirá que tenhamos mais flexibilidade na gestão deste laboratório:

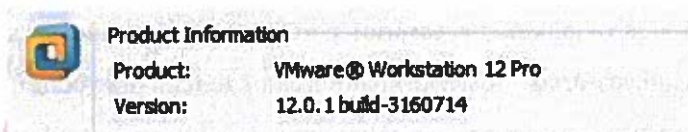


Figura 15: VMware Workstation 12 Pro, Fonte própria

4.2 Servidores Virtuais

4.2.1 Visão Geral da Solução

Com o objetivo de testar e explorar a solução, foram implementadas três servidores virtuais, conforme mostra a tabela 2:

Tabela 2: Máquinas Virtuais, Fonte própria

Sistema Operativo	Microsoft Windows Server 2012 R2 Standard	Microsoft Windows Server 2012 R2 Datacenter	Microsoft Windows Server 2012 R2 Datacenter
Nome do Servidor Virtual	DCPG1946	SQLPG1946	VMMPG1946
Armazenamento	60GB	250GB	250GB 150GB
CPUs/Cores	1/1	1/1	½
Memória RAM	2GB	2GB	6GB
Configuração da Rede	IP: 110.1.1.2/16 GW: 110.1.1.15 DNS Server: 127.0.0.1	IP: 110.1.1.3/16 GW: 110.1.1.2 DNS Server: 110.1.1.2	IP: 110.1.1.4/16 GW: 110.1.1.2 DNS Server: 110.1.1.2

Para a criação deste laboratório foi definida uma estrutura da rede virtual a ser implementada, como demonstrado na Figura 16.

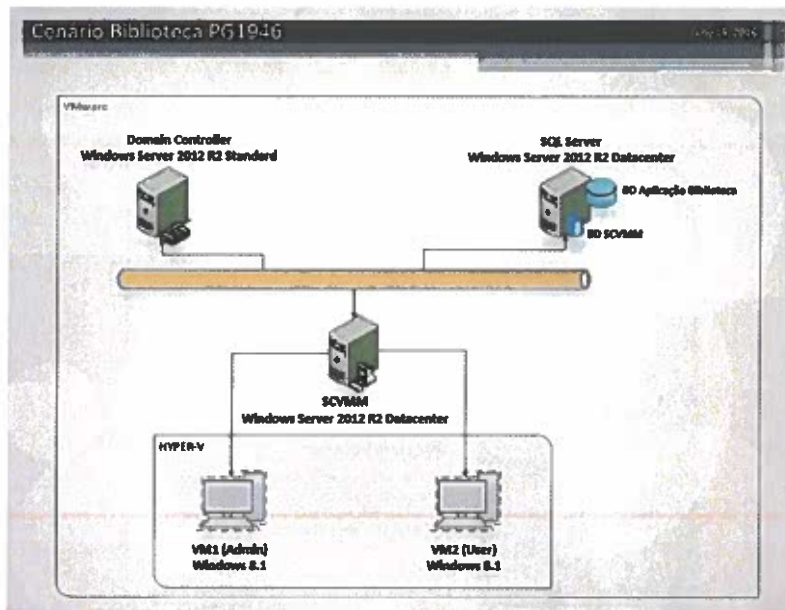


Figura 16: Estrutura da Rede Virtual, Fonte própria

4.2.2 Criação dos Servidores Virtuais

Nesta secção será explicado ao detalhe os passos necessários a serem efetuados para a criação dos servidores virtuais necessários para este laboratório

Estes servidores irão correr sobre a máquina física, sobre o VMware Workstation 12 Professional.

4.2.2.1 Servidor Controlador de Domínio

A este servidor será dado o nome de DCPG1946, foi iniciado o processo de configuração do servidor selecionando a opção “New Virtual Machine”:

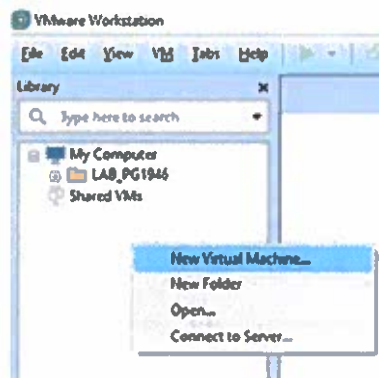


Figura 17: VMware criação de nova máquina virtual, Fonte própria

É apresentado um guia de instalação passo a passo para a criação da nova máquina virtual, no qual escolhemos a opção “Custom”:



Figura 18: Assistente de criação de máquinas virtuais VMware, Início, Fonte própria

No seguinte passo, selecionamos a imagem do Sistema Operativo que pretendemos instalar no servidor virtual, neste caso selecionámos a imagem da versão licenciada Standard do Microsoft Window Server 2012 R2:

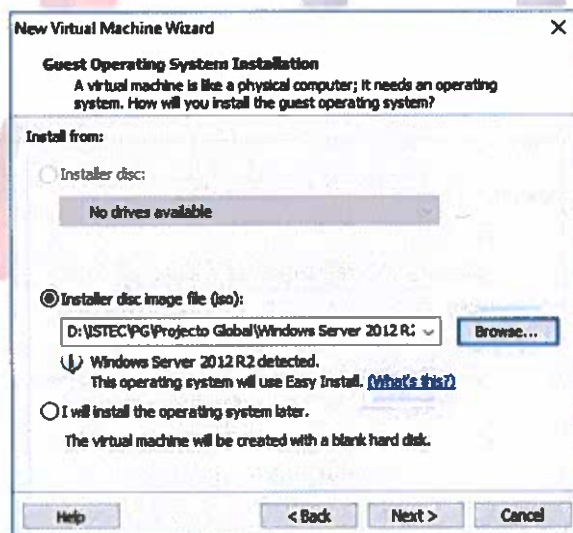


Figura 19: Assistente de criação de máquinas virtuais VMware, SO, Fonte própria

De seguida, vamos introduzir a chave de licenciamento da Microsoft e a selecionamos a versão do Sistema Operativo do qual seleccionámos a imagem no passo anterior para instalação:

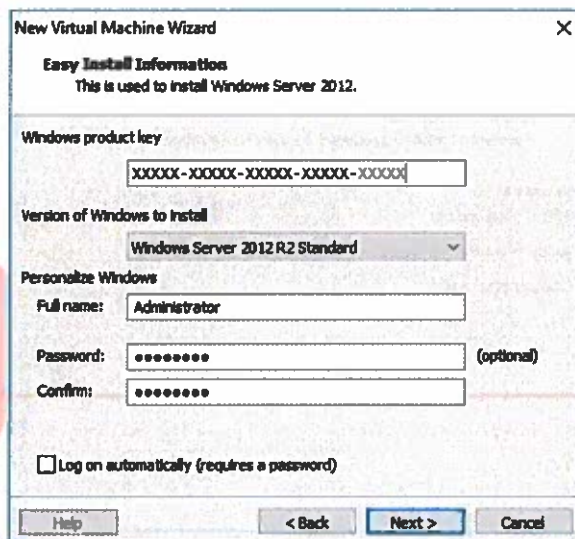


Figura 20: Assistente de criação de máquinas virtuais VMware, Licenciamento, Fonte própria

Em seguida, é definido o nome do servidor virtual pelo qual a mesma será referenciada no VMware, DCPG1946, seleccionamos também no disco da máquina anfitriã a localização onde pretendemos que a mesma seja guardada:

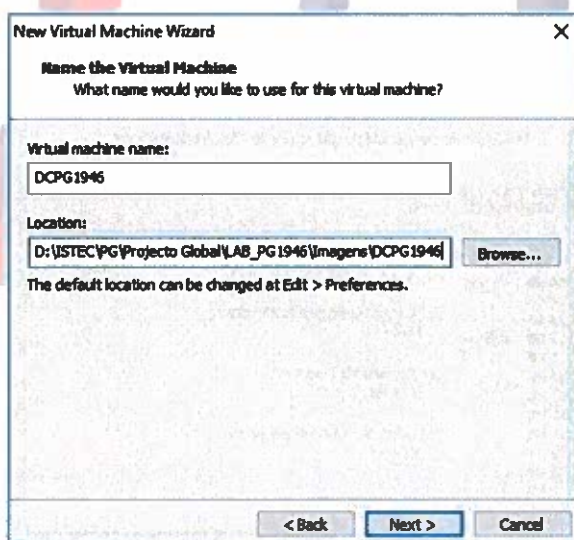


Figura 21: Assistente de criação de máquinas virtuais VMware, Nome do servidor virtual, Fonte própria

No passo seguinte, selecionamos o número de processadores e o número de cores do processador atribuídos ao servidor virtual, onde teremos de ter em conta os recursos necessários para este servidor virtual, bem como os recursos disponíveis na máquina anfitriã.

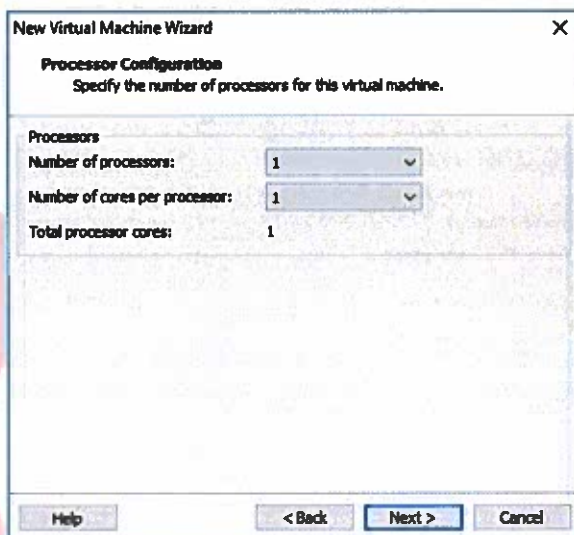


Figura 22: Assistente de criação de máquinas virtuais VMware, Processador, Fonte própria

Seguidamente, vamos definir a memória RAM que será alocada ao servidor virtual, aqui uma vez mais temos de ter em conta os recursos que este servidor virtual necessita, bem como os recursos que dispomos na máquina anfitriã:

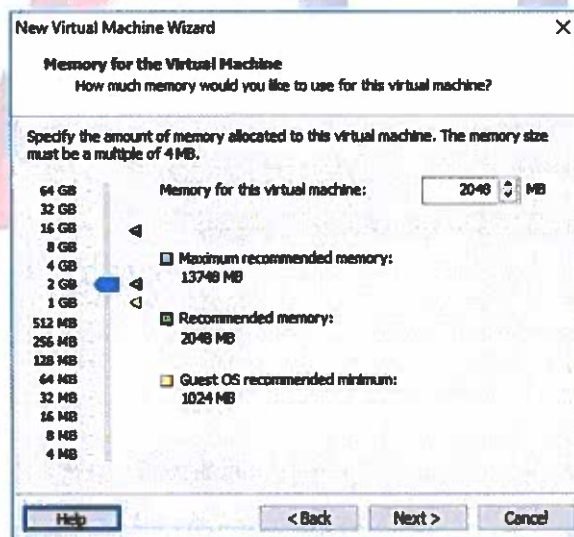


Figura 23: Assistente de criação de máquinas virtuais VMware, Memória, Fonte própria

Neste passo, é definido o tipo de interface de rede que pretendemos configurar no servidor virtual, neste caso iremos inicialmente selecionar a configuração do interface de rede do tipo NAT, de modo a podermos instalar todos os atualizações necessárias no servidor virtual:

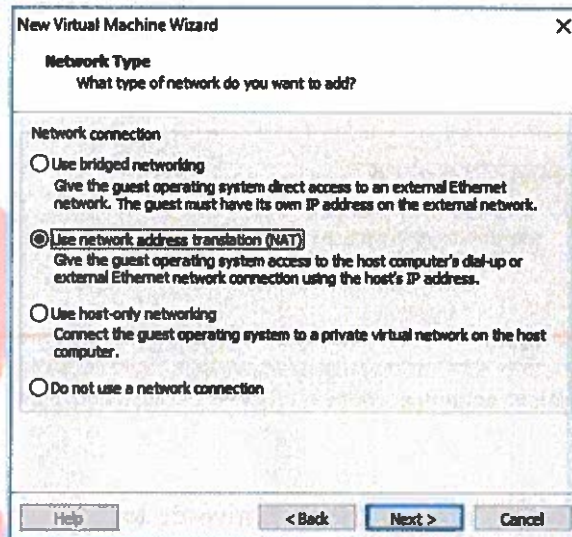


Figura 24: Assistente de criação de máquinas virtuais VMware, Interface de Rede, Fonte própria

No passo de criação do disco virtual, vamos definir o tamanho do mesmo e selecionamos a alocação imediata do espaço definido, escolher esta opção torna o servidor virtual mais rápido, mas obriga a que exista esse espaço já disponível na máquina anfitriã. Vamos selecionar também neste passo se pretendemos que o disco virtual seja formado por múltiplos ficheiros, ou apenas num único:

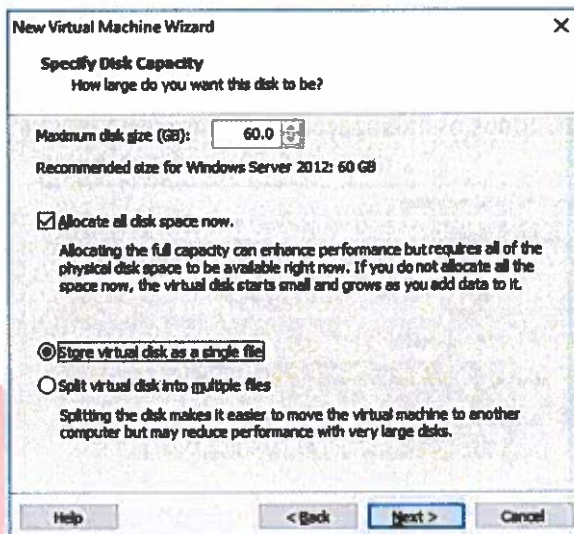


Figura 25: Assistente de criação de máquinas virtuais VMware, Espaço de Armazenamento, Fonte própria

Por último, é apresentado um sumário descritivo de toda a configuração com a qual o servidor virtual será criado:

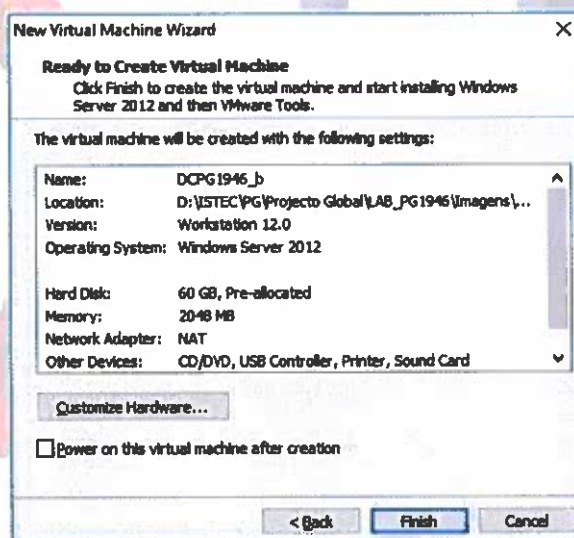


Figura 26: Assistente de criação de máquinas virtuais VMware, Resumo de Instalação, Fonte própria

Antes de ser iniciado o servidor virtual, teremos ainda de adicionar um segundo interface de rede, de modo a que seja criada uma rede interna de comunicação entre os servidores virtuais.

Para tal seleccionamos a opção “Settings” do servidor virtual criado e depois seleccionamos a opção “Add” e escolhemos o “Network Adapter”:

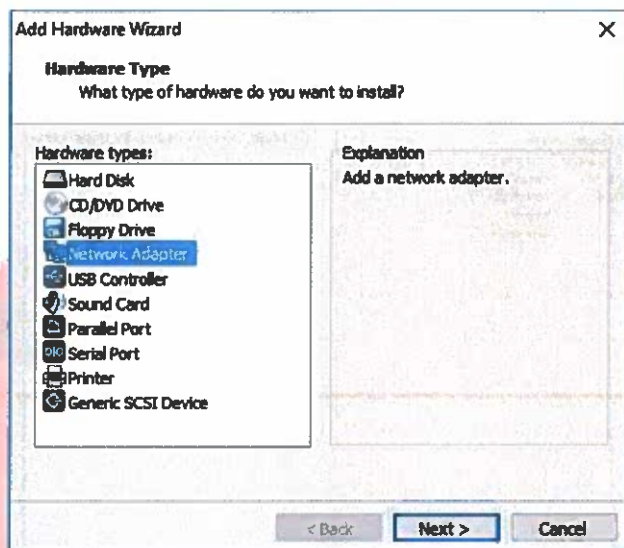


Figura 27: VMware configuração de *hardware*, Interface de Rede 1, Fonte própria

Agora é selecionado então o tipo de interface de rede pretendido, neste caso seleccionamos a opção “Host-only”:

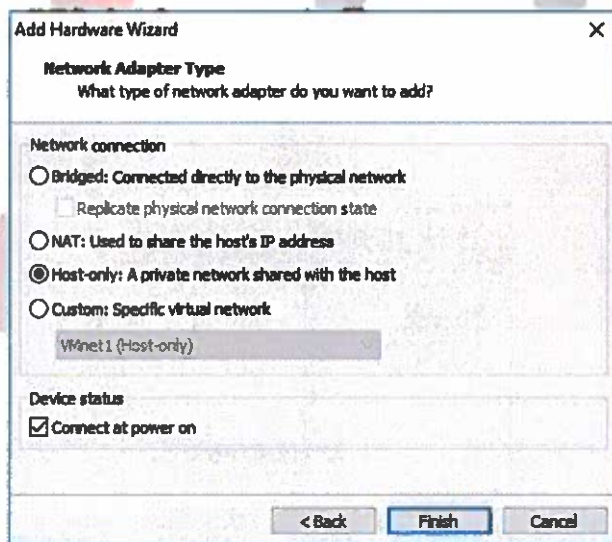


Figura 28: VMware configuração de *hardware*, Interface de Rede 2, Fonte própria

No final o resultado deverá ser o apresentado na imagem seguinte:

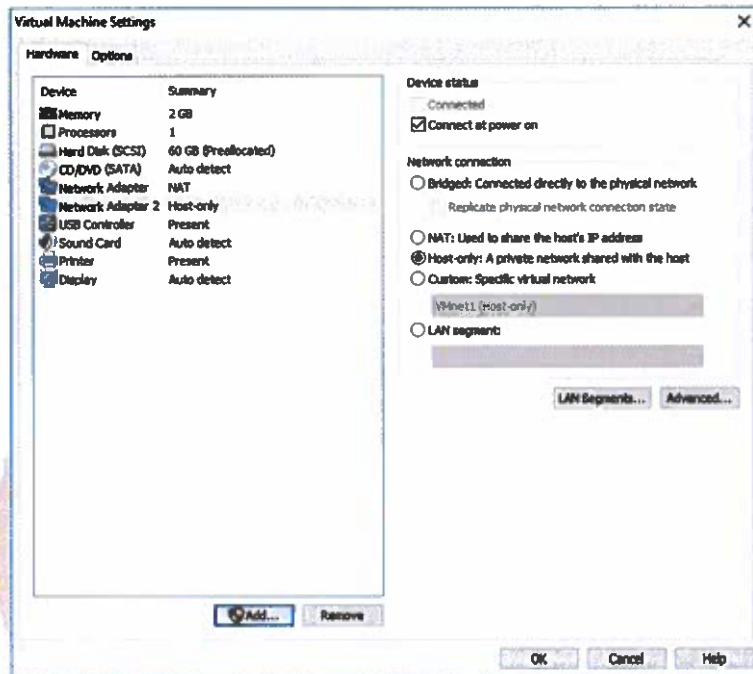


Figura 29: VMware configuração de hardware, Interface de Rede 3, Fonte própria

4.2.2.2 Servidor SQL

Ao servidor seguinte será dado o nome de SQLPG1946, tal como na criação do servidor DCPG1946, foi iniciado o processo de configuração do servidor selecionando a opção “New Virtual Machine”:

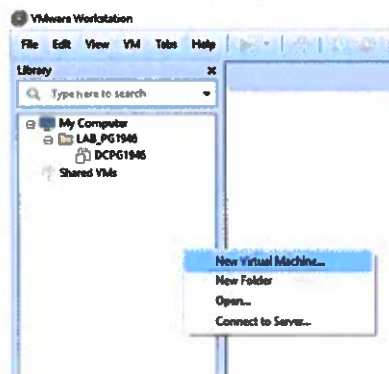


Figura 30: VMware criação de nova máquina virtual 2, Fonte própria

Após a seleção da opção anterior, é apresentado um guia de instalação passo a passo para a criação da nova máquina virtual, no qual escolhemos a opção “Custom”:



Figura 31: Assistente de criação de máquinas virtuais VMware, Início 2, Fonte própria

No passo seguinte, selecionamos a imagem do SO que pretendemos instalar no servidor virtual, neste caso seleccionámos a imagem da versão Datacenter do Microsoft Window Server 2012 R2, ao contrário do SO instalado no servidor virtual DCPG1946, este é uma versão “Evaluation” da Microsoft, que nos dará um período experimental de 180 dias, sem necessidade de ativação do licenciamento:

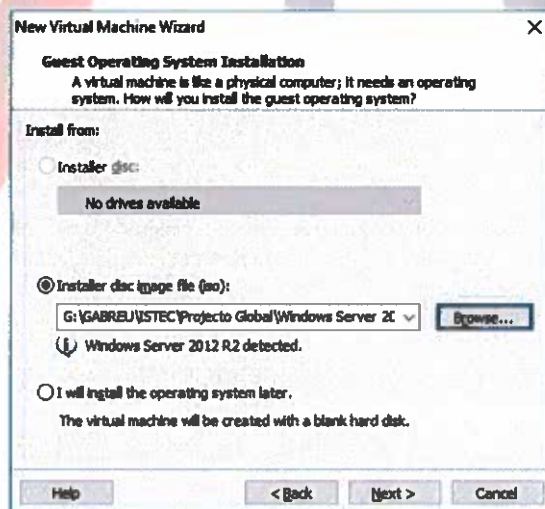


Figura 32: Assistente de criação de máquinas virtuais VMware, SO 2, Fonte própria

Em seguida, é definido o nome do servidor virtual pelo qual a mesma será referenciada no VMware, neste caso será SQLPG1946, selecionamos também no disco da máquina anfitriã a localização onde pretendemos que a mesma seja guardada:

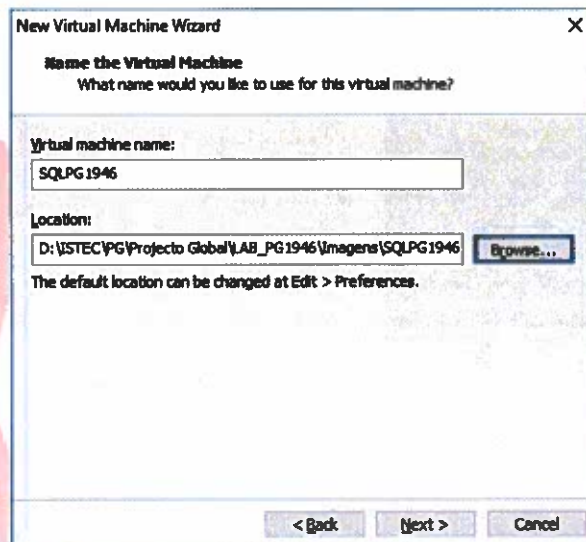


Figura 33: Assistente de criação de máquinas virtuais VMware, Nome do servidor virtual 2, Fonte própria

Seguindo para o próximo passo, selecionamos o número de processadores e o número de cores do processador atribuídos ao servidor virtual, onde teremos de ter em conta os recursos necessários para este servidor virtual, bem como os recursos ainda disponíveis na máquina anfitriã.

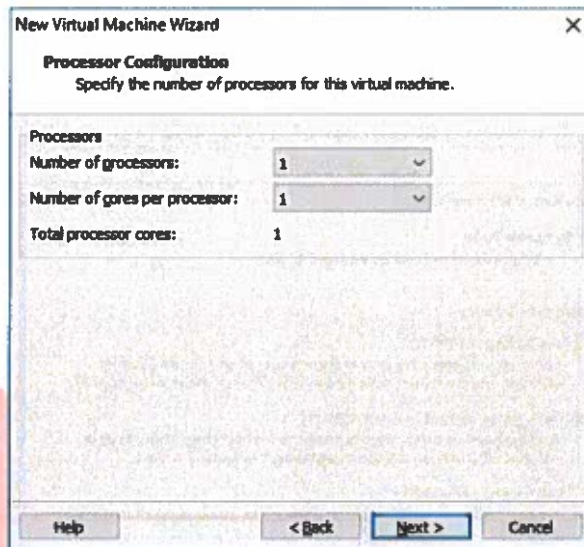


Figura 34: Assistente de criação de máquinas virtuais VMware, Processador 2, Fonte própria

De Seguida, vamos definir a memória RAM que será alocada ao servidor virtual, aqui uma vez mais temos de ter em conta os recursos que este servidor virtual necessita, bem como os recursos que ainda dispomos na máquina anfitriã:

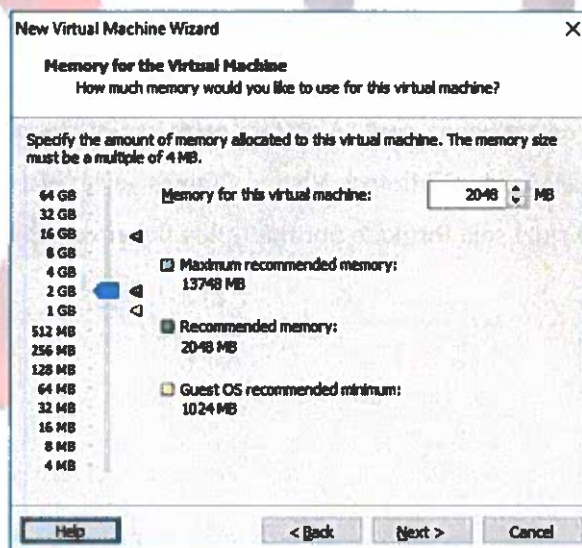


Figura 35: Assistente de criação de máquinas virtuais VMware, Memória 2, Fonte própria

Neste passo, é definido o tipo de interface de rede que pretendemos configurar no servidor virtual, pelo que iremos inicialmente selecionar a configuração do interface de rede do tipo NAT, de modo a podermos instalar todos os atualizações necessárias no servidor virtual:

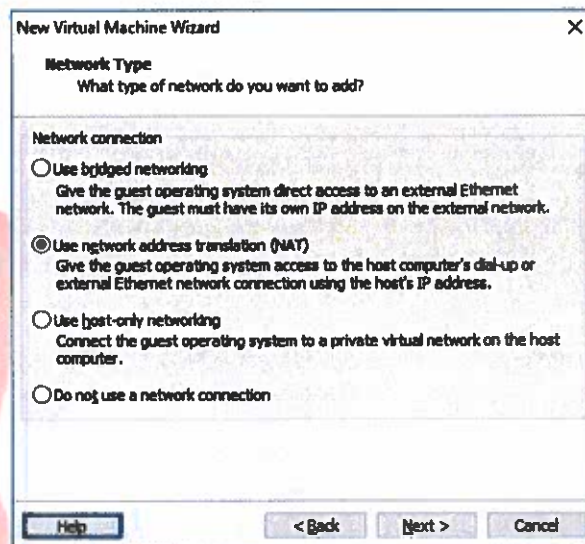


Figura 36: Assistente de criação de máquinas virtuais VMware, Interface de Rede 2, Fonte própria

Agora no passo de criação do disco virtual, vamos definir o tamanho do mesmo e não selecionamos a alocação imediata do espaço no armazenamento físico da máquina anfitriã. Foi definido um espaço maior de armazenamento pois será neste servidor virtual que ficarão alojadas as bases de dados do SCVMM e da Biblioteca Virtual. Vamos selecionar também neste passo, se pretendemos que o disco virtual seja formado por múltiplos ficheiros, ou apenas num único:

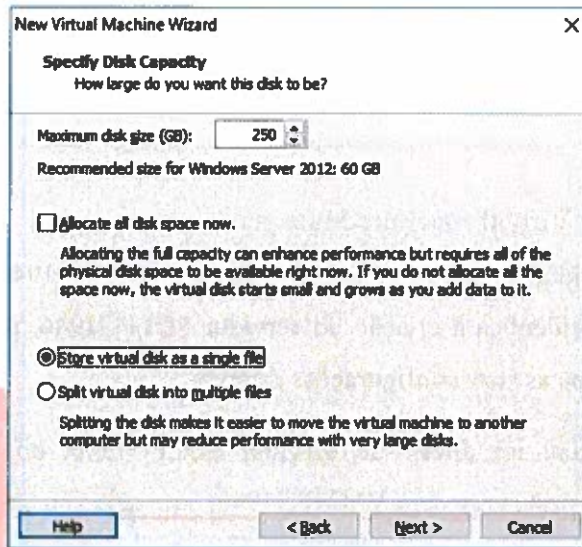


Figura 37: Assistente de criação de máquinas virtuais VMware, Espaço de Armazenamento 2, Fonte própria

No último passo, é apresentado um sumário descritivo de toda a configuração, com a qual o servidor virtual será criado:

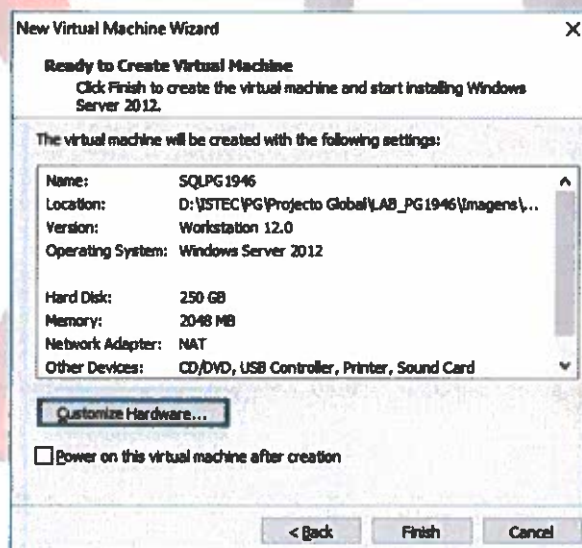


Figura 38: Assistente de criação de máquinas virtuais VMware, Resumo de Instalação 2, Fonte própria

Tal como foi realizado para o servidor DCPG1946, antes de ser também iniciado este servidor virtual, teremos ainda de adicionar um segundo interface de rede, de modo a que seja

criada uma rede interna de comunicação entre os servidores virtuais. Para tal bastará seguirmos os mesmos passos anteriormente descritos.

4.2.2.3 Servidor Virtual Machine Manager

A este último servidor virtual, será dado o nome de VMMPG1946, a configuração deste servidor é em quase tudo idêntica à criação do servidor SQLPG1946, pelo que iremos apenas apresentar os passos em que as suas configurações diferem.

Este servidor virtual irá diferir do servidor SQLPG1946, no nome pelo qual será referenciado no VMware, neste caso será VMMPG1946, aqui selecionamos também no disco da máquina anfitriã a localização onde pretendemos que a configuração da mesma seja guardada:

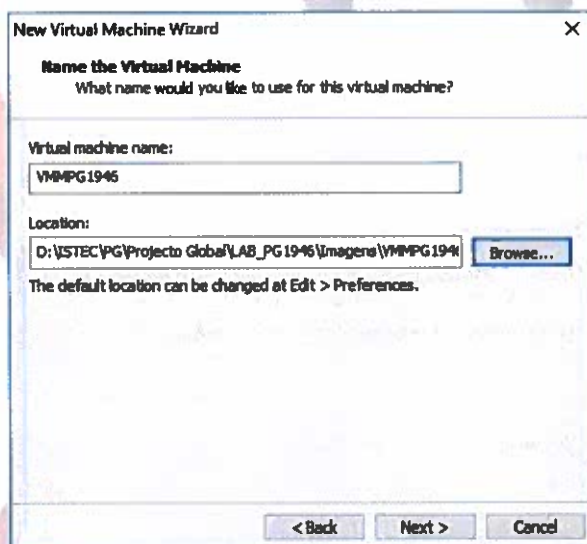


Figura 39: Assistente de criação de máquinas virtuais VMware, Nome do servidor virtual 3, Fonte própria

O passo seguinte em que a configuração deste servidor virtual difere, é na atribuição do número de processadores e do número de cores do processador que lhe iremos atribuir, em que neste perante as funcionalidades que irá desempenhar irá necessitar de mais recursos, pelo que iremos atribuir-lhe um processador, mas ao invés do que configurámos nos restantes servidores virtuais, vamos atribui-lhe dois cores.

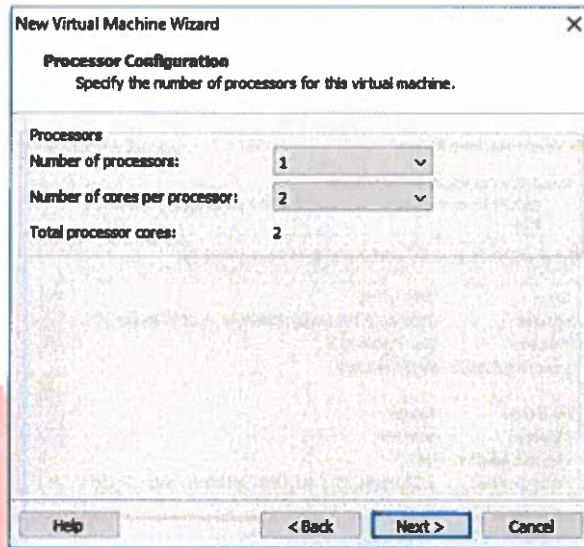


Figura 40: Assistente de criação de máquinas virtuais VMware, Processador 3, Fonte próprio

Tal como já referido no passo anterior, perante as funcionalidades que este servidor virtual irá desempenhar, iremos atribui-lhe um maior número de recursos da máquina física, pelo que no passo onde definimos a memória RAM alocada a este servidor, ao contrário do que foi definido nos restantes servidores, iremos alocar-lhe uma memória de 6144MB:

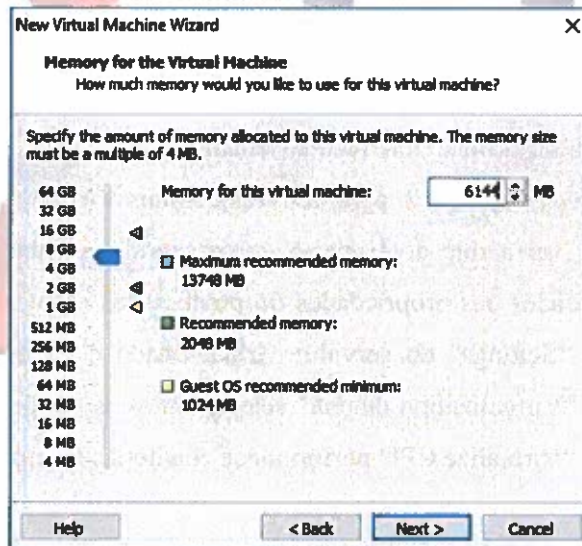


Figura 41: Assistente de criação de máquinas virtuais VMware, Memória 3, Fonte própria

Por último, é apresentado um sumário descritivo de toda a configuração, com a qual o servidor virtual será criado:

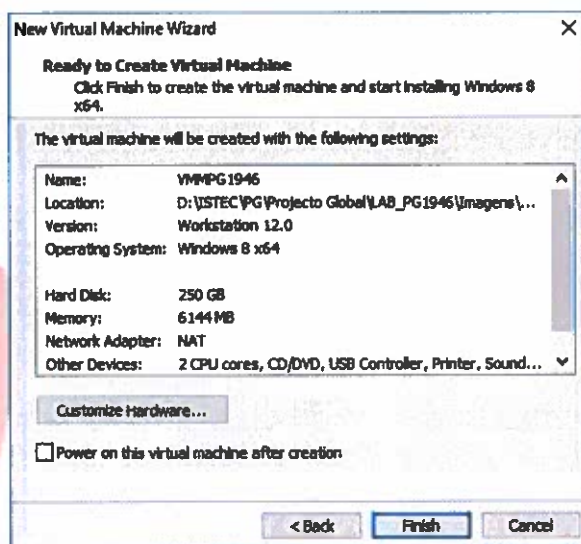


Figura 42: Assistente de criação de máquinas virtuais VMware, Resumo de Instalação 3, Fonte própria

Assim como realizado para os servidores DCPG1946 e SQLPG1946, antes de ser também iniciado este servidor virtual, teremos ainda de adicionar um segundo interface de rede, de modo a que seja criada uma rede interna de comunicação entre os servidores virtuais. Para tal bastará seguirmos os mesmos passos anteriormente descritos.

Neste servidor iremos utilizar um recurso chamado de *Nested Virtualization*, em que iremos executar um *hypervisor* do tipo 2, para isso iremos mais à frente na secção Configuração dos Servidores Virtuais ativar a role do Hyper-V neste servidor virtual. Para tal é necessário ativarmos duas funcionalidades nas propriedades do processador deste servidor virtual, iremos então seleccionar a opção “Settings” do servidor virtual criado e depois escolhemos a opção “Processors” e no campo “Virtualization engine” seleccionamos as opções “Virtualize Intel VT-x/EPT or AMD-V/RVI” e “Virtualize CPU performance counters”: (Siemens, 2014)

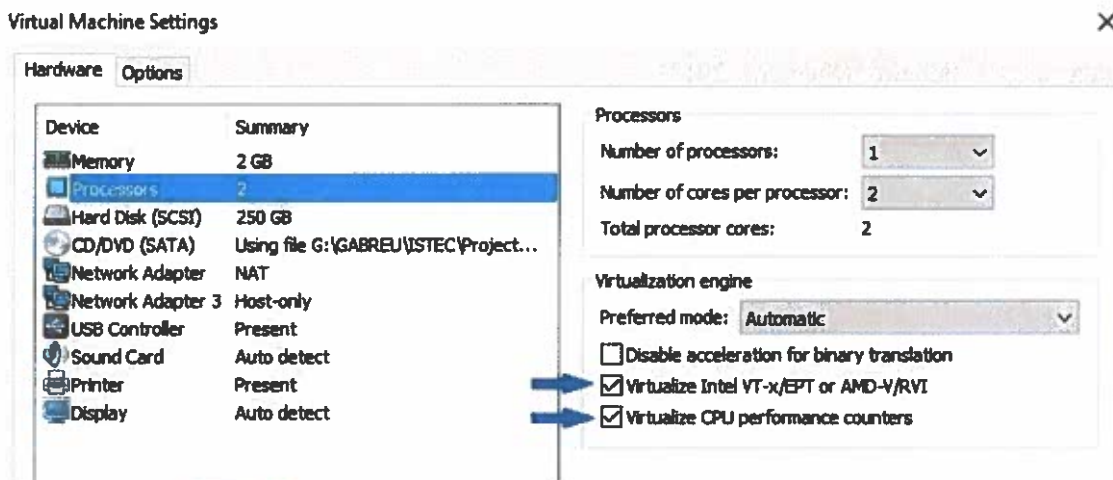


Figura 43: VMware configuração de hardware, Processador, Fonte própria

Para que possamos usar o recurso *Nested Virtualization* neste servidor virtual, ativando a role do Hyper-V, teremos ainda de executar mais um passo. Na máquina física acedemos à pasta onde definimos que iria ser guardada a configuração deste servidor virtual, dentro dessa pasta está um ficheiro cuja extensão (*.vmx), neste caso o nome do ficheiro é VMMPG1946.vmx, em que vamos abri-lo com o Notepad: (Siemens, 2014)

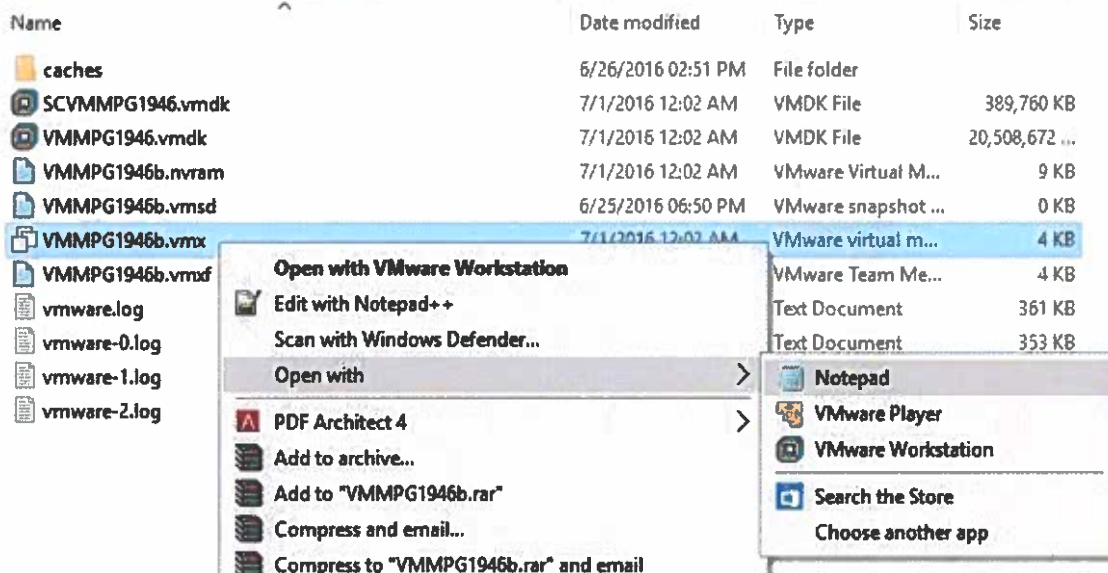


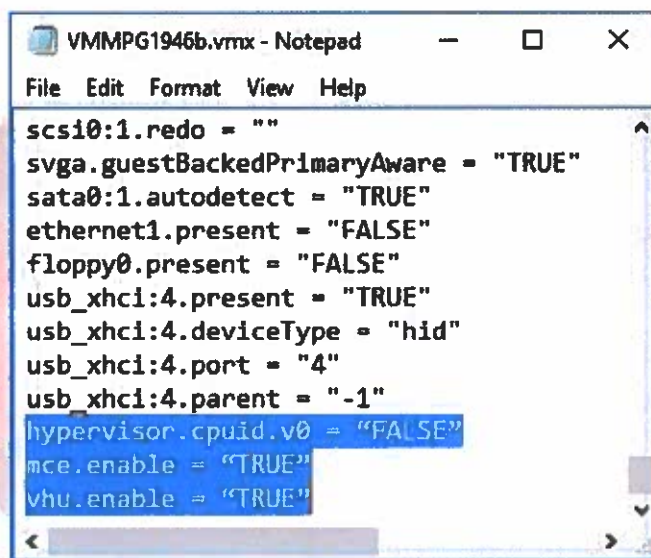
Figura 44: Editar ficheiro (*.vmx), Fonte Própria

Após abrirmos o ficheiro no “Notepad”, vamos adicionar-lhe o seguinte código e em seguida salvar o ficheiro: (Siemens, 2014)

hypervisor.cpuid.v0 = “FALSE”

mce.enable = “TRUE”

vhv.enable = “TRUE”



```
scsi0:1.redo = ""
svga.guestBackedPrimaryAware = "TRUE"
sata0:1.autodetect = "TRUE"
ethernet1.present = "FALSE"
floppy0.present = "FALSE"
usb_xhci:4.present = "TRUE"
usb_xhci:4.deviceType = "hid"
usb_xhci:4.port = "4"
usb_xhci:4.parent = "-1"
hypervisor.cpuid.v0 = "FALSE"
mce.enable = "TRUE"
vhv.enable = "TRUE"
```

Figura 45: Editar ficheiro (*.vmx) 2, Fonte Própria

4.2.3 Configuração dos Servidores Virtuais

Nesta secção será explicado ao detalhe os passos necessários a serem efetuados para a configuração dos servidores virtuais, bem como a instalação do *software* necessárias às funções que cada servidor irá desempenhar neste laboratório.

Para criar uma rede privada num domínio, o primeiro passo após a criação dos servidores é definir o endereçamento IP de cada servidor dentro da rede privada. Em seguida será a atribuição do nome pelo qual cada servidor será conhecido dentro do domínio. Depois teremos de escolher qual o servidor que pretendemos promover a controlador de domínio, em que neste caso foi escolhido o DCPG1946, e neste adicionamos a role da *Active Directory Domain Services*. Após ter sido adicionada esta role ao servidor ser-nos-á questionado pelo sistema se pretendemos promover este servidor a controlador de domínio. Iremos então promover o referido servidor a controlador de domínio, através do assistente de configuração do *Active Domain Services*, no qual iremos também criar o respetivo domínio. No assistente de configuração do *Active Domain Services*, será também selecionada a opção de instalação da role de DNS server. Por fim depois da criação do domínio e a promoção do servidor DCPG1946 a controlador de domínio, o passo final será adicionar os restantes servidores, SQLPG1946 e VMMPG1946 ao domínio criado.

4.2.3.1 Configuração das Interfaces de Rede

Após a instalação do SO em cada um dos servidores virtuais DCPG1946, SQLPG1946 e VMMPG1946, iremos configurar as interfaces de rede de acordo com o planeado na secção deste capítulo Visão Geral da Solução.

Dado que os passos para configuração das interfaces de rede são iguais nos três servidores virtuais, mas explicá-los na mesma secção, demonstrando apenas em separado a configuração de IPs, que é diferente em cada servidor.

Para iniciarmos a configuração, já com o servidor virtual ligado, dentro do mesmo abrimos na linha de comandos o menu de configuração “Network Connections”, através do comando “ncpa.cpl”:

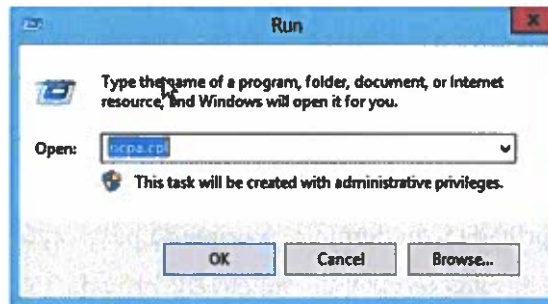


Figura 46: Abrir janela dos Interface de Rede, Fonte Própria

Através do comando anterior abre-se a janela “Network Connections”, em que podemos visualizar as interfaces de rede que estão configuradas no servidor virtual.



Figura 47: Janela dos Interfaces de Rede, Fonte Própria

Em seguida, iremos renomear os interfaces de rede existentes, de modo a facilitar no reconhecimento de cada uma das redes, privada e pública:

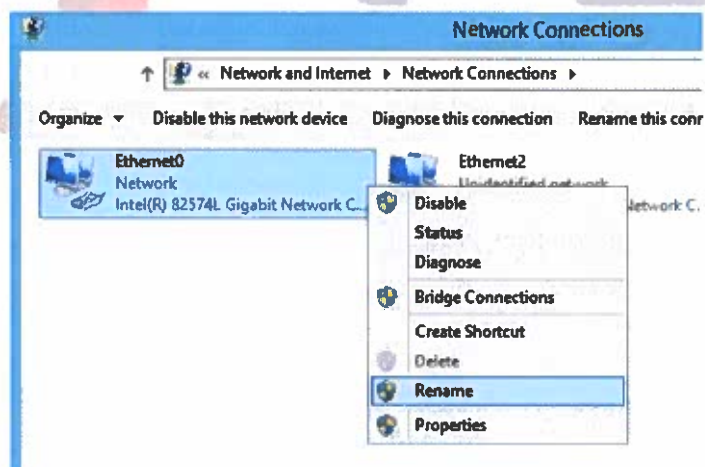


Figura 48: Renomear Interfaces de Rede, Fonte Própria

Iremos atribuir à interface de rede pública, com acesso à internet, o nome de NAT e à interface de rede privada, para comunicação entre as diversas máquinas no domínio, o nome de LAN:

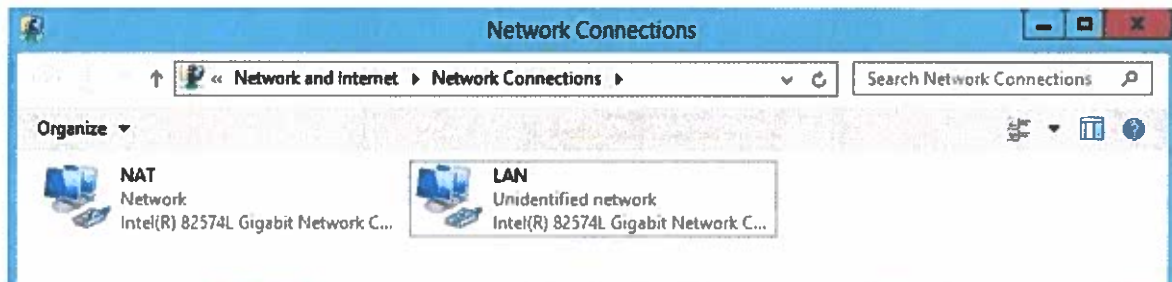


Figura 49: Renomear Interfaces de Rede 2, Fonte Própria

No passo seguinte, vamos alterar o *binding* das interfaces de rede de modo a definir a sua prioridade nas rotas de cada servidor. Para tal na janela “Network Connections” carregamos na tecla ALT, para termos acesso à barra de menu e seleccionamos “Advanced” > “Advanced Settings...”:

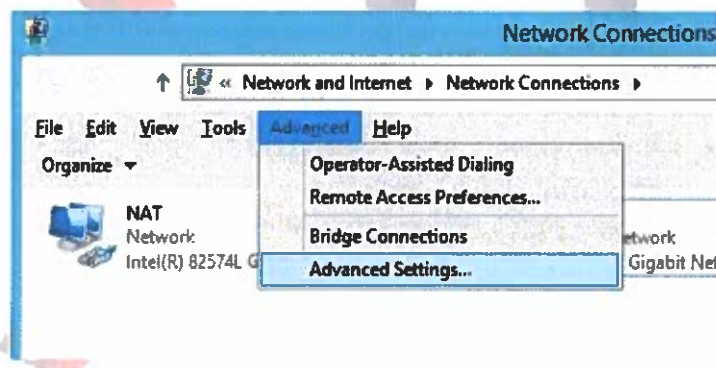


Figura 50: Alterar Hierarquia dos Interfaces de Rede, Fonte Própria

Devemos definir a hierarquia das interfaces de rede de cada servidor virtual da seguinte forma: 1º LAN / 2º NAT. Para tal devemos seleccionar uma das interfaces de rede e com as setas do lado direito da caixa “Connections” coloca-las pela ordem pretendida:

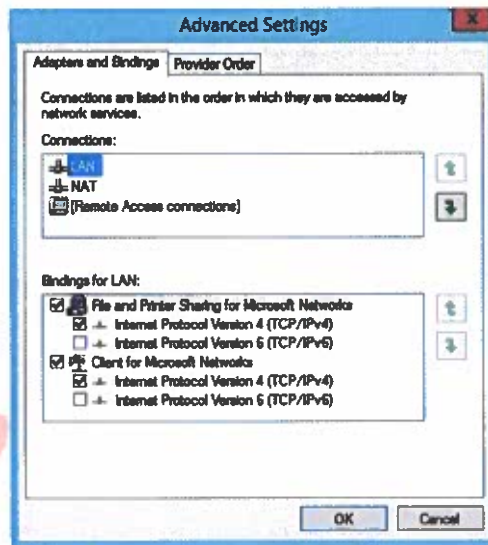


Figura 51: Alterar Hierarquia dos Interfaces de Rede 2, Fonte Própria

Vamos agora atribuir as definições das interfaces de rede, no caso da interface de rede à qual demos o nome de NAT, iremos deixá-la com os parâmetros padrão em que ficará com o DHCP ativo. Iremos apenas aceder às definições da interface de rede à qual demos o nome de LAN:

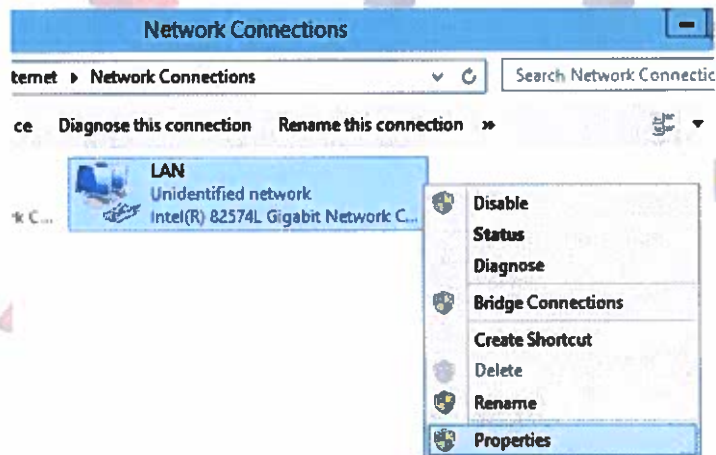


Figura 52: Propriedades do Interface de Rede, Fonte Própria

De seguida, iremos atribuir as definições da interface de rede conforme planeado para cada servidor virtual. Vamos seleccionar a opção “Internet Protocol Version 4 (TCP/IPv4)” e carregamos no botão “Properties”:

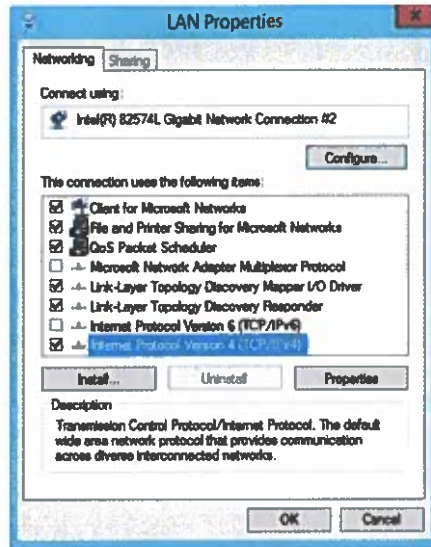


Figura 53: Propriedades do Interface de Rede 2, Fonte Própria

Por fim, definimos o endereçamento IP da interface de rede para cada servidor virtual.

No servidor DCPG1946 vamos colocar o endereçamento já previamente definido:

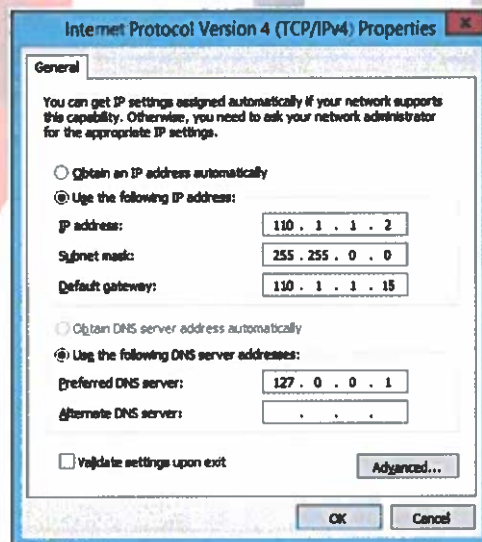


Figura 54: Alterar Endereçamento IP do Interface de Rede, Fonte Própria

No servidor seguinte, SQLPG1946, vamos colocar o seguinte endereçamento já previamente definido:

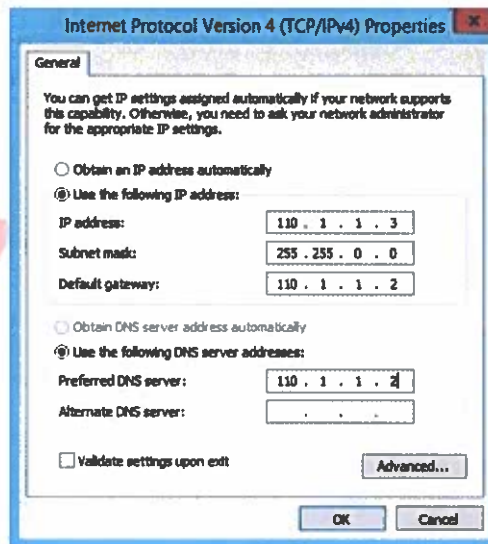


Figura 55: Alterar Endereçamento IP do Interface de Rede 2, Fonte Própria

Finalmente no servidor VMMPG1946, vamos colocar o endereçamento já previamente definido:

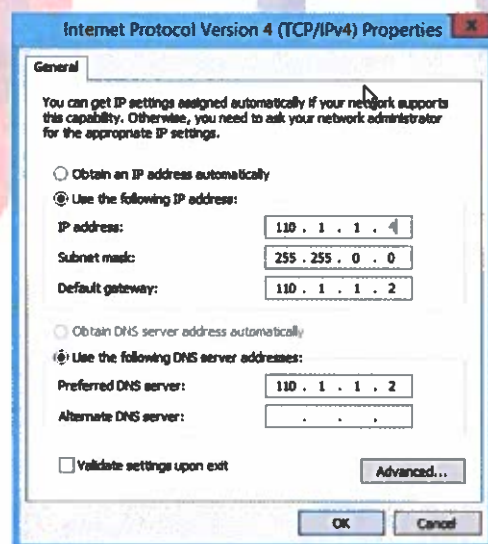


Figura 56: Alterar Endereçamento IP do Interface de Rede 3, Fonte Própria

4.2.3.2 Controlador de Domínio

Este servidor, ao qual demos o nome de DCPG1946, será o Controlador de Domínio, em que para tal iremos explicar ao detalhe os passos necessário para que o mesmo seja promovido a Controlador de Domínio, as roles do Windows que serão necessárias ativar, as GPOs que serão criadas, entre outras configurações base necessárias para desempenhar a sua função neste laboratório.

4.2.3.2.1 Alterar o nome do servidor

Neste passo vamos alterar o nome do nosso servidor virtual, pelo qual o mesmo ficará conhecido dentro da rede privada. Para tal no “Server Manager” > “Local Server” carregamos em cima do nome do servidor:



Figura 57: Server Manager, Fonte Própria

Em seguida, é aberta a janela “System Properties”, para alterarmos o nome do servidor, carregamos no botão “Change...”:



Figura 58: Alterar Nome do Servidor, Fonte Própria

Na janela seguinte, alteramos o nome padrão com o qual foi criado o servidor para o pretendido, neste caso DCPG1946 e carregamos no botão “OK”:



Figura 59: Alterar Nome do Servidor 2, Fonte Própria

4.2.3.2.2 Adicionar a Role Active Directory Domain Services

Para que possamos criar o domínio da Biblioteca Virtual e promover o servidor DCPG1946 a controlador de domínio, após a configuração do endereçamento IP no servidor, teremos de adicionar a role “Active Directory Domain Services” ao servidor.

Para dar início a este processo, dentro do servidor virtual DCPG1946, iremos ao “Service Manager” e no menu selecionamos “Manage” > “Add Roles and Features”:

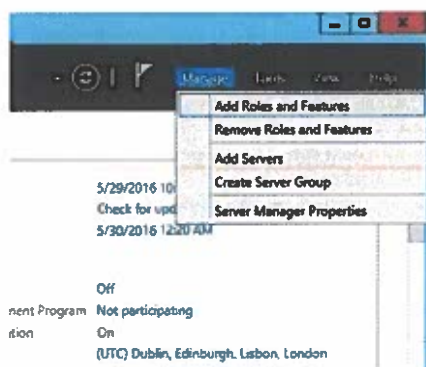


Figura 60: Server Manager 1, “Add Roles and Features”, Fonte Própria

No assistente de configuração “Add Roles and Features”, todos os passos que não são mencionados, serão mantidas as configurações padrão. Chegando à secção “Server Roles”, no quadro “Roles”, iremos selecionar a opção “Active Directory Domain Services”:

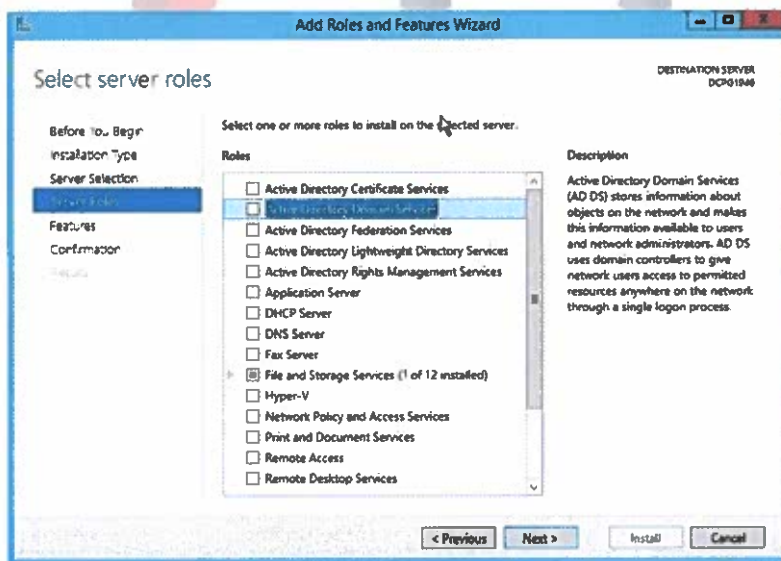


Figura 61: Assistente de configuração “Add Roles and Features”, Active Directory 1, Fonte Própria

Surgirá uma janela a indicar as “Roles” e “Features” que serão instaladas com a seleção desta opção:

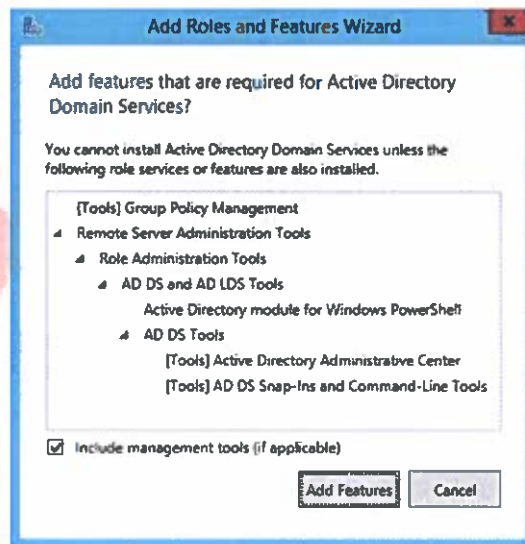


Figura 62: Assistente de configuração “Add Roles and Features”, *Active Directory 2*, Fonte Própria

Após termos carregado no botão “Add Features” da janela anterior, ficaremos com a opção “Active Directory Domain Services” selecionada:

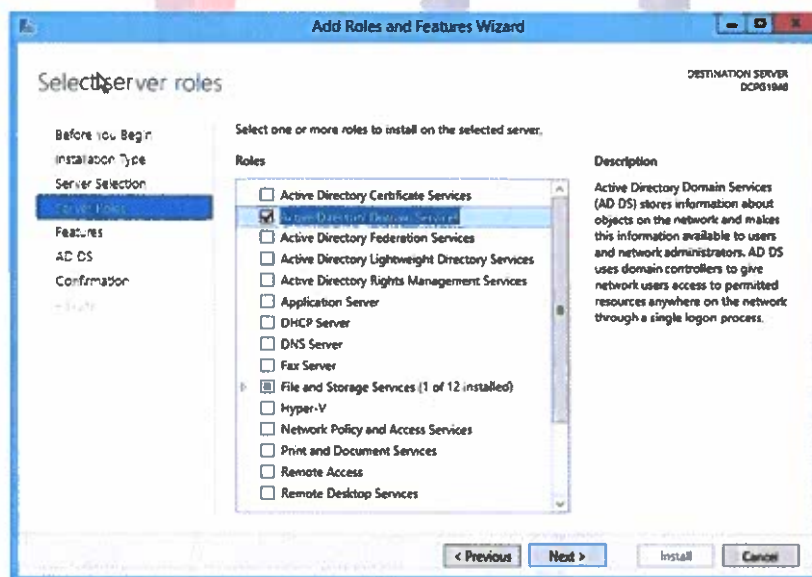


Figura 63: Assistente de configuração “Add Roles and Features”, *Active Directory 3*, Fonte Própria

No passo seguinte podemos confirmar que com a ativação da role “Active Directory Domain Services” é também adicionada a Feature “Group Policy Management”:

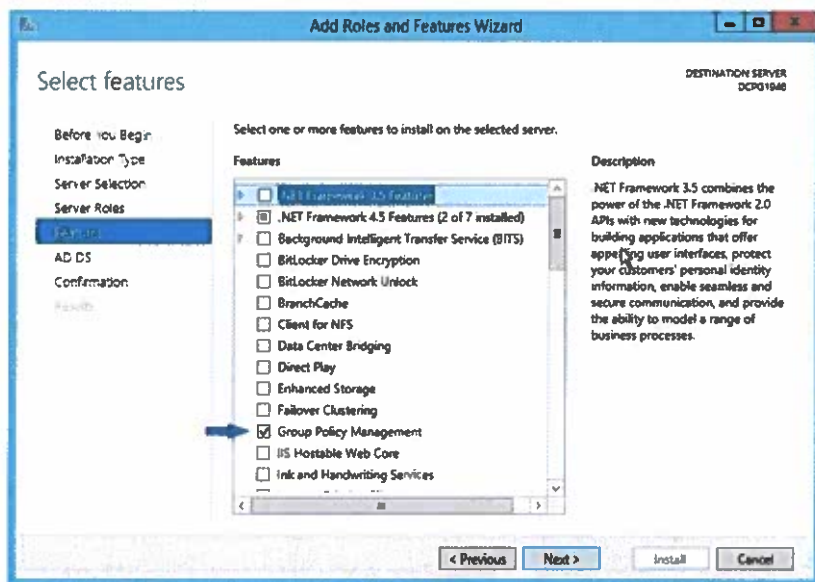


Figura 64: Assistente de configuração “Add Roles and Features”, Active Directory 4, Fonte Própria

Por fim, temos um quadro com o resumo de todas as Roles e Features que serão instaladas:

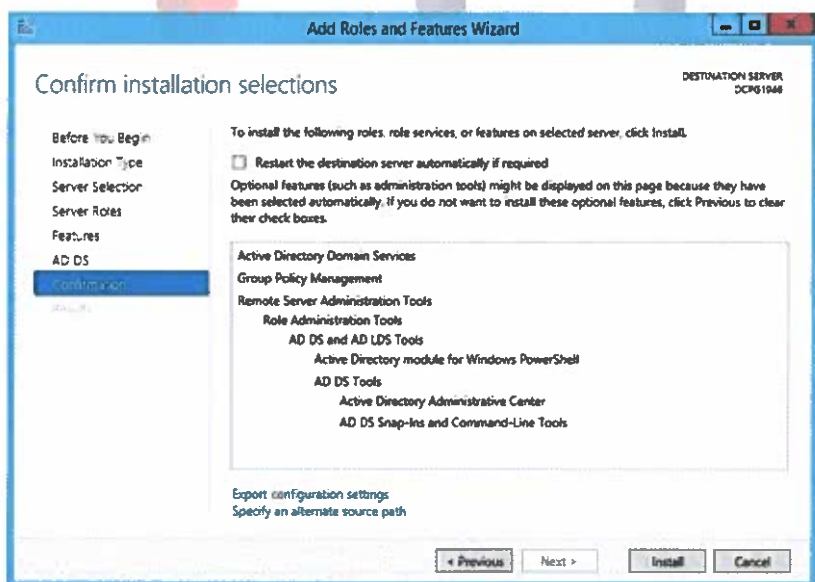


Figura 65: Assistente de configuração “Add Roles and Features”, Active Directory 5, Fonte Própria

4.2.3.2.3 Criar Domínio e Promover servidor a Controlador de Domínio

Uma vez instalada a role “Active Directory Domain Services”, iremos por fim criar o domínio e promover o servidor a controlador de domínio. Quando foi terminada a instalação da Role “Active Directory Domain Services”, na barra de menu do “Server Manager” teremos a indicação de um sinal de Aviso. Carregando no sinal de Aviso teremos a indicação de que para terminar a instalação desta role é necessário promover este servidor a controlador de domínio. Para tal vamos carregar no link “Promote this server to a domain controller”:

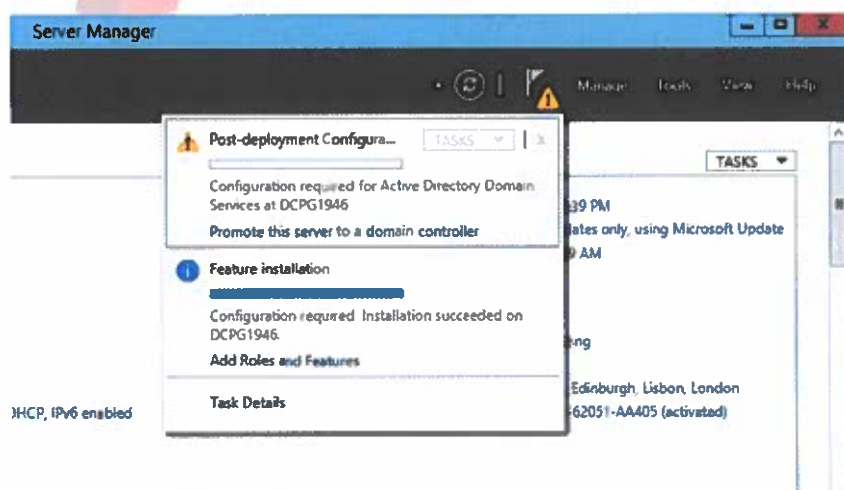


Figura 66: Promover servidor a controlador de domínio, Fonte Própria

Em seguida, é dado início ao assistente de configuração do “Active Directory Domain Services”: Através deste assistente de configuração iremos criar o domínio pretendido e promover o servidor PG1946 a controlador de domínio. No quadro do lado direito na secção “Deployment Configuration” selecionamos a opção “Add a new forest”

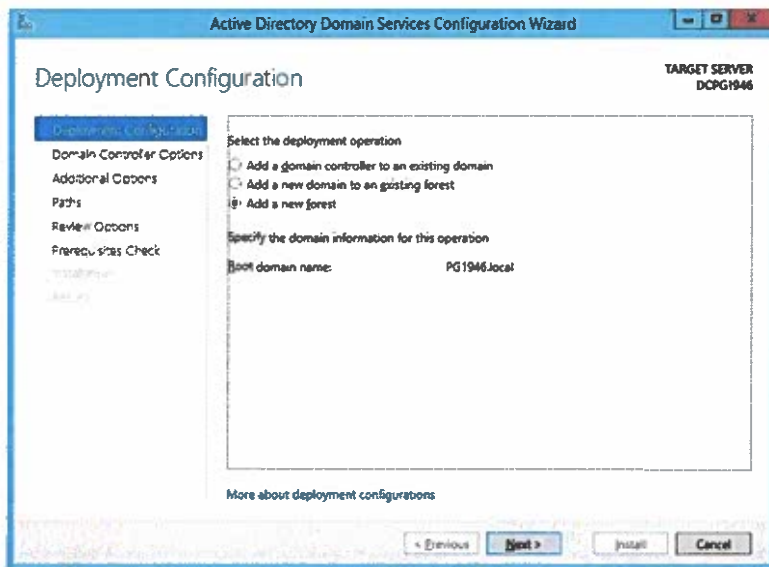


Figura 67: Assistente de configuração do “Active Directory and Domain Services”, Fonte Própria

No passo seguinte, “Domain Controller Options”, iremos selecionar o SO que o controlador de domínio tem, neste caso o Windows Server 2012 R2. Vamos também aqui selecionar a opção “Domain Name System (DNS) server”, deste modo será também instalada esta role no controlador de domínio. Ainda antes de seguirmos para o próximo passo iremos definir a password de segurança para do modo de restauro do “Directory Services”.

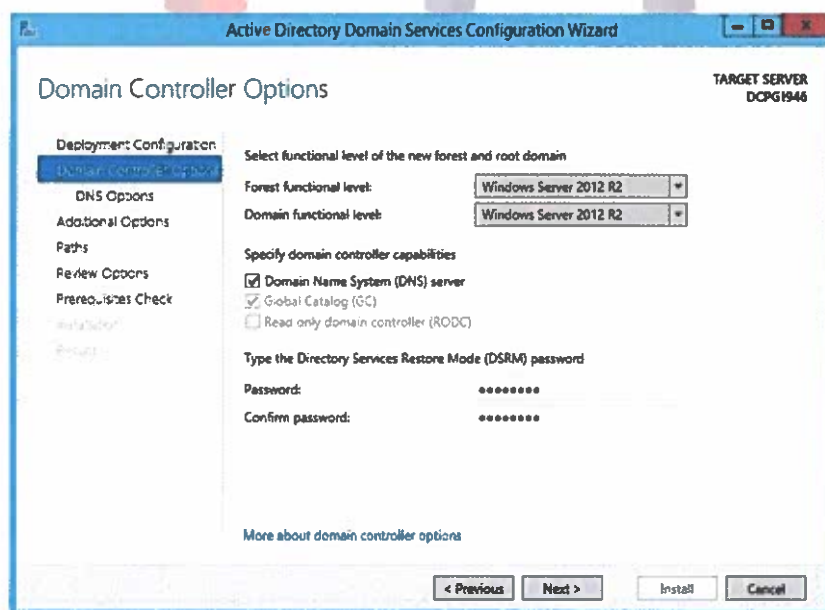


Figura 68: Assistente de configuração do “Active Directory and Domain Services” 2, Fonte Própria

Neste quadro podemos confirmar todas as configurações que serão efetuadas após concluir este assistente de configuração.

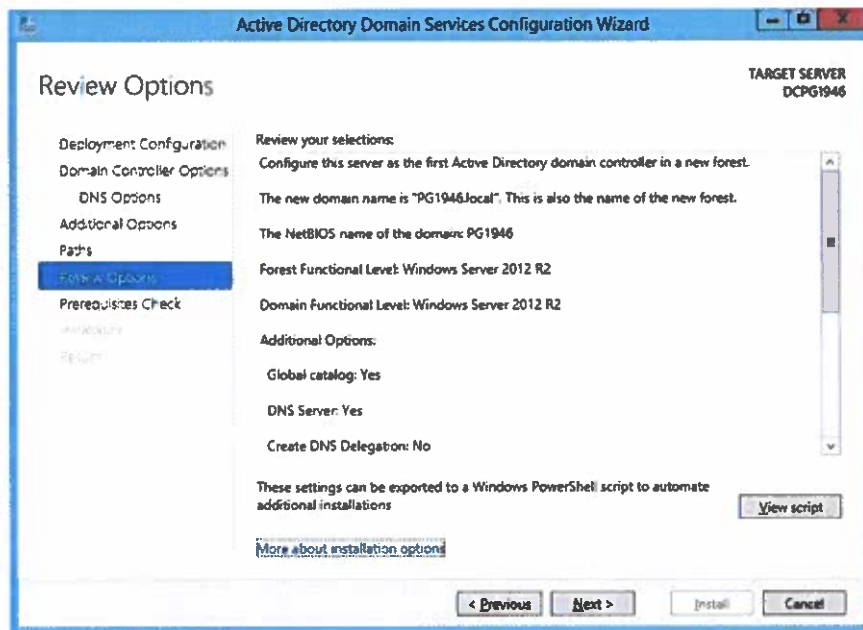


Figura 69: Assistente de configuração do “Active Directory and Domain Services” 3, Fonte Própria

Voltando à janela “Server Manager” na secção “Local Server” podemos verificar que o nosso domínio já se encontra criado e o servidor DCPG1946 já está no domínio.

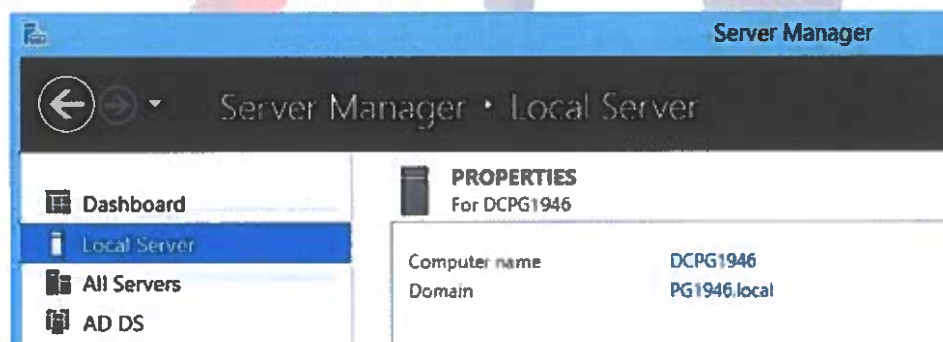


Figura 70: Server Manager 2, Fonte Própria

Após ter sido criado o domínio e promovido o servidor a controlador de domínio, iremos passar a entrar no respetivo servidor com a conta de Administrador do domínio:



Figura 71: Login no Controlador de Domínio, Fonte Própria

4.2.3.2.4 Adicionar a Role DHCP Server

Apesar de terem sido atribuídos IPs fixos a cada um dos servidores do laboratório, iremos mais à frente através do SCVMM, instalado no servidor VMMPG1946, criar máquinas clientes e adiciona-las a este domínio, através de um *template*. Uma vez que as referidas máquinas clientes serão criadas através de um *template*, as suas interfaces de rede estarão configuradas para receber o seu endereçamento IP por DHCP. Iremos adicionar a role “DHCP Server” ao controlador de domínio

Vamos dar início à instalação da role DHCP Server através do assistente de configuração “Add Roles and Features”, para tal vamos ao “Server Manager”, no menu seleccionamos “Manage” > “Add Roles and Features”:

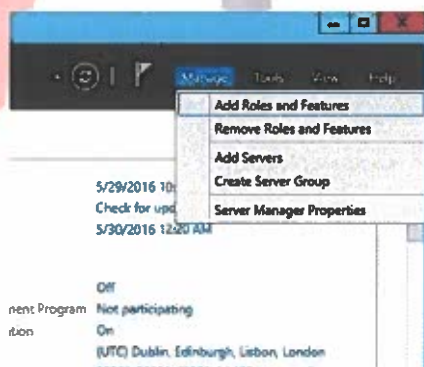


Figura 72: Server Manager 2, “Add Roles and Features”, Fonte Própria

No assistente de configuração “Add Roles and Features”, na secção “Server Roles” iremos seleccionar a role que pretendemos instalar, neste caso “DHCP Server”:

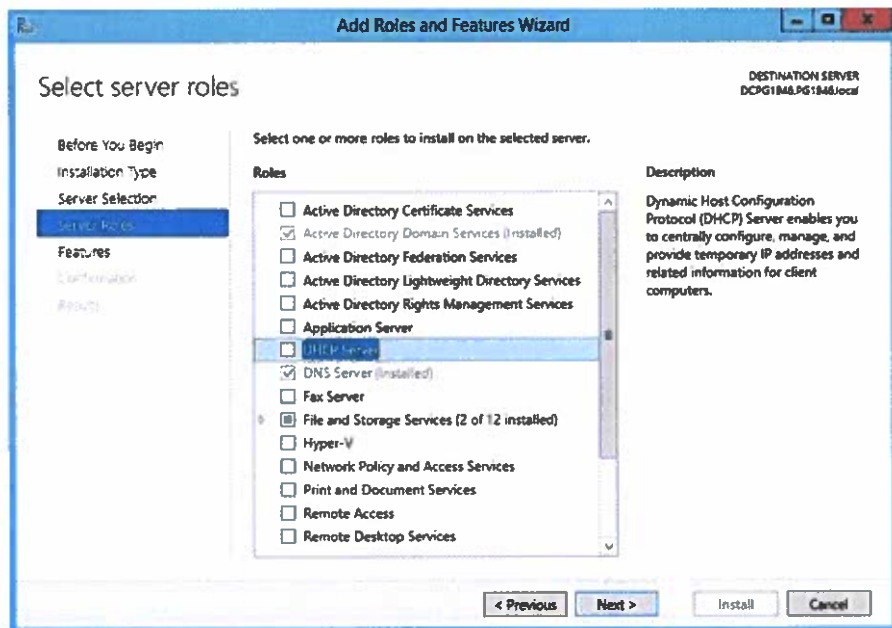


Figura 73: Assistente de configuração “Add Roles and Features”, DHCP Server 1, Fonte Própria

No fim teremos um quadro com a informação das “Roles and Features” que irão ser instaladas:

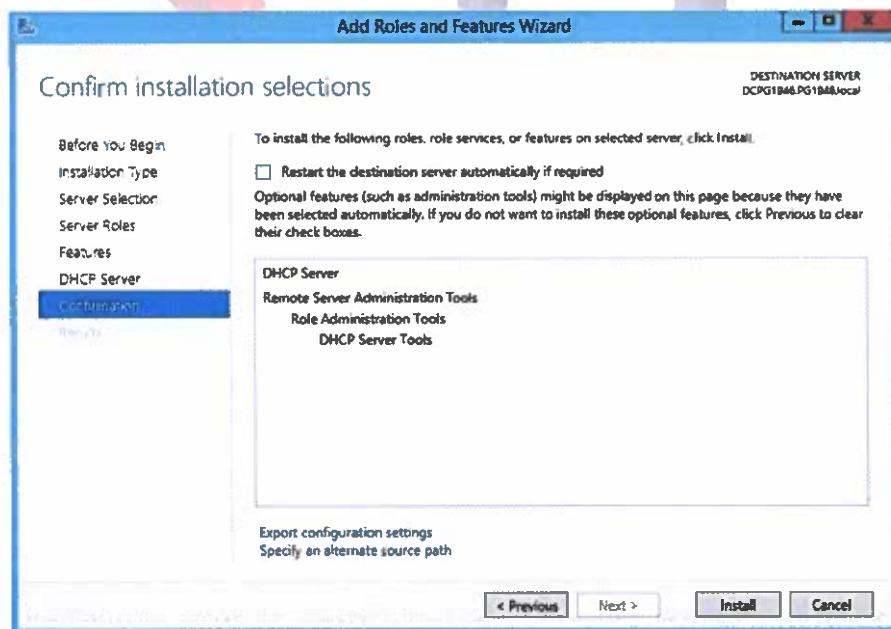


Figura 74: Assistente de configuração “Add Roles and Features”, DHCP Server 2, Fonte Própria

4.2.3.2.5 Criar Range de IPs para atribuição pelo DHCP Server

Vamos definir uma range de IPs reservados para atribuição às novas máquinas que sejam adicionadas ao domínio sem IP fixo. Neste laboratório todos os servidores foram definidos com o endereçamento IP manualmente, pelo que será estático e sem atribuição automático do “DHCP server”. Esta range de IPs que vamos definir no “DHCP server” será para atribuição do endereçamento IP das máquinas clientes que mais à frente serão adicionadas ao domínio. Essas máquinas clientes terão a sua interface de rede LAN com atribuição do endereçamento IP por DHCP, como tal ser-lhes-á atribuído um IP desta range que vamos definir no “DHCP server”.

Para darmos inicio a esta configuração teremos de ir às configurações do DHCP server, no menu do “Server Manager” vamos seleccionar “Tools” > “DHCP”:

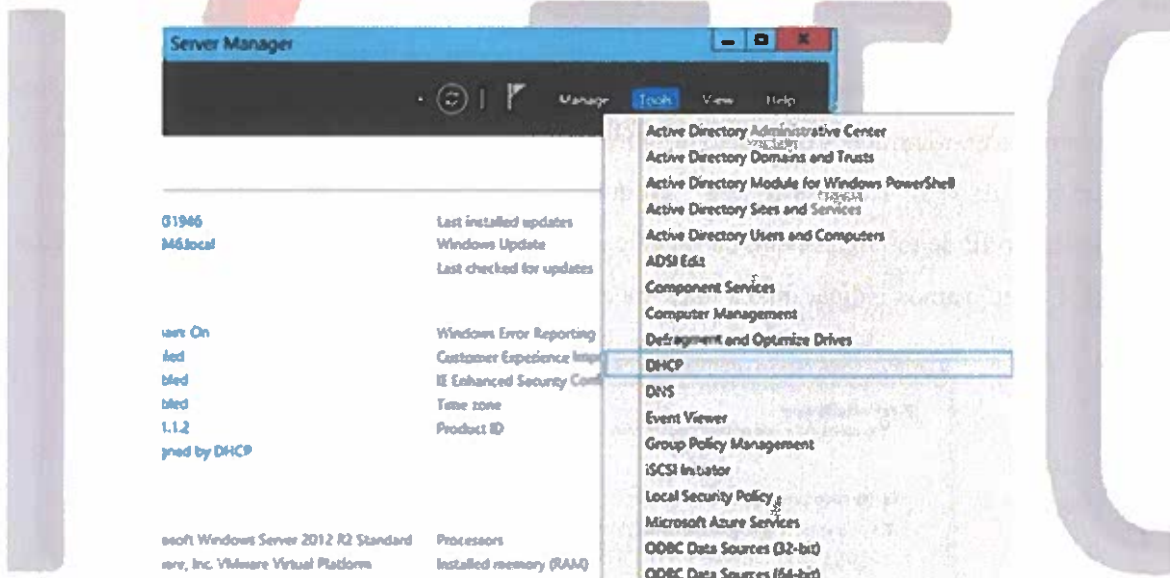


Figura 75: Server Manager, Tools DHCP, Fonte Própria

Na janela do DHCP vamos criar a nova range de IPs, seleccionamos IPv4 > “New Scope...”:

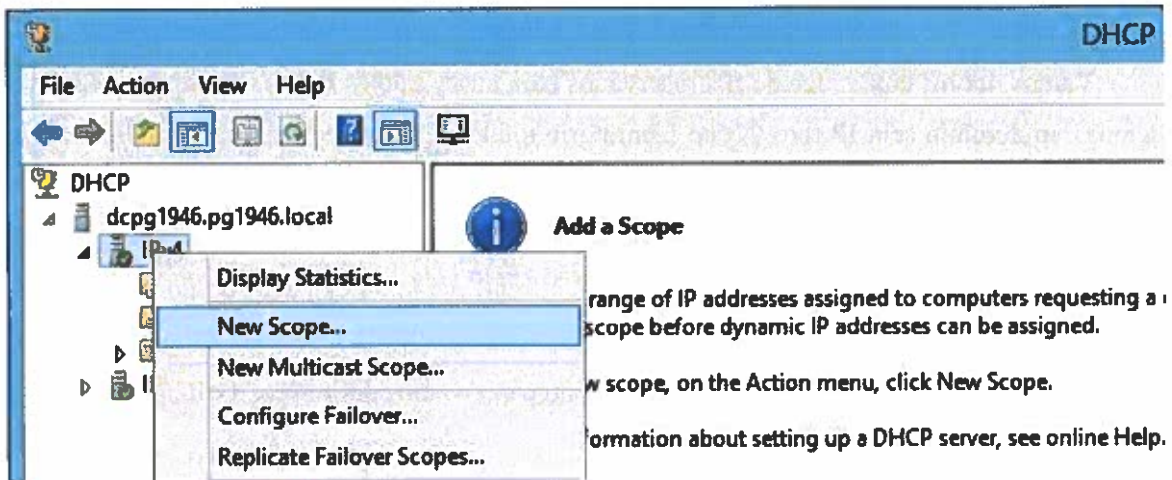


Figura 76: Windows DHCP, Adicionar nova range de IPs, Fonte Própria

É iniciado o assistente de configuração “New Scope”, onde vamos apenas referir os passos que diferem das configurações que estão definidas de modo padrão no assistente de configuração. Como tal no passo “IP Address Range”, no quadro “Configuration settings for DHCP server” vamos definir o IP de início e de fim da range e no quadro “Configuration settings that propagate to DHCP Client” vamos indicar qual a máscara de rede no campo “Subnet mask”:

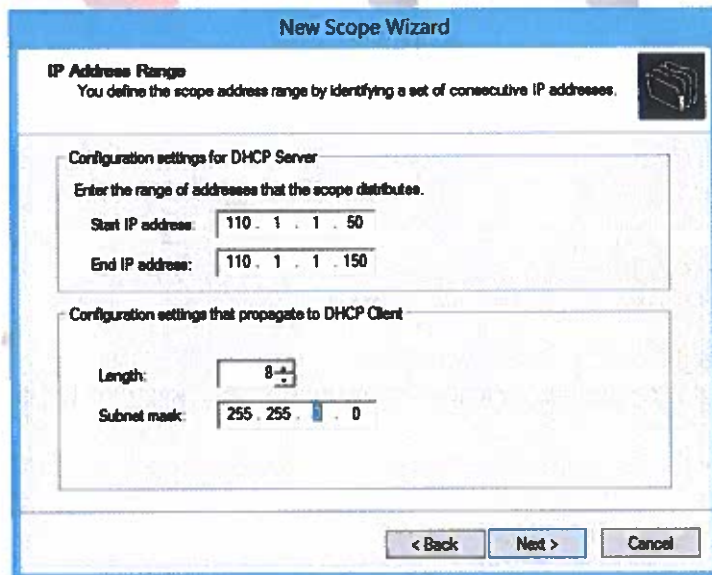


Figura 77: Assistente de configuração “New Scope”, “IP Address Range”, Fonte Própria

Em seguida, no “Lease Duration” vamos colocar o tempo que a máquina cliente terá o mesmo IP, para efeitos deste laboratório definimos um tempo alargado, neste caso 8 dias:

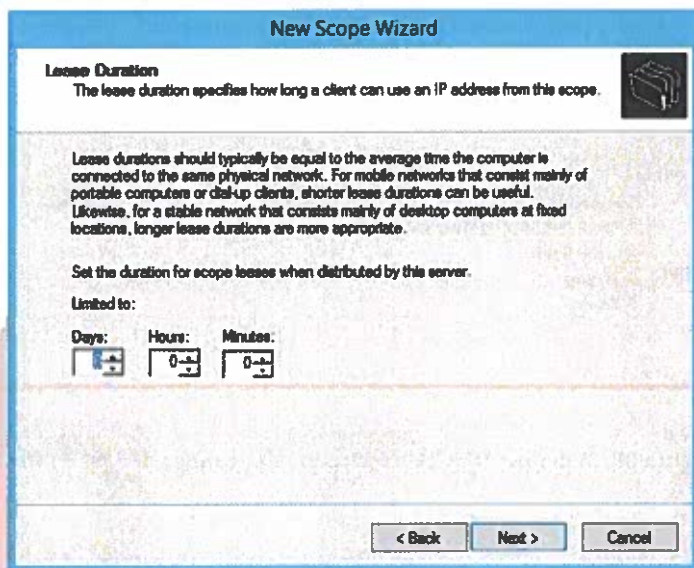


Figura 78: Assistente de configuração “New Scope”, “Lease Duration”, Fonte Própria

No próximo passo, vamos indicar o nome do domínio onde está o servidor DNS e o IP do respetivo servidor que tem o DNS, neste caso será o IP da interface de rede LAN que atribuímos ao servidor DCPG1946:

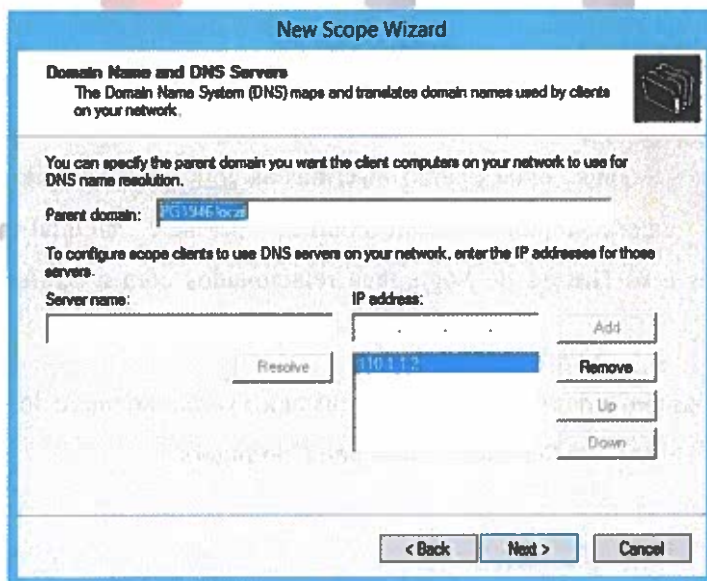


Figura 79: Assistente de configuração “New Scope”, “Domain Name and DNS Servers”, Fonte Própria

Após ser terminada a configuração no assistente de configuração “New Scope”, podemos verificar a nossa range de IPs já definida no DHCP, com o nome PG1946DHCP:

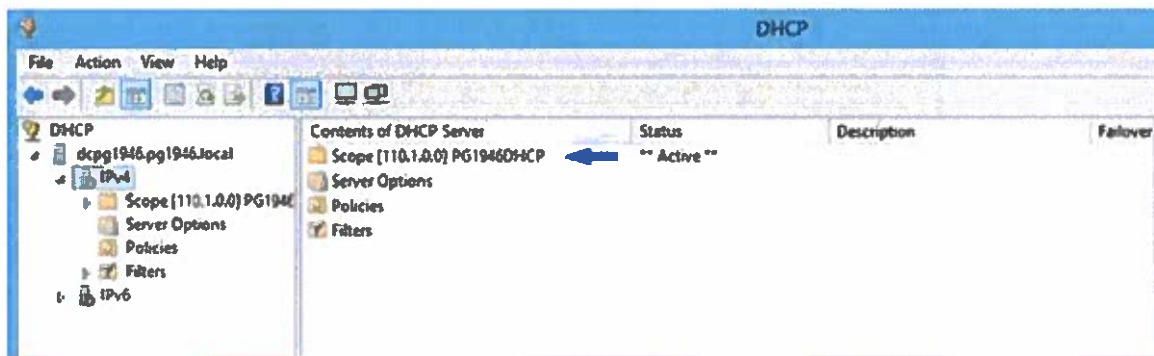


Figura 80: Windows DHCP, Adicionada nova range de IPs, Fonte Própria

4.2.3.2.6 Unidades Organizacionais, Contas e Grupos de Serviço

Por questões de boas práticas de segurança, é necessário criarmos Unidades Organizacionais. Deste modo será criada uma Unidade Organizacional principal, à qual iremos chamar BiApp, referente a todo este laboratório da Biblioteca Virtual. Dentro desta serão criadas outras Unidades Organizacionais, para os Servidores, à qual daremos o nome de “Servers”, onde ficarão inseridos todos os servidores do laboratório. Uma UO com o nome “Workstations” onde serão colocadas todas as máquinas clientes, criadas a partir do SCVMM. Outra UO à qual será dado o nome “SCVMM Accounts”, na qual iremos criar as contas de utilizadores e o Grupo de Segurança necessários à configuração do SCVMM. Será também criada uma UO com o nome “SQL Server Service Accounts” onde ficarão inseridas as contas de utilizadores para gestão do SQL Server. Por fim será criada uma UO, com o nome “AppUsers”, na qual estarão inseridas as contas de utilizadores e os Grupos de Segurança relacionados com a Aplicação da Biblioteca Virtual.

Para criarmos as Unidades Organizacionais primeiro vamos ao menu do “Server Manager”, selecionamos “Tools” > “Active Directory Users and Computers”:

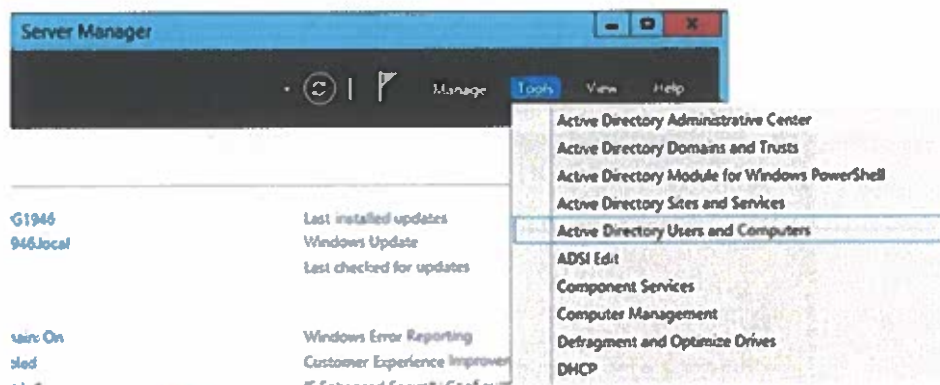


Figura 81: Server Manager, “Active Directory Users and Computers”, Fonte Própria

Vamos criar a Unidade Organizacional principal, dentro da qual iremos criar as restantes. Os passos de criação da Unidade Organizacional principal serão repetidos para a criação das restantes Unidades Organizacionais. A única diferença será no primeiro passo, em que para as restantes UO, vamos selecionar a UO principal BiApp, em vez do domínio.

Para iniciarmos a criação da UO, selecionamos o domínio e escolhemos a opção “New” > “Organizational Unit”:

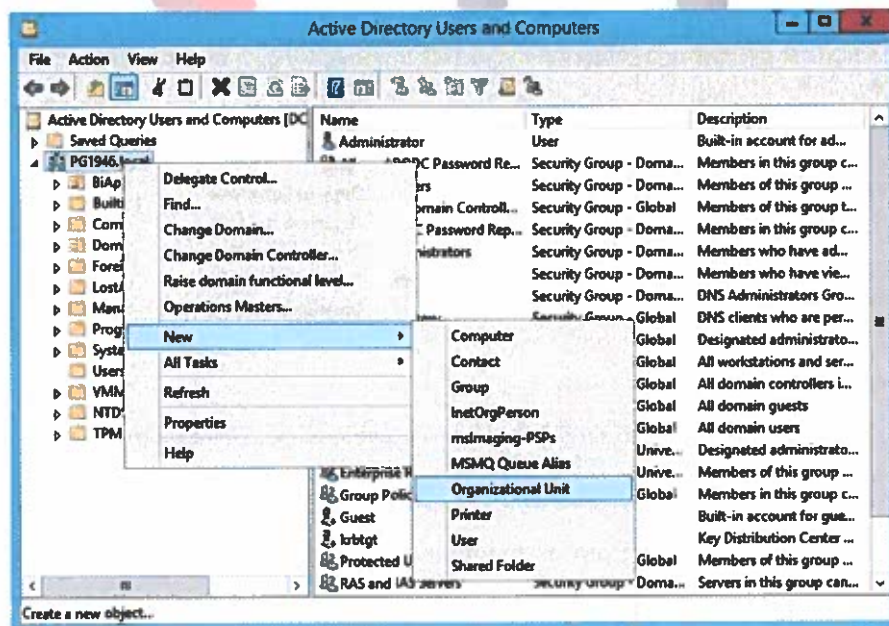


Figura 82: Criar nova Unidade Organizacional 1, Fonte Própria

A esta Unidade Organizacional principal, iremos dar o nome da Aplicação da Biblioteca Virtual, BiApp:

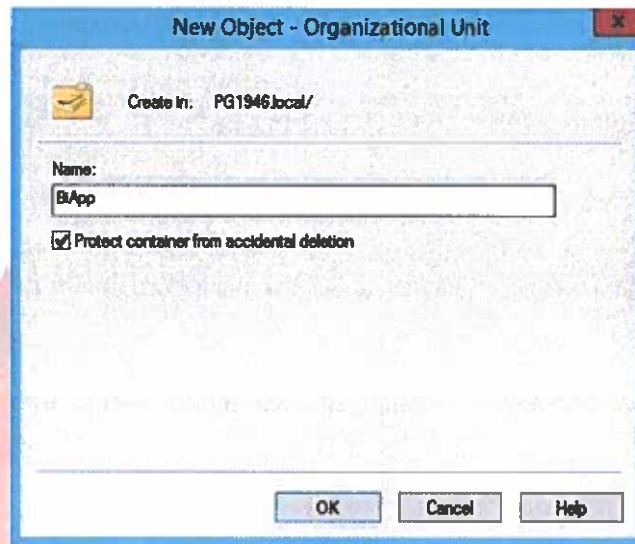


Figura 83: Criar nova Unidade Organizacional 2, Fonte Própria

Aqui podemos ver todas as UO criadas para este laboratório.

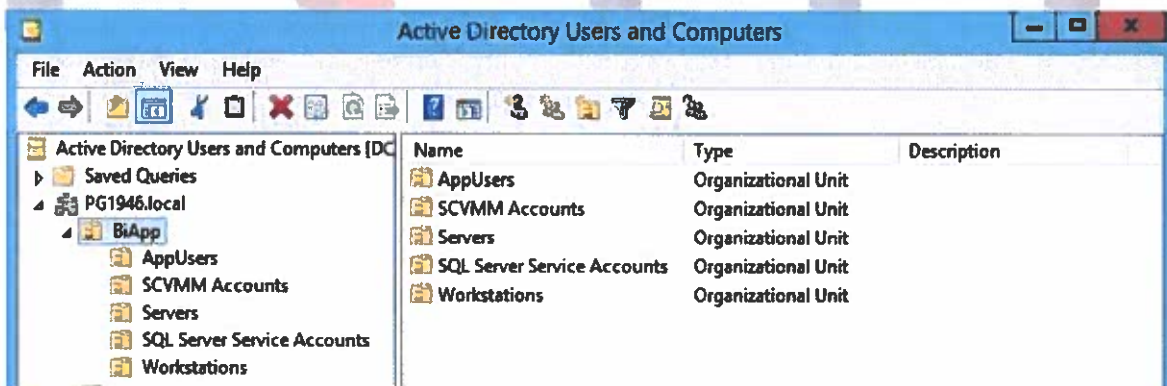


Figura 84: Unidades Organizacionais do Laboratório, Fonte Própria

O contentor padrão onde ficam as máquinas recentemente adicionadas ao domínio é o contentor com o nome "Computers".

Por forma, a que as novas máquinas cliente criadas através do SCVMM sejam colocadas na UO "Workstations", iremos alterar o destino padrão de novas máquinas adicionadas ao domínio,

através do comando “redircmp” na linha de comandos do *PowerShell*. (Microsoft Corporation, 2012)

Assim todas as máquinas cliente que iremos criar no SCVMM e adicionar ao domínio, ficam automaticamente na UO "Workstations" onde teremos configuradas as *Group Policy Objects* para estas máquinas cliente da biblioteca.

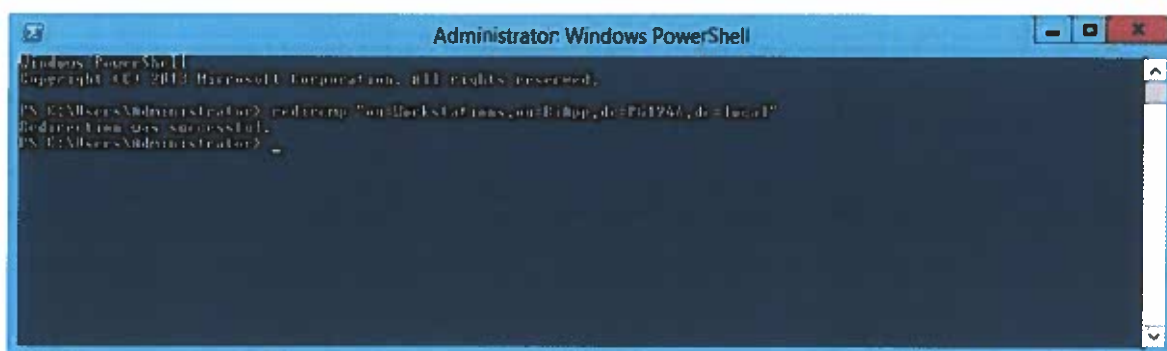


Figura 85: Alterar conteúdo padrão de novas máquinas no domínio, Fonte Própria

Vamos iniciar agora a criação dos Grupos de Segurança. Seleccionamos a Unidade Organizacional “SCVMM Accounts” e escolhemos a opção “New” > “Group”. Os passos serão repetidos para a criação dos restantes Grupos de Segurança necessários, dentro da respetiva Unidade Organizacional.

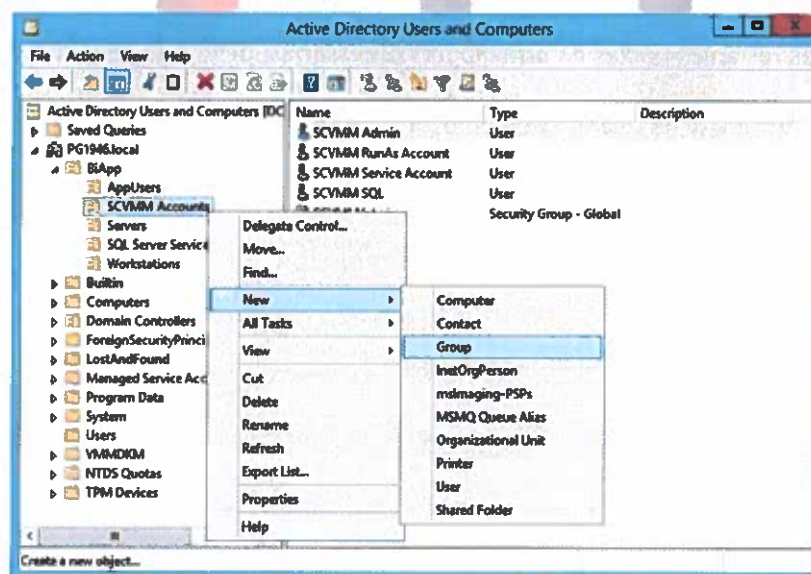


Figura 86: Criar novo Grupo de Segurança 1, Fonte Própria

Para além do Grupo de Segurança, “SCVMMAdmins” que iremos criar dentro da UO “SCVMM Accounts”, será também necessário criar um Grupo de Segurança “AppAdmin” e “AppUser” dentro da UO “AppUsers”.

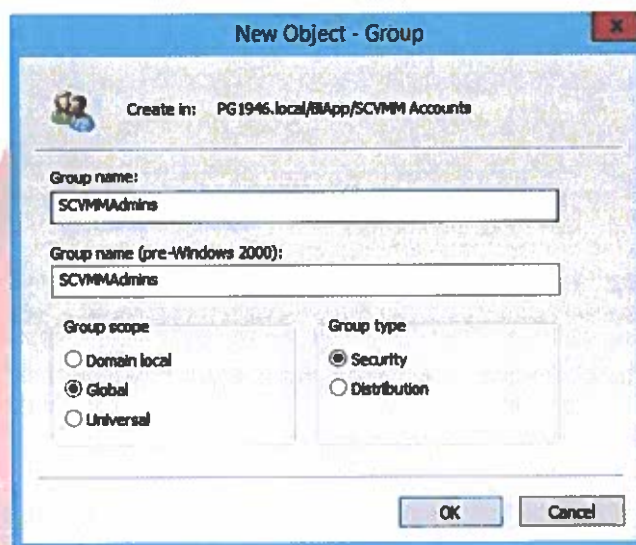


Figura 87: Criar novo Grupo de Segurança 2, Fonte Própria

Vamos criar também as respetivas contas de utilizadores necessárias para o laboratório. Vamos demonstrar a criação da conta “SCVMM Admin” e iremos repetir estes passos para a criação de todas as restantes contas de utilizadores necessárias, dentro da respetiva UO.

Selecionamos a UO na qual pretendemos inserir as contas de utilizador, neste caso iremos selecionar a UO “SCVMM Accounts” e escolhemos a opção “New” > “User”.

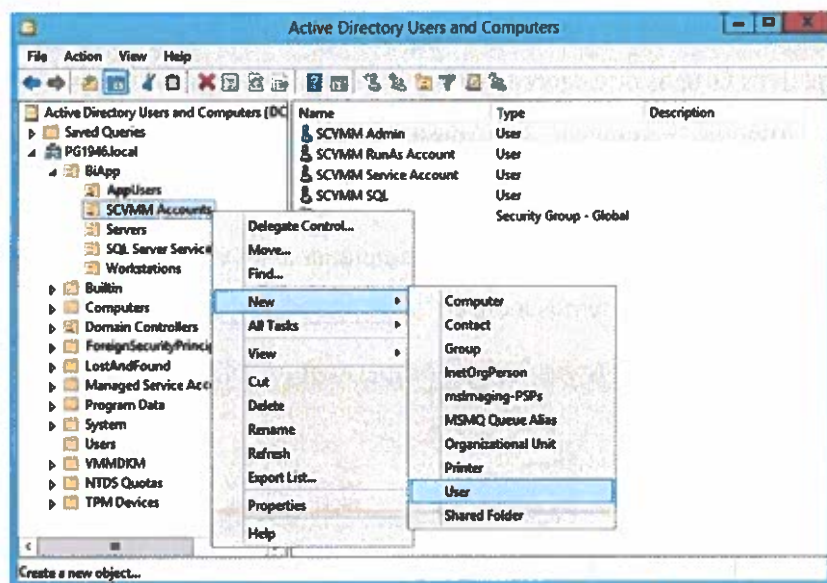


Figura 88: Criar nova conta de utilizador 1, Fonte Própria

Na janela de criação da conta de utilizador, colocamos o primeiro nome “SCVMM”, inserimos o último nome “Admin”, por fim no campo “User login name” iremos colocar o nome pelo qual será feito o login, “admin_scvmm”. É necessário criarmos também as contas “runas_scvmm”, “svc_scvmm” e “sql_scvmm” dentro da mesma OU “SCVMM Accounts”. Na OU “SQL Server Service Accounts” iremos criar a conta “svc_sql” e por fim na OU “AppUsers” serão criadas as contas “Supervisor1” e “User1”.

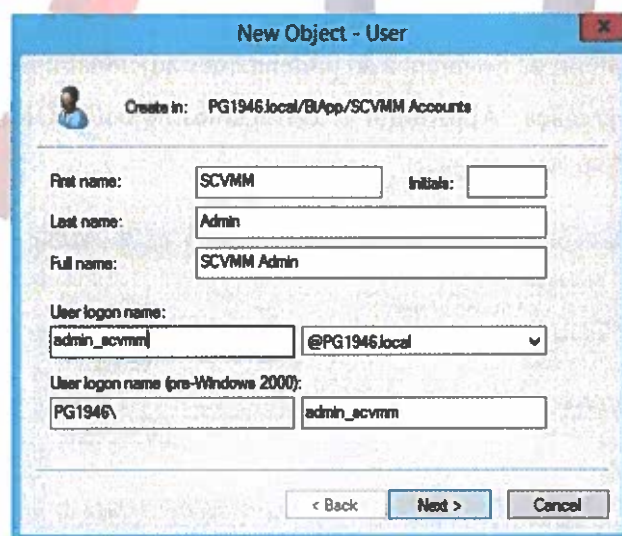


Figura 89: Criar nova conta de utilizador 2, Fonte Própria

Após a criação de todos os Grupos de Segurança e contas de utilizador, vamos adicionar as contas aos respetivos Grupos de Segurança. Os passos aqui demonstrados para adicionar a conta “supervisor1” ao Grupo de Segurança “AppAdmin” serão repetidos para as restantes contas que terão de ser adicionadas aos Grupos de Segurança já criados.

Vamos selecionar a conta que pretendemos adicionar a um Grupo de Segurança, neste caso será a conta “supervisor1” e escolhemos a opção “Add to a group”.

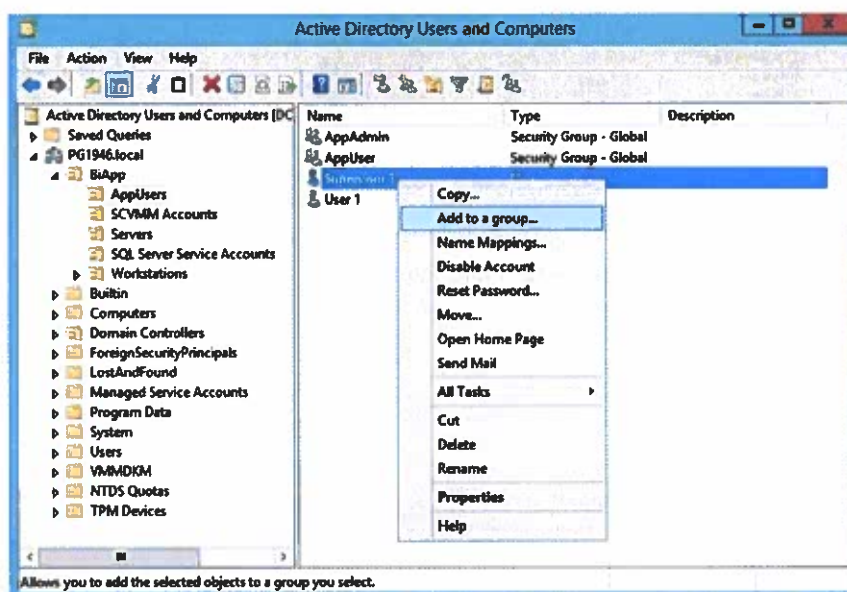


Figura 90: Adicionar conta ao Grupo de Segurança 1, Fonte Própria

Selecionamos o Grupo de Segurança ao pretendemos adicionar a respetiva conta, ara este caso será o Grupo de Segurança “AppAdmin” e carregamos no botão OK. Desta forma a conta ficará associada ao Grupo de Segurança.

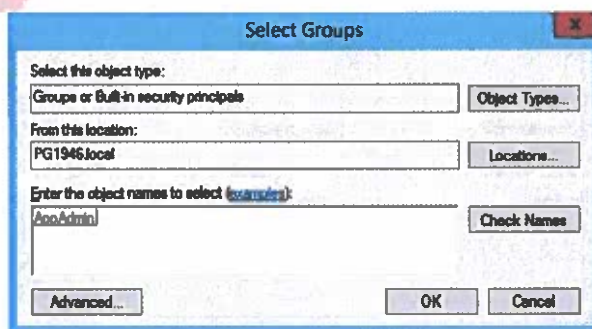


Figura 91: Adicionar conta ao Grupo de Segurança 2, Fonte Própria

Aqui podemos verificar todas as contas inseridas no Grupo de Segurança “SCVMMAdmins”

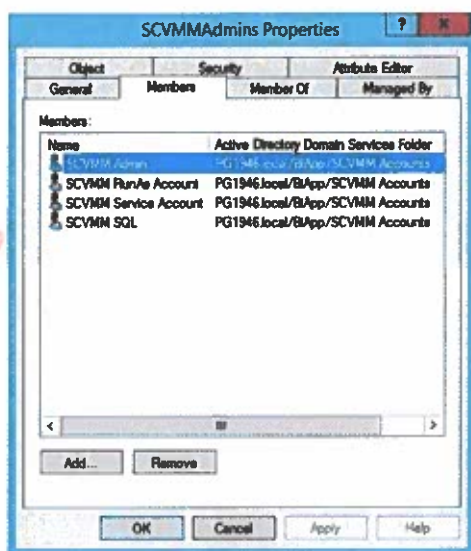


Figura 92: Contas do Grupo de Segurança “SCVMMAdmins”, Fonte Própria

No Grupo de Segurança “AppAdmin” foram adicionados as seguintes contas:



Figura 93: Contas do Grupo de Segurança “AppAdmin”, Fonte Própria

Por último, ao Grupo de Segurança “AppUser” foram adicionadas as contas ilustradas na figura seguinte:

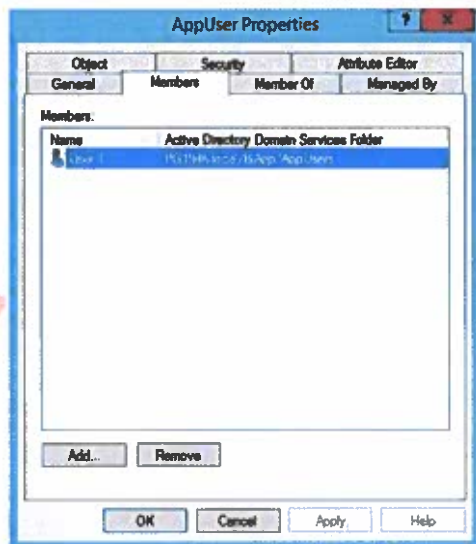


Figura 94: Contas do Grupo de Segurança “AppUser”, Fonte Própria

Iremos mostrar uma visualização geral de todas as Unidades Organizacionais e os seus conteúdos, após as configurações efetuadas nos passos anteriores.

Na Unidade Organizacional “AppUsers” temos o seguinte conteúdo:



Figura 95: Conteúdo Unidade Organizacional “AppUsers”, Fonte Própria

A Unidade Organizacional “SCVMM Accounts” ficou com o seguinte conteúdo:



Figura 96: Conteúdo Unidade Organizacional “SCVMM Accounts”, Fonte Própria

Na Unidade Organizacional “Servers” ficámos com o seguinte conteúdo:

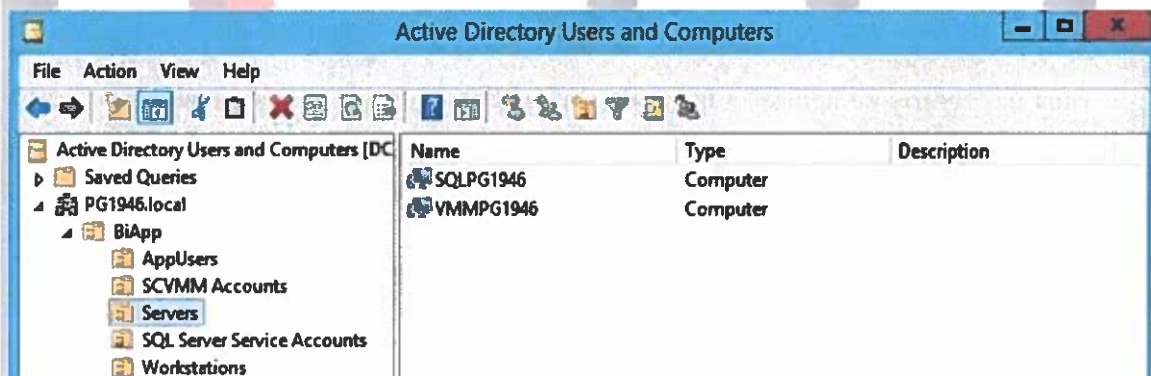


Figura 97: Conteúdo Unidade Organizacional “Servers”, Fonte Própria

Na Unidade Organizacional “SQL Server Service Accounts” temos o seguinte conteúdo:

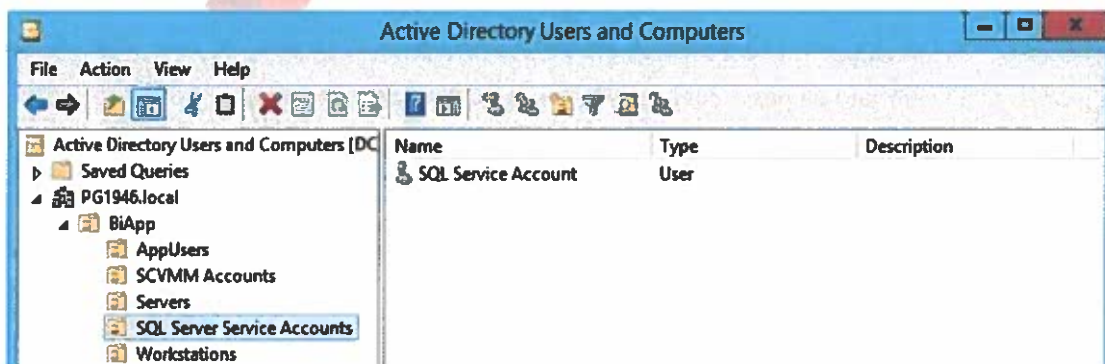


Figura 98: Conteúdo Unidade Organizacional “SQL Server Service Accounts”, Fonte Própria

Por fim, na Unidade Organizacional “Workstations” ficámos com o seguinte conteúdo:

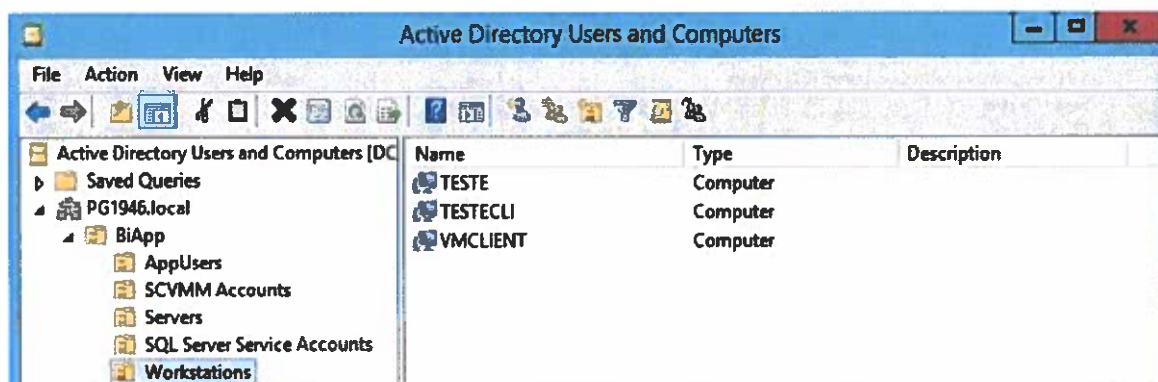


Figura 99: Conteúdo Unidade Organizacional “Workstations”, Fonte Própria

4.2.3.2.7 Group Policy Objects

Uma das formas de agilizar a tarefa de configuração dos servidores e máquinas cliente numa rede, é através das *Group Policy Objects*. As *Group Policy Objects* permitem mudar configurações, restringir ações ou mesmo a instalação de aplicações em toda a rede.

Neste laboratório iremos criar algumas *Group Policy Objects* que nos irão permitir que todas as máquinas no domínio tenham a Firewall ativa, que em todas as máquinas cliente que sejam adicionadas ao domínio sejam instaladas certas aplicações, que determinados grupos de utilizadores tenham um fundo do *desktop* específico.

Todos os ficheiros que iremos utilizar na configuração das *Group Policy Objects*, foram adicionados previamente a uma pasta no Controlador de Domínio, que está partilhada com todo o domínio, de modo a que o seu conteúdo possa ser despoletado para configuração em todas as máquinas do domínio, pois é o Controlador de Domínio que irá despoletar as respetivas *Group Policy Objects*. O caminho da pasta deverá ser o caminho de rede e não o caminho local, que será “\\DCPG1946\GPO Deployment\”

Para iniciarmos a criação de uma *Group Policy Object*, teremos de o fazer no “Group Policy Management”, para tal vamos ao menu do “Server Manager” e seleccionamos “Tools” > “Group Policy Management”

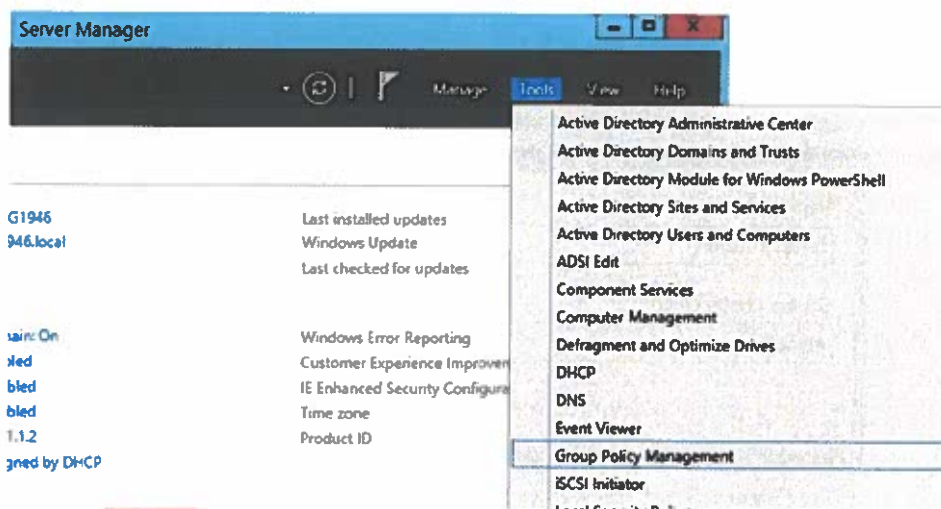


Figura 100: Server Manager, "Group Policy Management", Fonte Própria

Instalação de Aplicações

Vamos proceder à criação de uma *Group Policy Object* que irá fazer a instalação da Aplicação da Biblioteca Virtual em todas as máquinas cliente adicionadas ao domínio. Para as restantes *Group Policy Objects* de instalação de aplicações iremos repetir estes passos. Esta *Group Policy Object* será associada à Unidade Organizacional "Workstations".

Já no "Group Policy Management" selecionamos a Unidade Organizacional à qual pretendemos que a *Group Policy Objects* que vamos criar fique associada, escolhemos a opção "New".

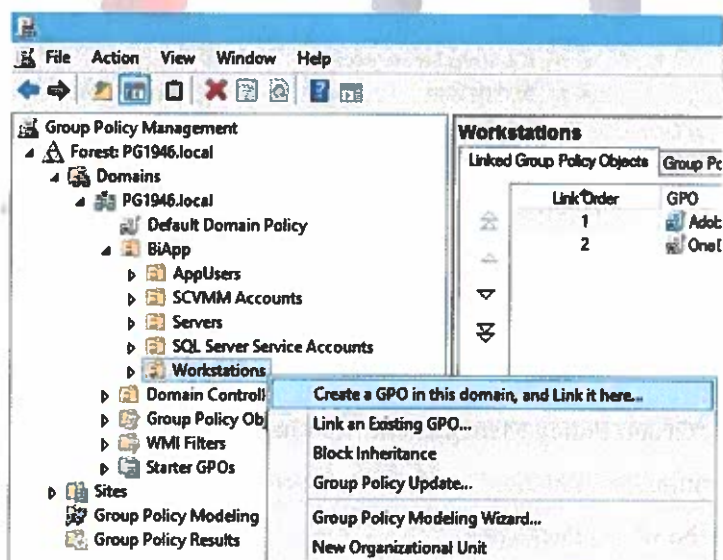


Figura 101: Criar Group Policy Object 1, Fonte Própria

Damos o nome à Group Policy que estamos a criar, neste caso será “BiApp Installer”.

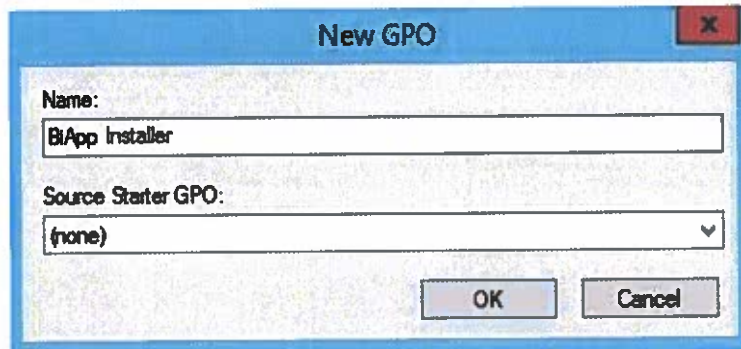


Figura 102: Criar Group Policy Object 2, Fonte Própria

Para configurar a Group Policy Object que acabámos de criar, seleccionamos a mesma e escolhemos a opção “Edit”.

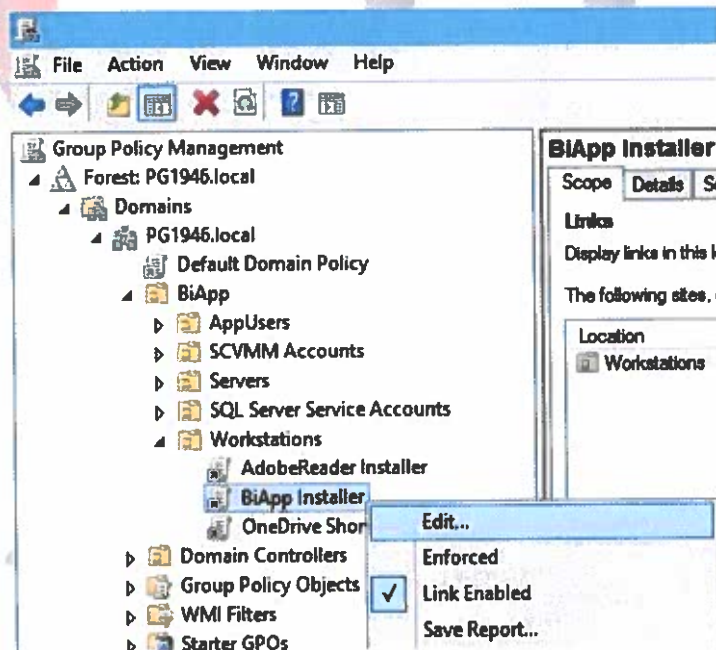


Figura 103: Editar Group Policy Object, Fonte Própria

No editor do “Group Policy Management” no quadro do lado esquerdo, vamos seleccionar “Computer Configuration” > “Policies” > “Software Settings” > “Software Installation”, depois escolhemos a opção “New” > “Package”.

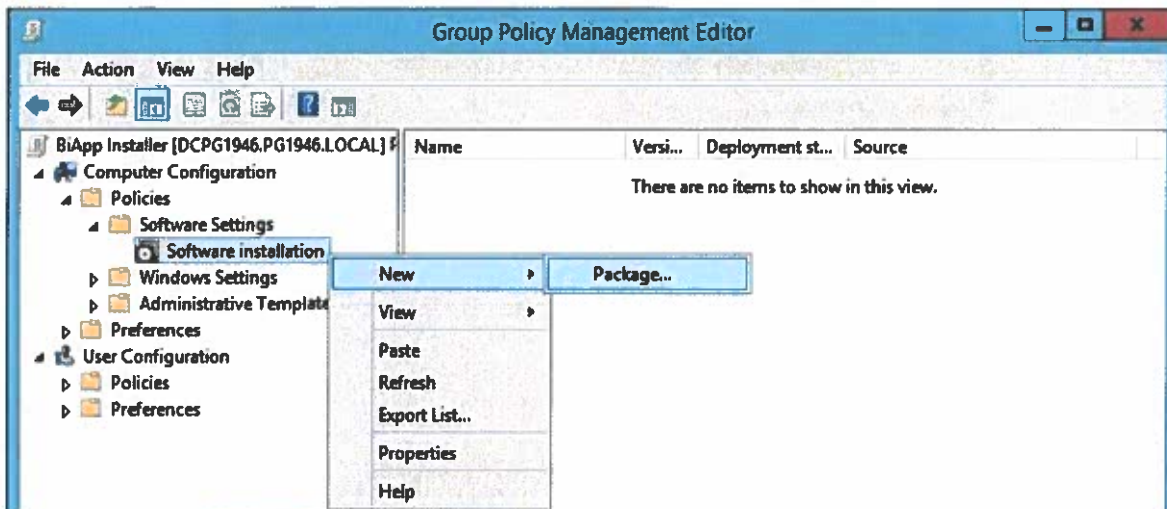


Figura 104: Configurar Group Policy Object 1, Instalação de Aplicações, Fonte Própria

Na janela seguinte escolhemos a opção “Assigned” e carregamos no botão OK.

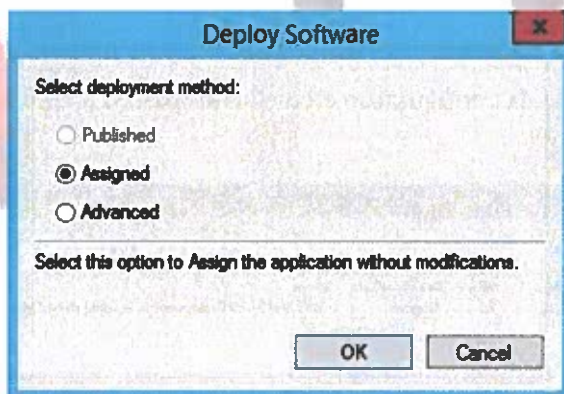


Figura 105: Configurar Group Policy Object 2, Instalação de Aplicações, Fonte Própria

Em seguida selecionamos o ficheiro com a extensão (.msi) que pretendemos que seja instalado.

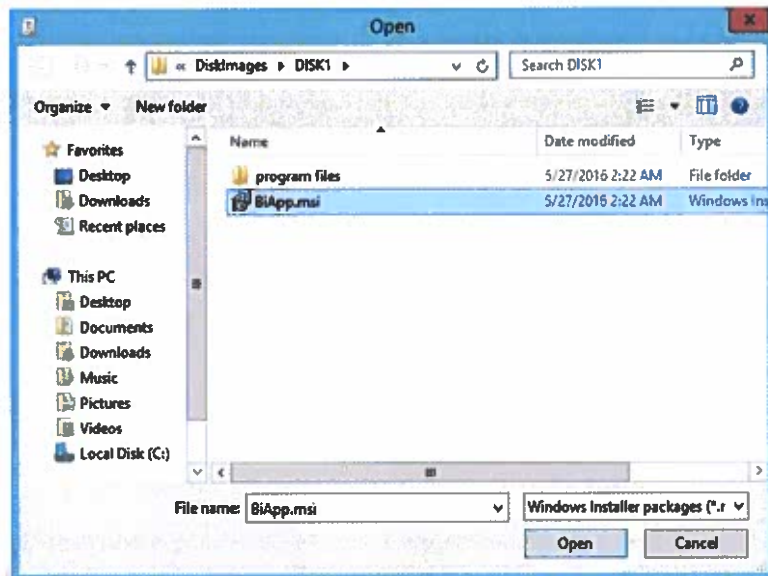


Figura 106: Configurar *Group Policy Object* 3, Instalação de Aplicações, Fonte Própria

No final, podemos confirmar na janela do “Group Policy Management”, no quadro do lado direito já temos a informação da configuração efetuada nos passos anteriores, incluindo o caminho para o ficheiro de instalação.

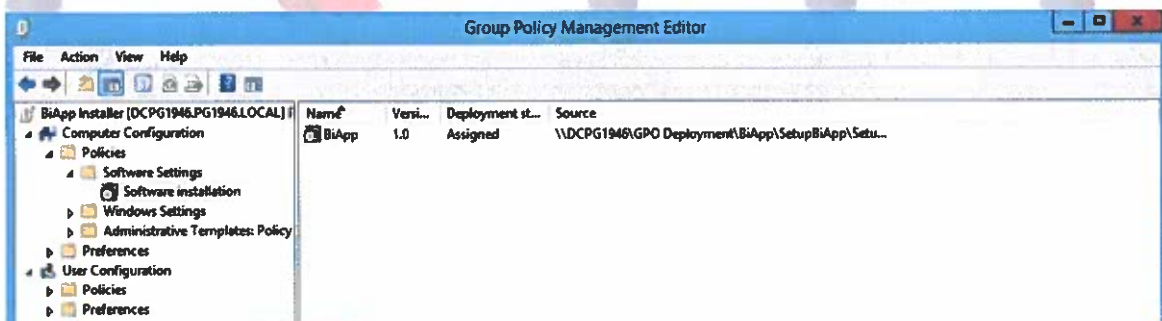


Figura 107: Configurar *Group Policy Object* 4, Instalação de Aplicações, Fonte Própria

Definir Imagem de Fundo do *Desktop*

Criámos outra *Group Policy Object* para ser definido um fundo de *desktop* específico para um grupo de utilizadores. Os passos de criação da *Group Policy Object* são sempre idênticos, pelo que não iremos aqui voltar a repeti-los, saltando logo para o passo do editor do “Group Policy Management”. Esta *Group Policy Object* será associada à Unidade Organizacional “AppUsers”.

Já no editor do “Group Policy Management”, no quadro do lado esquerdo seleccionamos “User Configuration” > “Policies” > “Administrative Templates Policy definitions” > “Desktop”

> “Desktop” e depois no quadro do lado direito vamos selecionar “Desktop Wallpaper” e escolhermos a opção “Edit”.

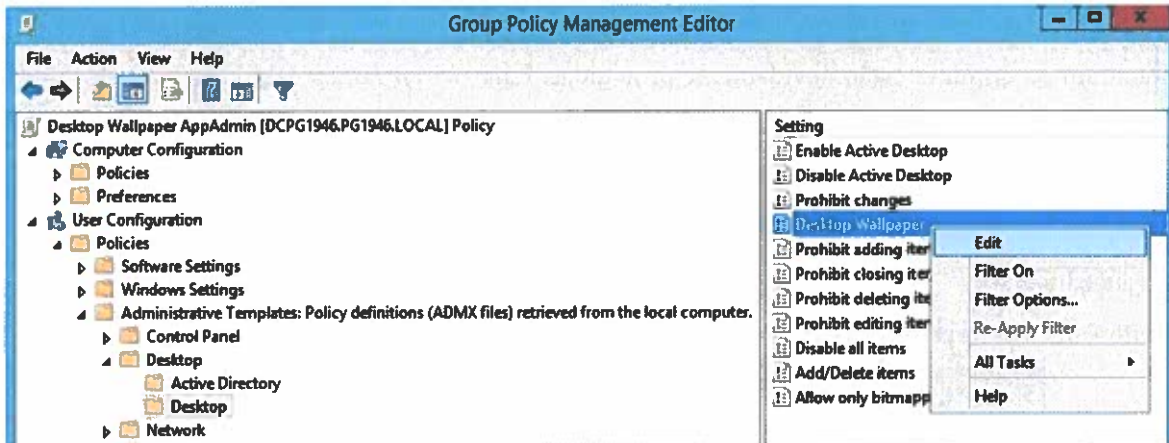


Figura 108: Configurar Group Policy Object 1, Wallpaper, Fonte Própria

Selecionamos a opção “Enable”, no campo “Wallpaper Name” colocamos o caminho para a imagem de fundo do *desktop* que pretendemos. Iremos repetir estes passos para configurar a *Group Policy Object* que irá definir a imagem de fundo do *desktop* do Grupo de Segurança “AppUser”.

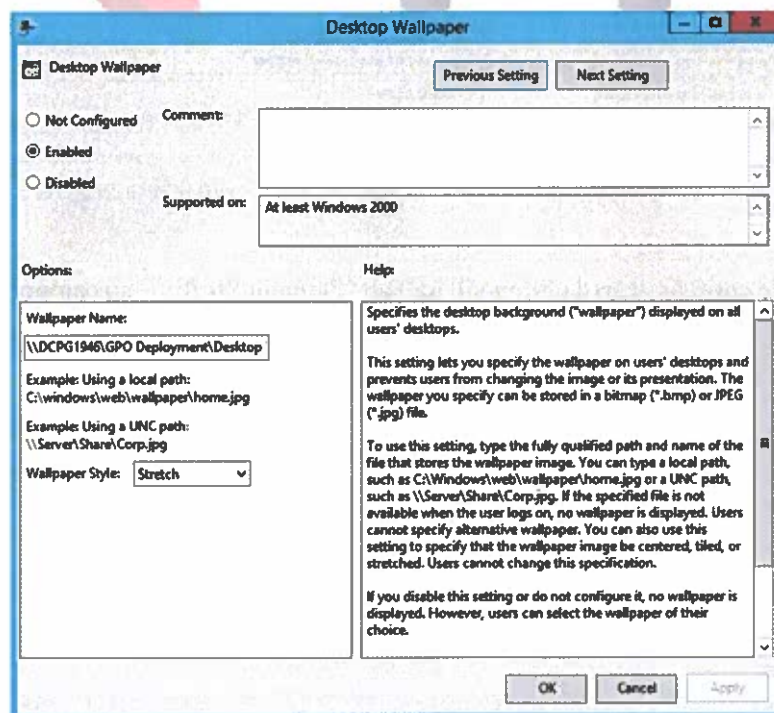


Figura 109: Configurar Group Policy Object 2, Wallpaper, Fonte Própria

Ativar a Firewall

Vamos criar uma outra *Group Policy Object* que irá forçar a ativação da Firewall em todas as máquinas no domínio. Desta forma conseguimos garantir que todas as máquinas que se encontram no domínio estão devidamente protegidas. Esta *Group Policy Object* ficará associada ao domínio, de modo a que todas as máquinas no domínio sejam abrangidas.

Já no editor do “Group Policy Management”, no quadro do lado esquerdo selecionamos “Computer Configuration” > “Policies” > “Windows Settings” > “Windows Firewall with Advanced Security” > “Windows Firewall with Advanced Security” e depois no quadro do lado direito vamos carregar no link “Windows Firewall Settings”.

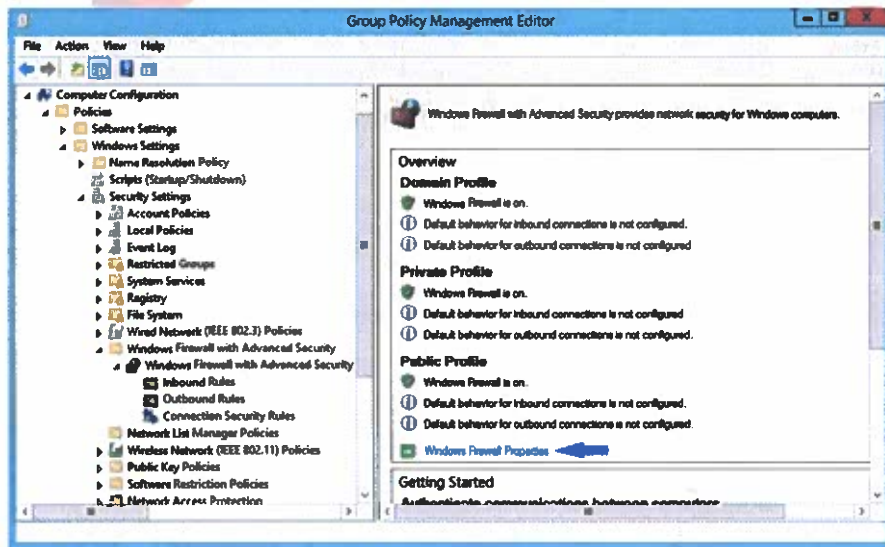


Figura 110: Configurar *Group Policy Object* 1, Firewall, Fonte Própria

Na janela de configuração da Firewall, na Tab “Domain Profile”, no campo “Firewall state” vamos selecionar a opção “On”.

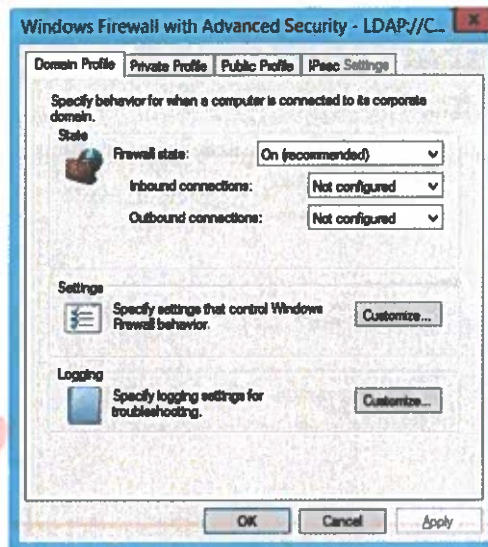


Figura 111: Configurar *Group Policy Object 2, Firewall*, Fonte Própria

Na mesma janela, mas na Tab “Private Profile”, no campo “Firewall state” vamos selecionar a opção “On”.



Figura 112: Configurar *Group Policy Object 2, Firewall*, Fonte Própria

Por último, ainda na mesma janela, na Tab “Public Profile”, no campo “Firewall state” vamos selecionar a opção “On”.

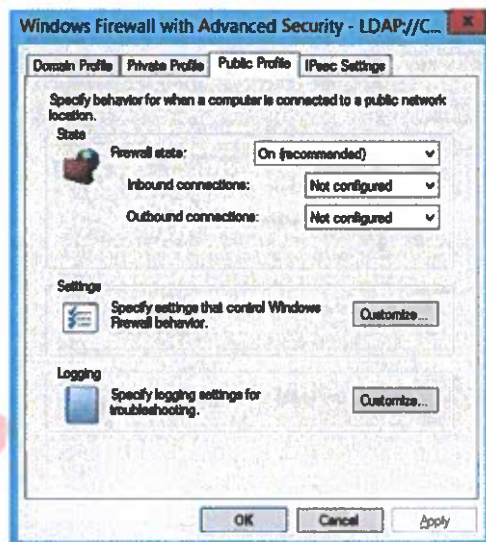


Figura 113: Configurar *Group Policy Object 3, Firewall*, Fonte Própria

Criar um Atalho no Desktop

Vamos configurar uma *Group Policy Object*, que irá criar um atalho para a aplicação OneDrive, que vem previamente instalada no Windows 8.1, que será instalado nas máquinas cliente.

Para tal, já no editor do “Group Policy Management”, no quadro do lado esquerdo selecionamos “Computer Configuration” > “Preferences” > “Windows Settings” > “Files” e escolhemos a opção “New” > “File”.

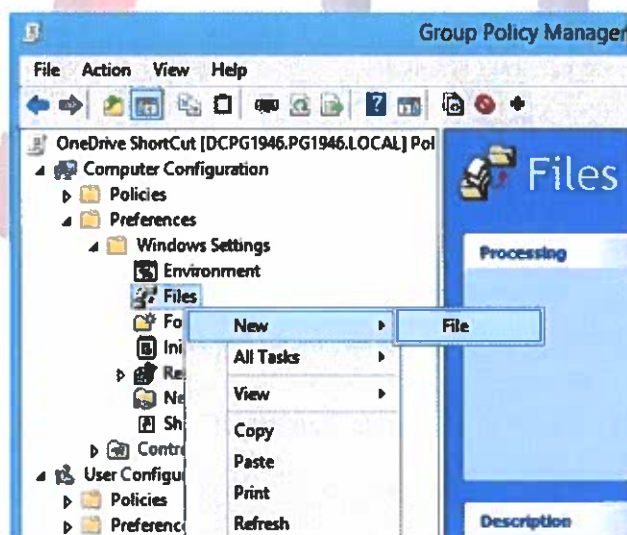


Figura 114: Configurar *Group Policy Object 1, File*, Fonte Própria

Na Tab “General”, no campo “Action” escolhemos a opção Update. No campo “Source file(s)” iremos carregar no botão “...” e vamos selecionar o “Shortcut”, que já contém a localização onde está instalado o OneDrive por defeito no Windows 8.1. Por fim, no campo “Destination File” vamos colocar o local onde pretendemos que seja colocado o “Shortcut”, neste caso vamos colocá-lo no *desktop* de todos os utilizadores da máquina, para tal colocamos o seguinte caminho “%CommonDesktopDir%\OneDrive.lnk”.



Figura 115: Configurar *Group Policy Object 2, File, Fonte Própria*

Após terminar a configuração, podemos verificar no editor do “Group Policy Management”, no quadro do lado direito o resumo da configuração feita.

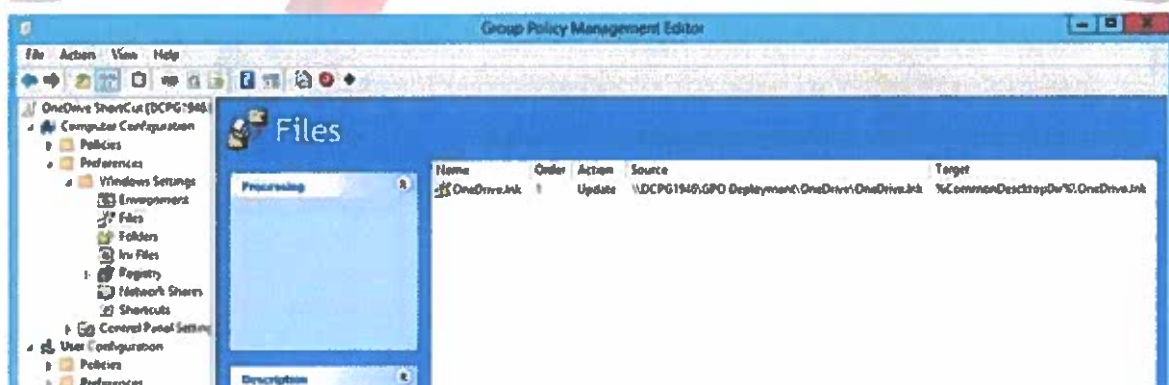


Figura 116: Configurar *Group Policy Object 3, File, Fonte Própria*

Resumo das *Group Policy Objects* Configuradas

Mostramos um resumo de todas *Group Policy Objects* que configurámos no nosso laboratório.

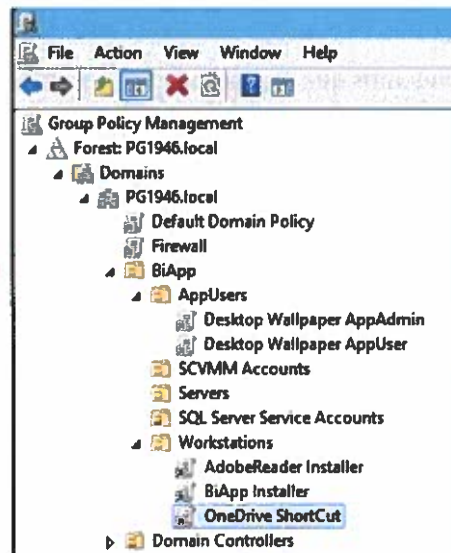


Figura 117: Resumo de todas as *Group Policy Object* configuradas, Fonte Própria

4.2.3.3 Servidor SQL

Criámos este servidor para alojar as bases de dados do SCVMM 2012 R2 e da Aplicação da Biblioteca Virtual. O SQL Server 2012 é um dos motores de base de dados suportados pelo SCVMM 2012 R2, cuja instalação vamos explicar mais à frente neste projeto. Neste servidor, ao qual demos o nome SQLPG1946, iremos instalar o SQL Server 2012, onde nesta secção vamos mostrar os passos necessários para a sua instalação.

4.2.3.3.1 Instalação do SQL Server 2012

Para dar início à instalação do SQL Server 2012 no servidor SQLPG1946, vamos em primeiro lugar adicionar no VMware, na drive CD/DVD do servidor virtual a imagem do “Microsoft SQL Server 2012”.

Para tal no VMware no quadro do lado esquerdo, seleccionamos o servidor virtual SQLPG1946 e em seguida escolhemos a opção “Settings”. Na janela “Virtual Machine Settings” seleccionamos a drive CD/DVD e no quadro do lado direito seleccionamos a opção “Use ISO image

file”, em seguida carregamos no botão “Browse” e escolhemos a imagem do SQL Server 2012 a instalar, por fim selecionamos as opções “Connected” e “Connect at power on”:



Figura 118: VMware configuração de hardware, Adicionar Imagem, Fonte própria

Dentro do servidor virtual vamos à pasta “This PC” e podemos verificar que já lá se encontra a imagem montada do SQL Server 2012:

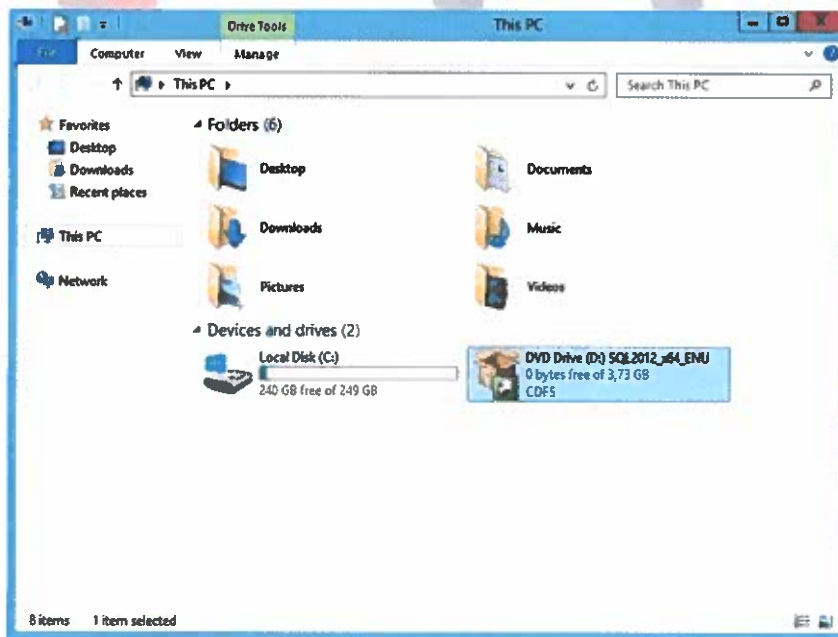


Figura 119: “This PC”, SQLPG1946, Fonte Própria

Após darmos início ao processo de instalação, escolhemos a opção de uma nova instalação: “New SQL Server stand-alone installation or add features to an existing installation”.

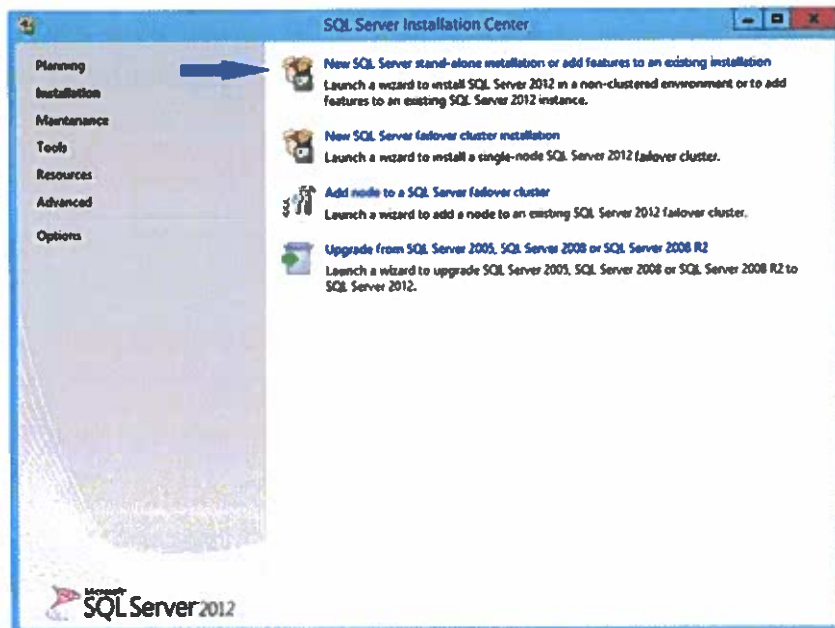


Figura 120: Assistente de Instalação 1, SQL Server 2012, Fonte Própria

Para além das funcionalidades obrigatórias nesta instalação, “Database Engine Services” e “Management Tools - Complete”, iremos também selecionar a opção “Cliente Tools Connectivity”.

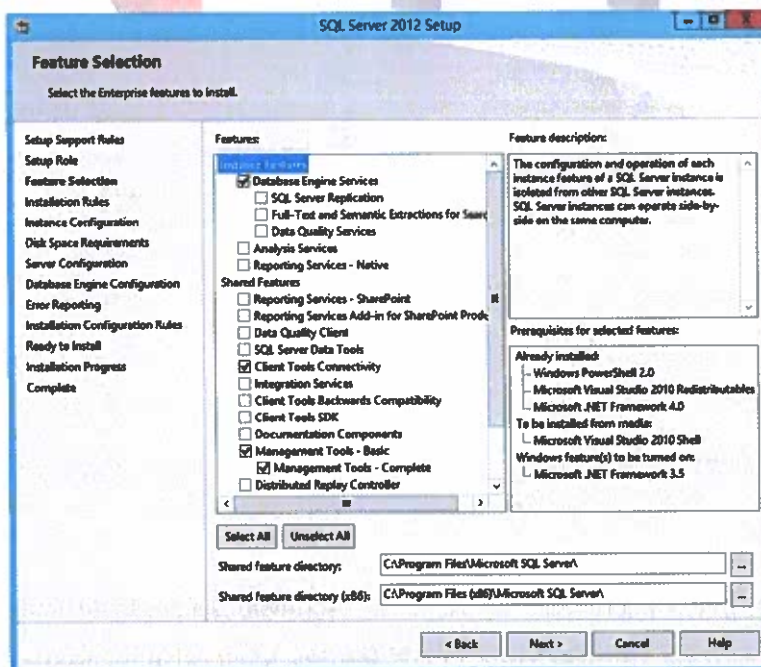


Figura 121: Assistente de Instalação 2, SQL Server 2012, Fonte Própria

No próximo passo, vamos definir o nome da instância do SQL Server, neste caso demos o nome de MSSQLPG1946.

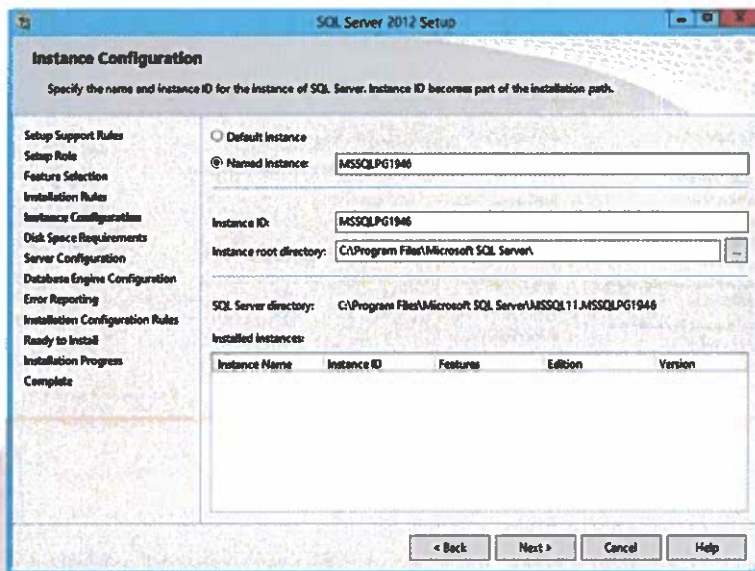


Figura 122: Assistente de Instalação 3, SQL Server 2012, Fonte Própria

No passo seguinte, é necessário associarmos os utilizadores, já criados anteriormente, aos serviços “SQL Server Agent”, “SQL Server Database Engine”, “SQL Server Reporting”, que irão correr neste servidor de instalação. No arranque de todos os serviços deverá estar seleccionada a opção “Automatic”.

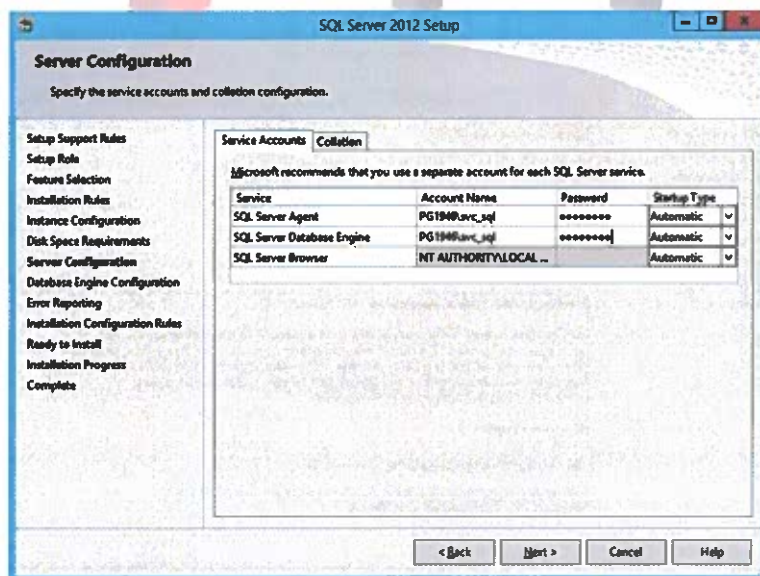


Figura 123: Assistente de Instalação 4, SQL Server 2012, Fonte Própria

De seguida adicionamos os grupos de segurança “SCVMAdmins” e “AppAdmin” como administradores do serviço “SQL Server”.

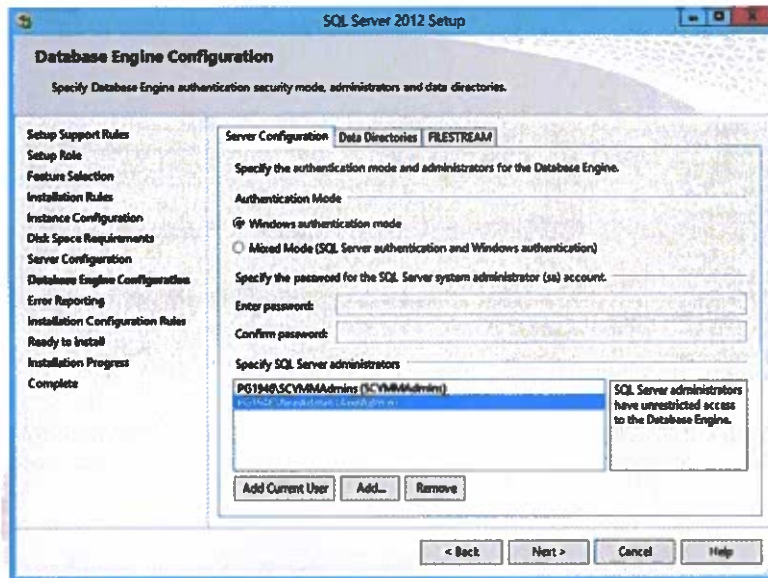


Figura 124: Assistente de Instalação 5, SQL Server 2012, Fonte Própria

No fim da instalação podemos verificar que todas as funcionalidades escolhidas foram corretamente instaladas.

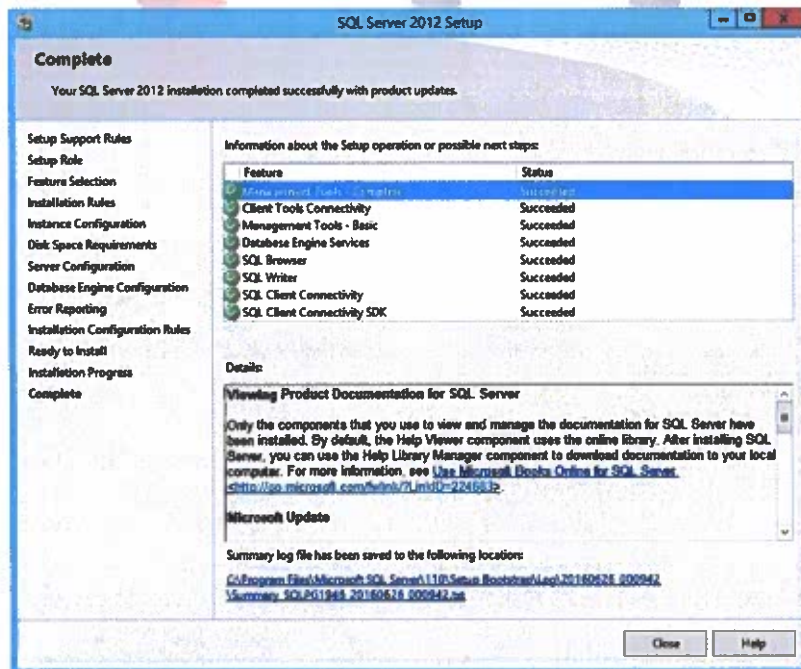


Figura 125: Assistente de Instalação 6, SQL Server 2012, Fonte Própria

Para que o servidor virtual VMMPG1946, que terá o SCVMM 2012 R2 instalado e as máquinas cliente que terão a Aplicação da Biblioteca Virtual instalada, possam comunicar com a instância do SQL Server 2012 onde estão as bases de dados, é necessário abrir certas portas na Firewall do Windows deste servidor.

Para tal, vamos aceder às configurações da Firewall do servidor virtual SQLPG1946, no menu do “Server Manger” vamos seleccionar “Tools” > “Windows Firewall with Advanced Security”.

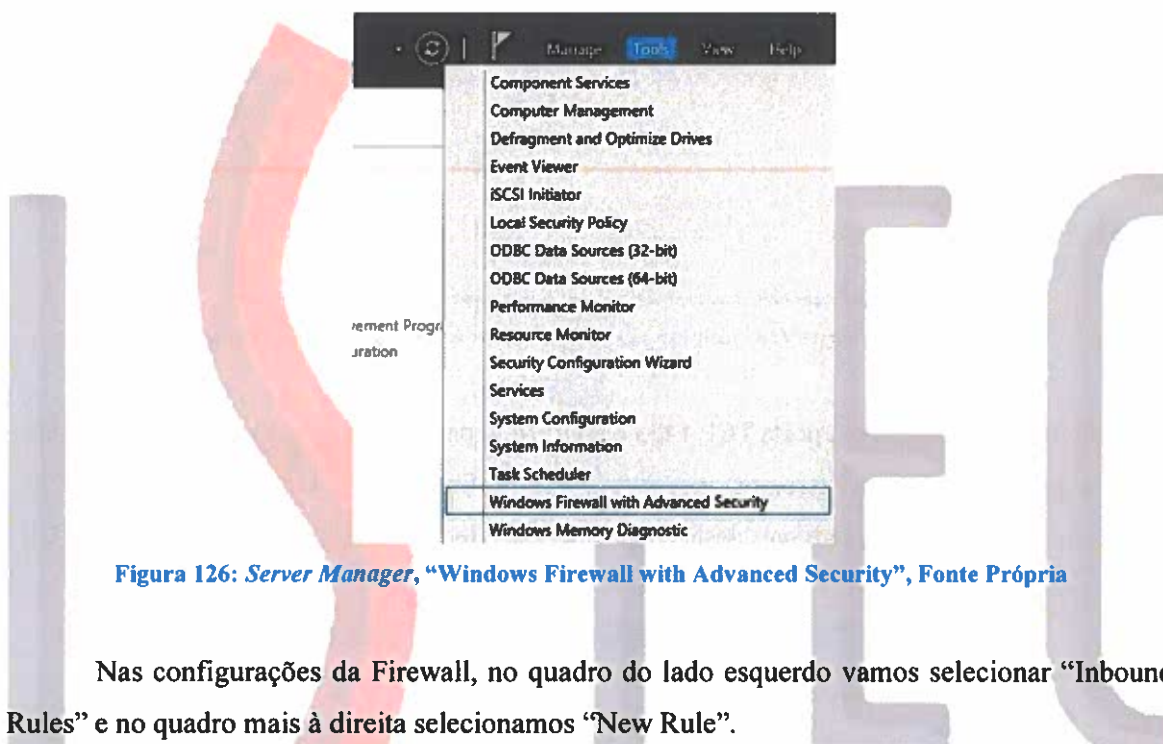


Figura 126: Server Manager, “Windows Firewall with Advanced Security”, Fonte Própria

Nas configurações da Firewall, no quadro do lado esquerdo vamos seleccionar “Inbound Rules” e no quadro mais à direita seleccionamos “New Rule”.



Figura 127: Windows Firewall 2, “Inbound Rules”, Fonte Própria

No Assistente de Configuração de novas regras da *firewall* selecionamos a opção “Port”.

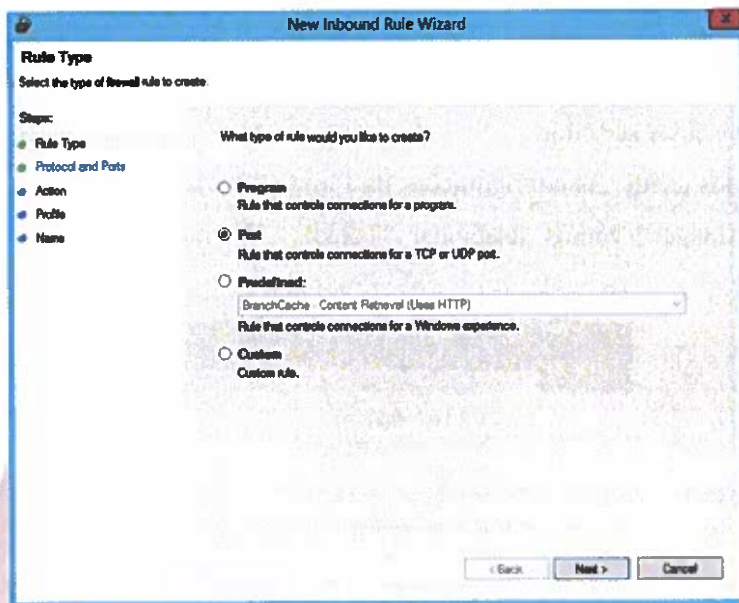


Figura 128: Assistente de Configuração Nova Regra da Firewall 1, Fonte Própria

O SQL Server utiliza a porta TCP 1433 e UDP 1434 para as ligações à instância onde estão as bases de dados, pelo que teremos criar uma regra na firewall do Windows que permita as ligações de outros servidores através destas portas. Vamos demonstrar os passos para a porta TCP 1433, passos estes que serão repetidos para configurar a regra da porta UDP 1434.

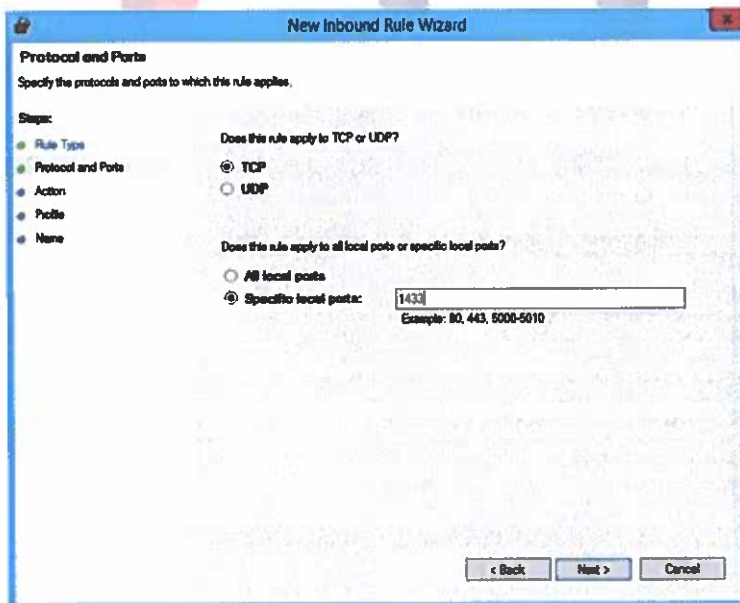


Figura 129: Assistente de Configuração Nova Regra da Firewall 2, Fonte Própria

Por fim atribuímos um nome à regra criada na *firewall*, de modo a que seja perceptível iremos dar-lhe o nome SQL TCP.

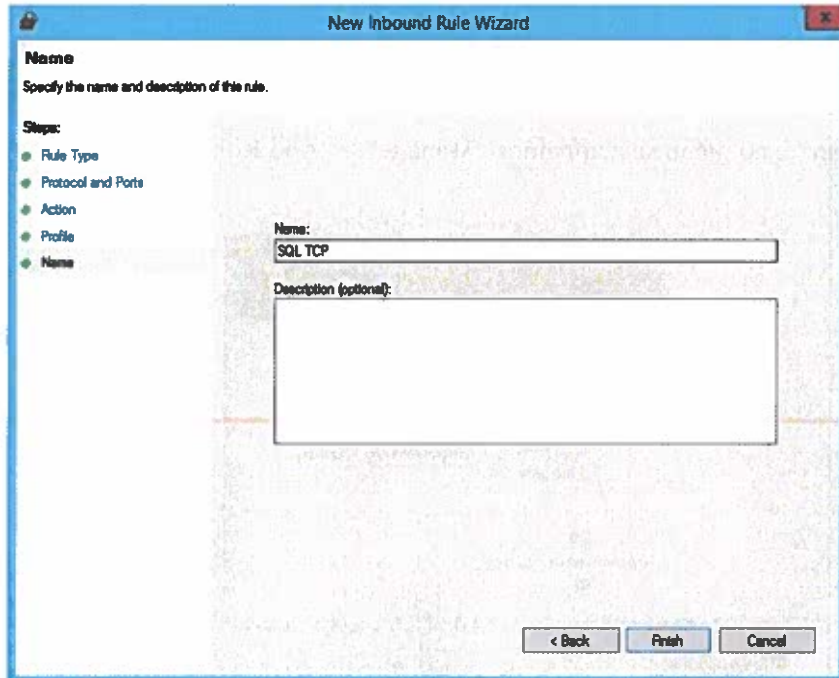


Figura 130: Assistente de Configuração Nova Regra da Firewall 3, Fonte Própria

No fim de ambas as regras criadas, para as portas TCP e UDP, poderemos verificar as mesmas já no quadro de regras da *firewall*.

Inbound Rules													
Name	Grúp	Profile	Enabled	Action	Override	Program	Local Addre...	Remote Address	Protocol	Local Port	Remote Port	Ac	
SQL UDP		Domain	Yes	Allow	No	Any	Any	Any	UDP	1434	Any	Ar	
SQL TCP		Domain	Yes	Allow	No	Any	Any	Any	TCP	1433	Any	Ar	

Figura 131: Windows Firewall 2, "Inbound Rules", Fonte Própria

4.2.3.4 Servidor Virtual Machine Manager

Neste servidor iremos utilizar a capacidade de executar um *hypervisor* dentro de uma máquina virtual, a esta funcionalidade chamamos de *nested virtualization*.

Para efeitos do projeto, a configuração do SCVMM 2012 R2 será *stanhd-alone* no servidor VMMPG1946, em que o próprio servidor será o único *Host* do SCVMM. Neste documento não iremos abordar as melhores práticas, nem as recomendações para configuração do SCVMM 2012 R2. Apenas se pretende ter um sistema para criação de máquinas cliente através de um *template*.

4.2.3.4.1 Adicionar a Role Hyper-V

Para que este servidor possa ser o *Host* do SCVMM, teremos de lhe adicionar a *role* do *Hyper-V*.

Para dar início a este processo, dentro do servidor virtual VMMPG1946, iremos ao “Service Manager” e no menu seleccionamos “Manage” > “Add Roles and Features”:

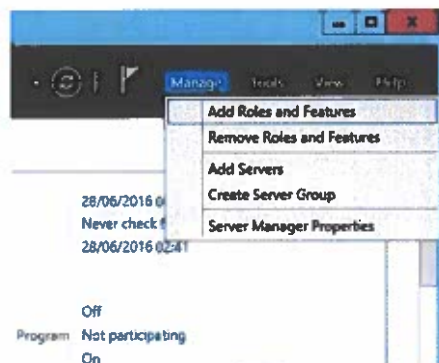


Figura 132: *Server Manager* 3, “Add Roles and Features”, Fonte Própria

Iniciado o assistente de configuração “Add Roles and Features”, todos os passos que não são mencionados, serão mantidas as configurações padrão. Chegando à secção “Server Roles”, no quadro “Roles”, iremos seleccionar a opção “Hyper-V”:

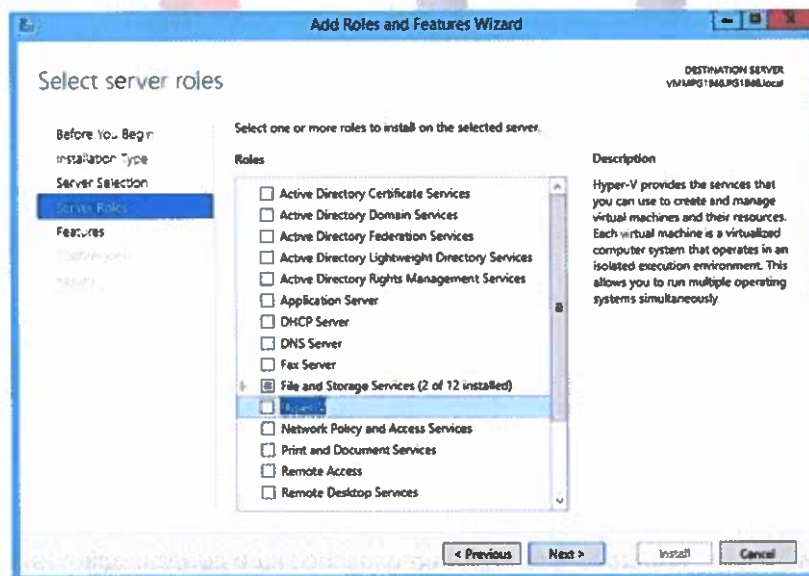


Figura 133: Assistente de configuração “Add Roles and Features”, Hyper-V 1, Fonte Própria

Irá surgir uma janela a indicar as “Roles and Features” que serão instalas com a seleção desta opção:



Figura 134: Assistente de configuração “Add Roles and Features”, Hyper-V 2, Fonte Própria

Depois de termos carregado no botão “Add Features” da janela anterior, ficaremos com a opção “Hyper-V” selecionada:

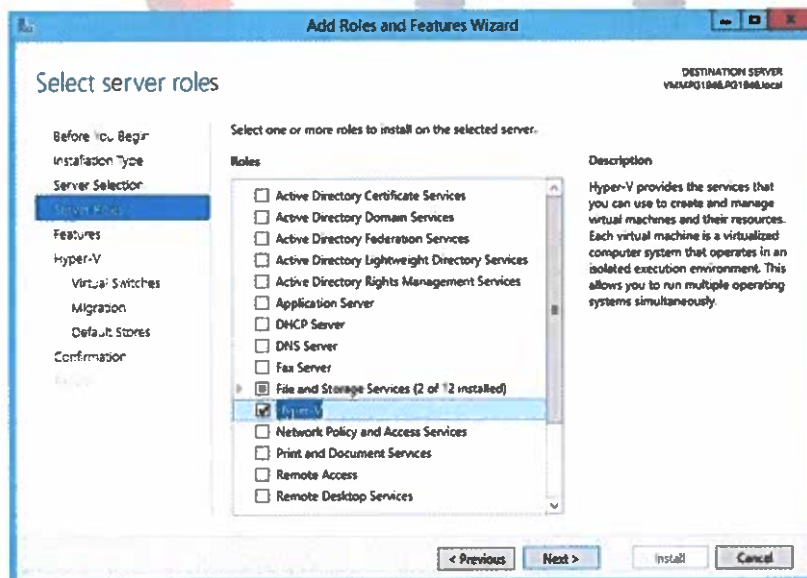


Figura 135: Assistente de configuração “Add Roles and Features”, Hyper-V 3, Fonte Própria

Vamos seleccionar ambas as interfaces de rede LAN e NAT, para que seja criados um *Switch* Virtual para cada uma delas no Hyper-V.

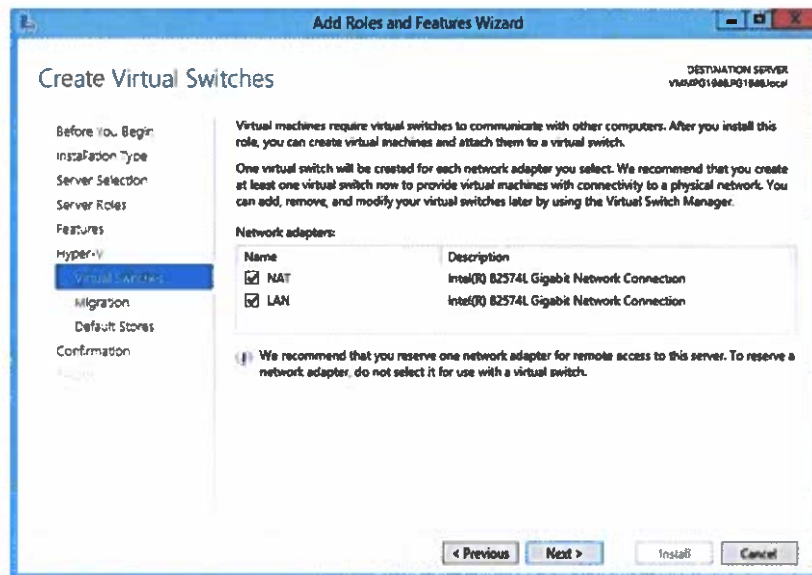


Figura 136: Assistente de configuração “Add Roles and Features”, Hyper-V 4, Fonte Própria

Uma vez que apenas teremos um *Host*, não iremos fazer migrações de máquinas criadas no SCVMM, pelo que vamos deixar desseleccionada a opção “Allow this server to send and receive migrations of virtual machines”.

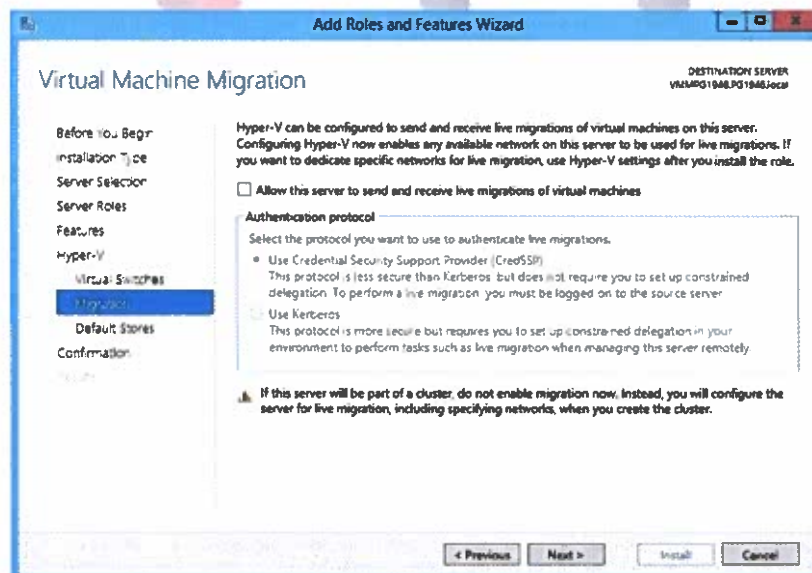


Figura 137: Assistente de configuração “Add Roles and Features”, Hyper-V 5, Fonte Própria

Neste passo iremos definir a localização do servidor onde ficaram alojados os ficheiros dos discos e configurações das máquinas virtuais criadas no Hyper-V.

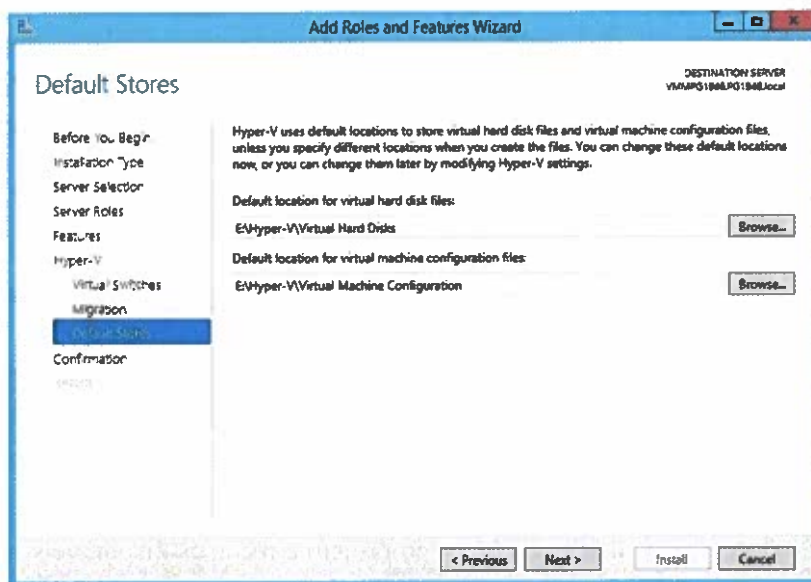


Figura 138: Assistente de configuração “Add Roles and Features”, Hyper-V 6, Fonte Própria

No final, temos um quadro com o resumo de todas as “Roles and Features” que serão instaladas:

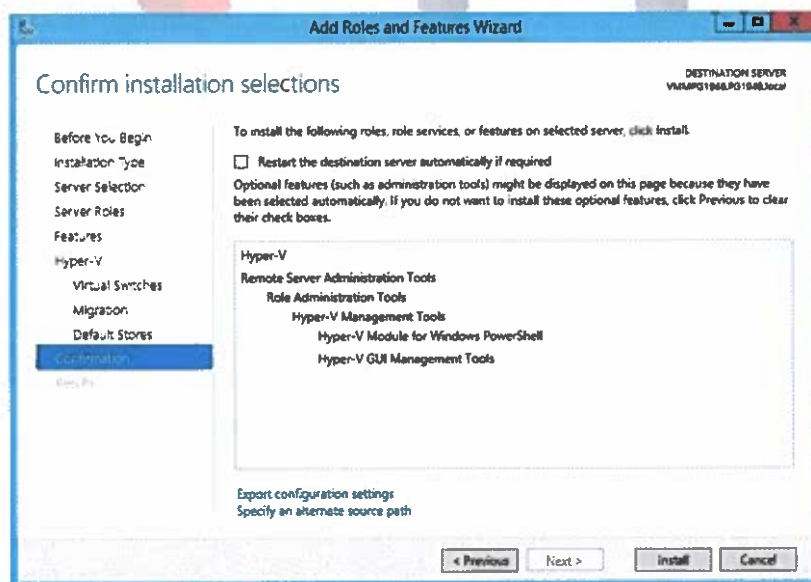


Figura 139: Assistente de configuração “Add Roles and Features”, Hyper-V 7, Fonte Própria

4.2.3.4.2 Instalação do SCVMM 2012 R2

Para darmos início à instalação do SCVMM 2012 R2 serão necessários três requisitos fundamentais, o primeiro já foi contemplado num dos passos anteriores, com a instalação do motor de base de dados, o segundo foi também já contemplado no passo anterior, com a ativação da role Hyper-V num dos servidores para ser o *Host* no SCVMM, que neste caso será o mesmo servidor que terá o SCVMM 2012 R2 instalado. O terceiro requisito é a instalação de *software*, neste caso será o “Windows Assessment and Deployment Kit (ADK) for Windows 8.1” e para podermos despoletar serviços que utilizem a camada de dados do SQL Server 2012 vamos instalar dois componentes de conectividade do “Microsoft SQL Server 2012 Feature Pack”, neste caso será o “Microsoft SQL Server 2012 Native Client” e o “Microsoft SQL Server 2012 Command Line Utilities”.(Microsoft Corporation, n.d.-g)

Num ambiente de produção aconselho a instalação deste serviço numa máquina dedicada a este serviço. Por uma questão de flexibilidade do projeto e recursos disponíveis iremos instalar este serviço no servidor VMMPG1946, que será também o *Host* do SCVMM.

O *software* necessário para instalação foi previamente colocado numa pasta partilhada do Controlador de Domínio, DCPG1946, de modo a poder ser acedida através a partir do servidor VMMPG1946, para instalação no mesmo.

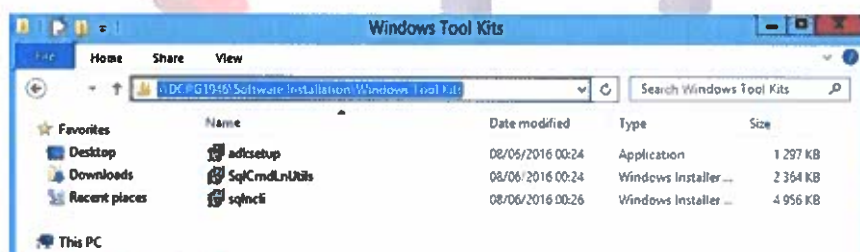


Figura 140: Windows Assessment and Deployment Kits (ADK), Fonte Própria

Para podermos dar início à instalação do SCVMM 2012 R2, vamos como foi referido atrás instalar o *software* necessário para o funcionamento do mesmo. Neste caso vamos iniciar pela instalação do “Windows Assessment and Deployment Kit (ADK) for Windows 8.1”.

Iniciado o assistente de instalação do “Windows Assessment and Deployment Kit for Windows 8.1”, vamos seleccionar a localização onde pretendemos que seja instalado o Kit.

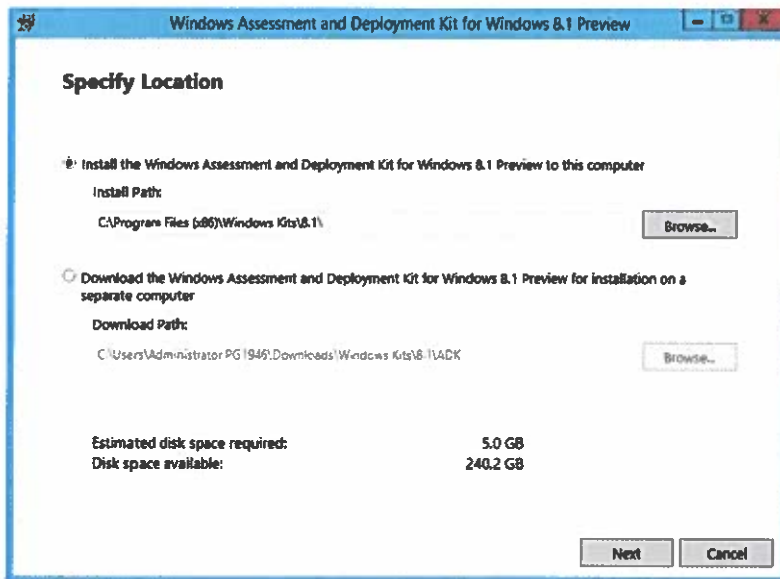


Figura 141: Instalação do Windows ADK 1, Windows 8.1, Fonte Própria

No próximo passo do assistente de instalação, vamos as *features* que pretendemos que sejam instaladas, neste caso vamos selecionar as opções “Deployment Tools” e “Windows Preinstallation Environment (Windows PE)”.

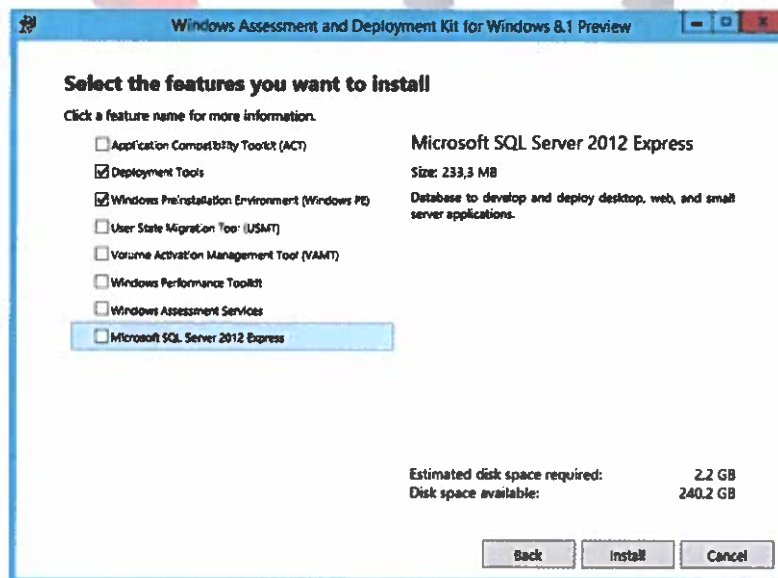


Figura 142: Instalação do Windows ADK 2, Windows 8.1, Fonte Própria

Após terminar a instalação do “Windows Assessment and Deployment Kit (ADK) for Windows 8.1”, vamos dar início à instalação do “Microsoft SQL Server 2012 Native Client”. Iniciando o assistente de instalação iniciado carregamos no botão “Next”:

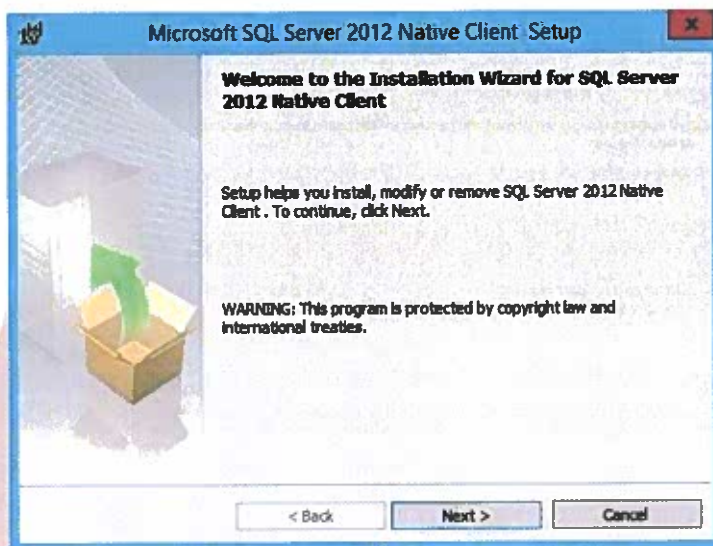


Figura 143: Instalação Microsoft SQL Server 2012 Native Client 1, Fonte Própria

No passo “Feature Selection” deixamos a opção por defeito que é a instalação do “SQL Server Native Client SDK”.

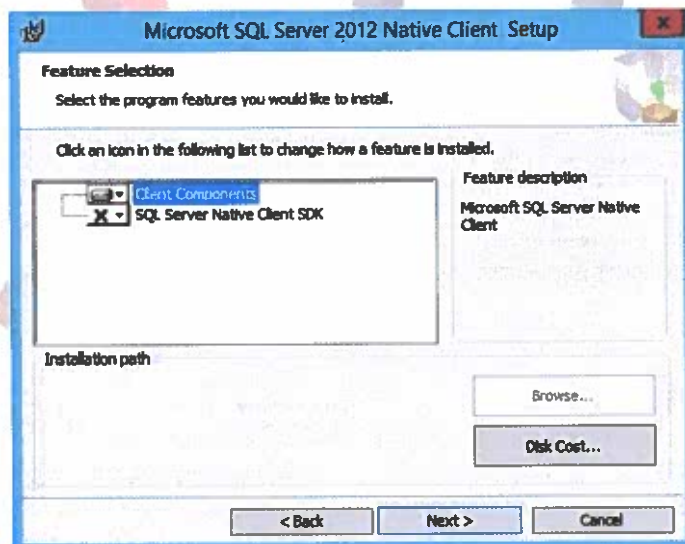


Figura 144: Instalação Microsoft SQL Server 2012 Native Client 2, Fonte Própria

Por último, vamos proceder à instalação do último *software* que é requisito para o funcionamento do SCVMM 2012 R2, vamos então passar à instalação do “Microsoft SQL Server 2012 Command Line Utilities”. A instalação deste *software* é simples sem ser necessário selecionar opções, para além de concordar com os Termos da Licença pelo que não iremos mostrar os passos de instalação sendo apenas selecionar a opção que concorda com os Termos da Licença e carregar no botão “Next”, finalizando com o botão “Install”. Mostramos apenas uma imagem início do assistente de instalação.

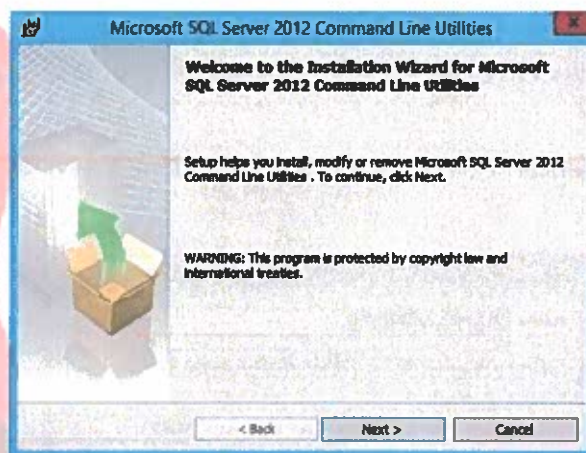


Figura 145: Instalação Microsoft SQL Server 2012 Command Line, Fonte Própria

Antes de iniciarmos a instalação do SCVMM neste servidor, teremos de ir ao servidor DCPG1946 que é o Controlador de Domínio e criar um novo contentor, este contentor que iremos criar será mais à frente utilizado para serem armazenadas as chaves de encriptação de informação das palavras-chave das contas e a informação das licenças dos Sistemas Operativos das máquinas criadas através do SCVMM. Para iniciarmos a criação do contentor no menu do “Server Manager” selecionamos “Tools” > “ADSI Edit”.

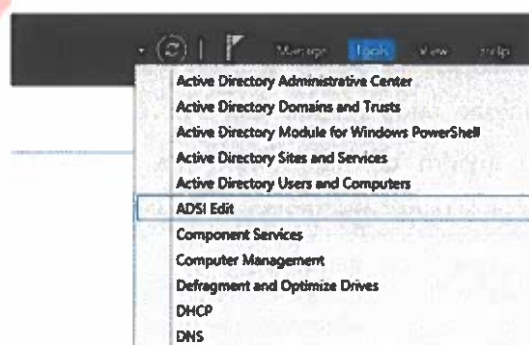


Figura 146: Server Manager, ADSI Edit, Fonte Própria

Na janela do “ADSI Edit” selecionamos no quadro do lado esquerdo “ADSI Edit” e escolhemos a opção “Connect to”.

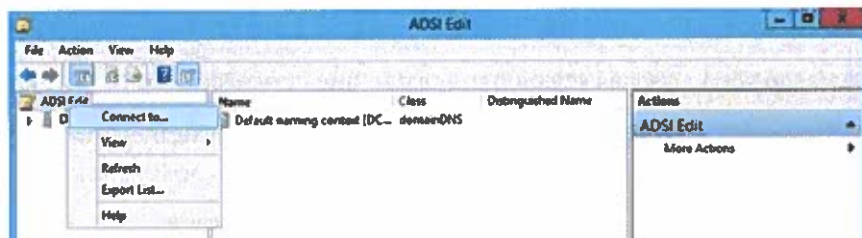


Figura 147: ADSI Edit 1, Criar Contentor, Fonte Própria

Abre-se uma nova janela “Connection Setting” e mantemos os valores padrão e carregamos apenas no botão “OK”.

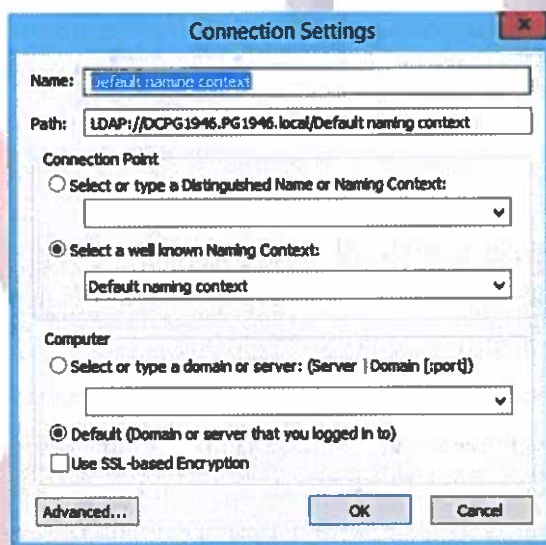


Figura 148: ADSI Edit 2, Criar Contentor, Fonte Própria

Neste momento selecionando o “Default naming context” no quadro do lado esquerdo já temos a informação do caminho onde ficarão todos os contentores. Para criarmos o nosso contentor, selecionamos no quadro do lado direito na coluna “Name” o nosso domínio “DC=PG1946,DC=local” e com o botão direito do rato selecionamos “New” > “Object”.



Figura 149: ADSI Edit 3, Criar Contentor, Fonte Própria

Iniciado o assistente de configuração “Create Object”, para criarmos um objeto do tipo contentor como pretendemos, vamos selecionar a opção “container”.

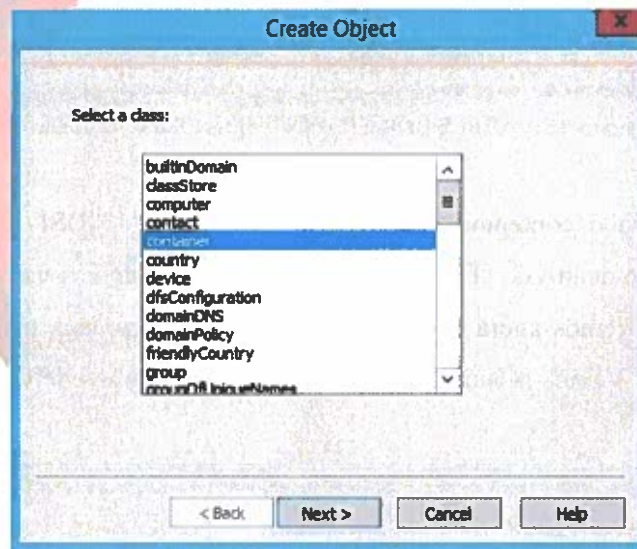


Figura 150: ADSI Edit 4, Criar Contentor, Fonte Própria

No passo seguinte vamos definir o nome que pretendemos dar ao contentor que estamos a criar, neste caso daremos o nome “VMMDKM”.

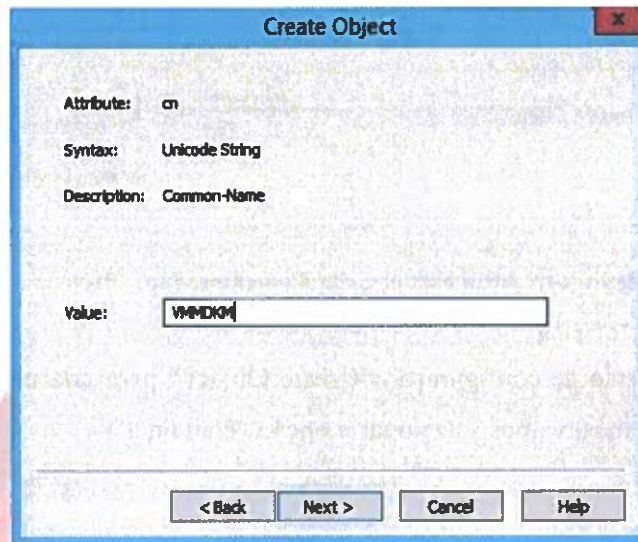


Figura 151: ADSI Edit 4, Criar Contendor, Fonte Própria

Depois de criado o contendor “VMMDKM”, na janela “ADSI Edit” selecionando o domínio, que se encontra dentro do “Default naming context”, podemos ver todos os contenedores existentes no domínio. Iremos agora adicionar um grupo de segurança para ter permissões de acesso a este contendor. Vamos selecionar o contendor que acabámos de criar e selecionamos a opção “Properties”.

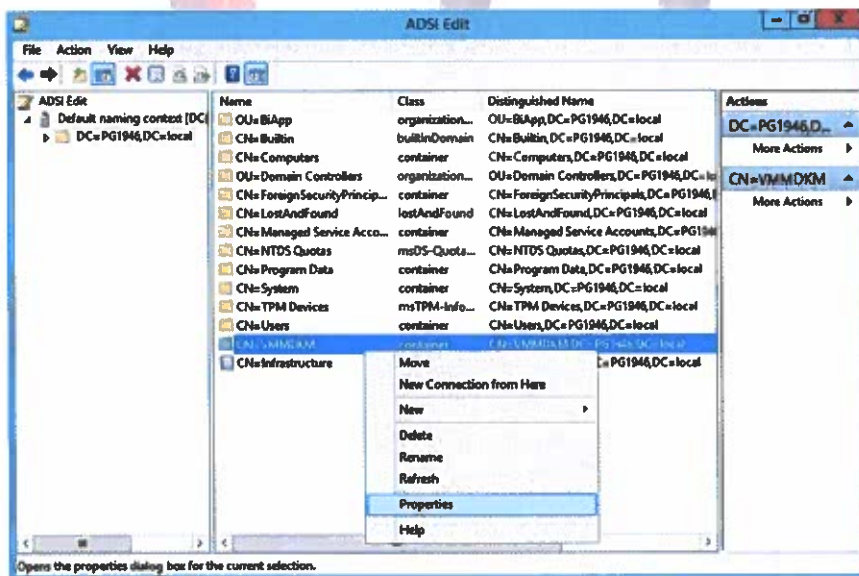


Figura 152: ADSI Edit 4, Criar Contendor, Fonte Própria

Nas propriedades do contendor que criámos vamos aceder às definições avançadas, para tal carregamos o botão “Advanced”.

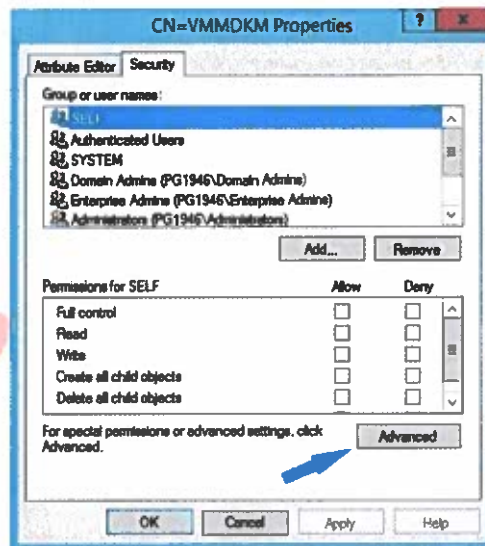


Figura 153: ADSI Edit 5, Criar Contendor, Fonte Própria

Nas definições avançadas do contendor, vamos carregar no botão “Add”.

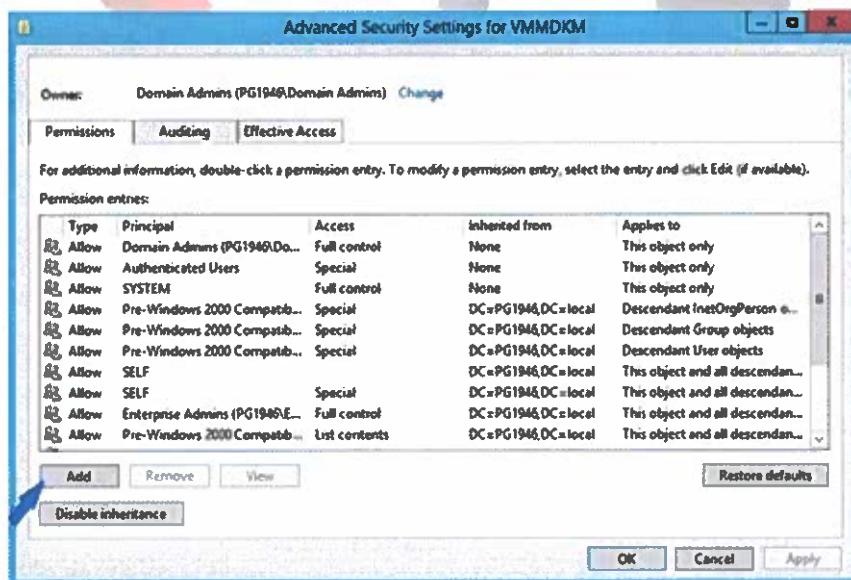


Figura 154: ADSI Edit 6, Criar Contendor, Fonte Própria

O próximo passo é adicionar o Grupo de Segurança que pretendemos adicionar para ter permissões no contentor que criámos. Vamos seleccionar o *link* “Select a principal” e seleccionamos o Grupo de Segurança já criado anteriormente com o nome “SCVMMAdmins”.

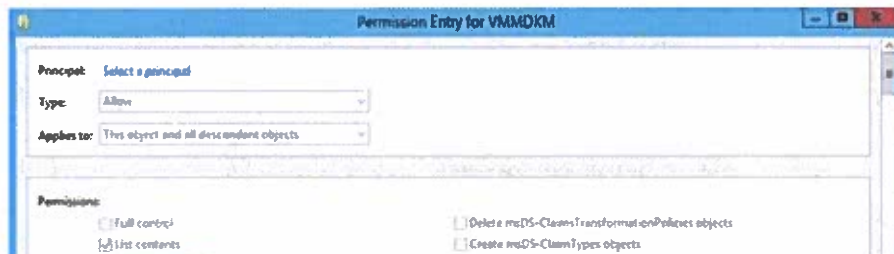


Figura 155: ADSI Edit 7, Criar Contentor, Fonte Própria

Após termos seleccionado o Grupo de Segurança, podemos ver que o mesmo já se encontra no campo “Principal”. Ainda no primeiro quadro vamos seleccionar no campo “Type”, a opção “Allow” e no campo “Applies to” vamos escolher a opção “This object and all descendant objects”. Em seguida no quadro “Permissions” vamos seleccionar a opção “Full control” deste modo o grupo adicionado terá todos os privilégios de acesso ao contentor criado. No fim carregamos no botão “OK”.

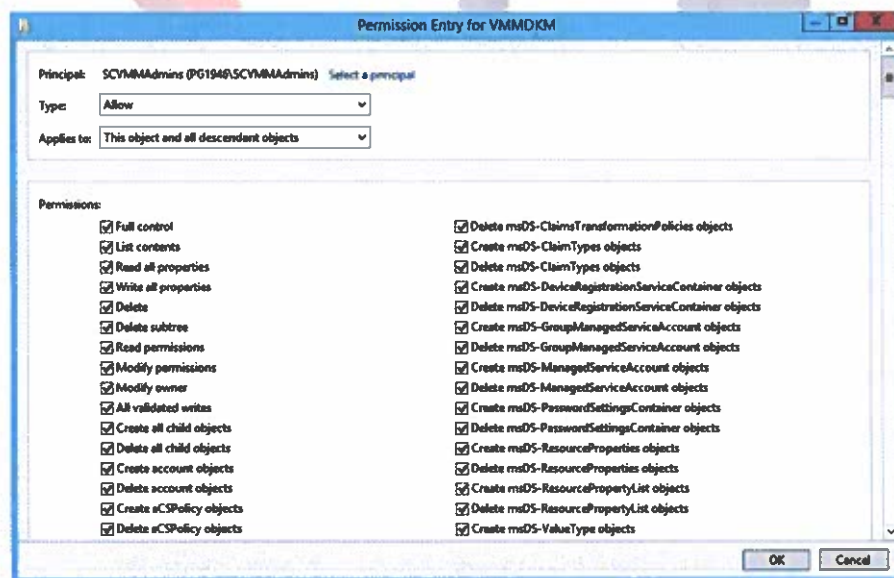


Figura 156: ADSI Edit 8, Criar Contentor, Fonte Própria

De volta à janela das definições avançadas do contentor, podemos ver que o Grupo de Segurança que adicionámos já lá se encontra. Carregamos no botão “OK” para aceitar as alterações efetuadas.

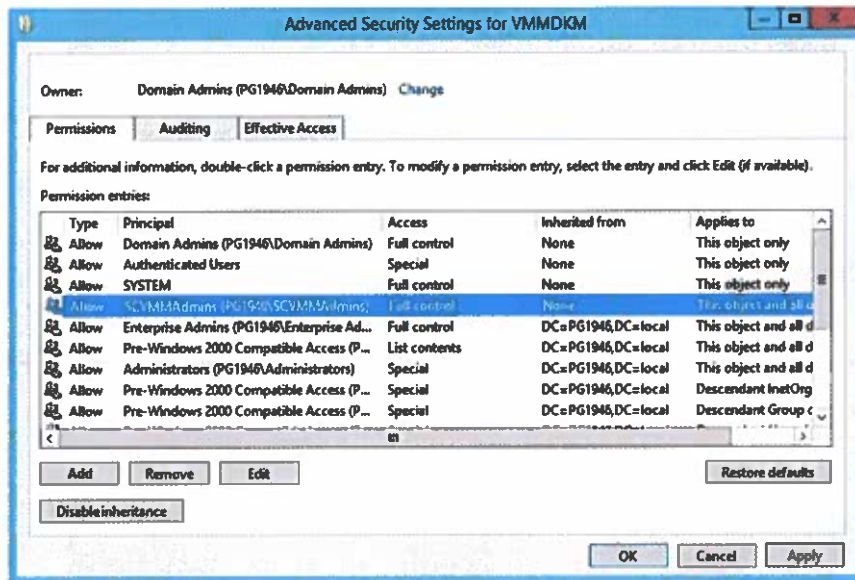


Figura 157: ADSI Edit 9, Criar Contentor, Fonte Própria

Por fim podemos confirmar que na janela de propriedades do contentor também já lá se encontra o nosso Grupo de Segurança adicionado e com privilégios totais.

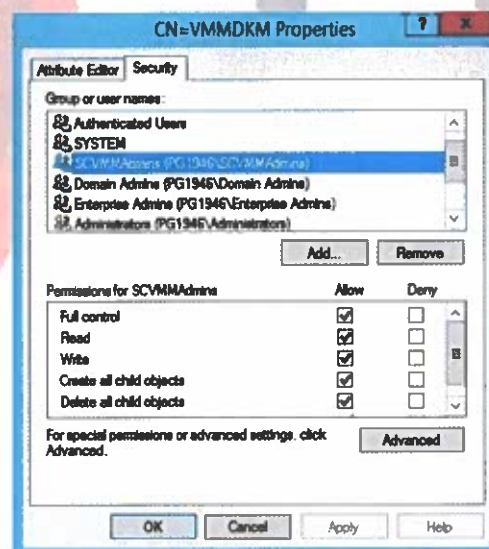


Figura 158: ADSI Edit 10, Criar Contentor, Fonte Própria

Tal como fizemos no Servidor SQLPG1946, para montar a imagem do “Microsoft SQL Server 2012”, iremos fazer os mesmos passos para montar a imagem do “Microsoft SCVMM 2012 R2” na drive CD/DVD do servidor VMMPG1946.

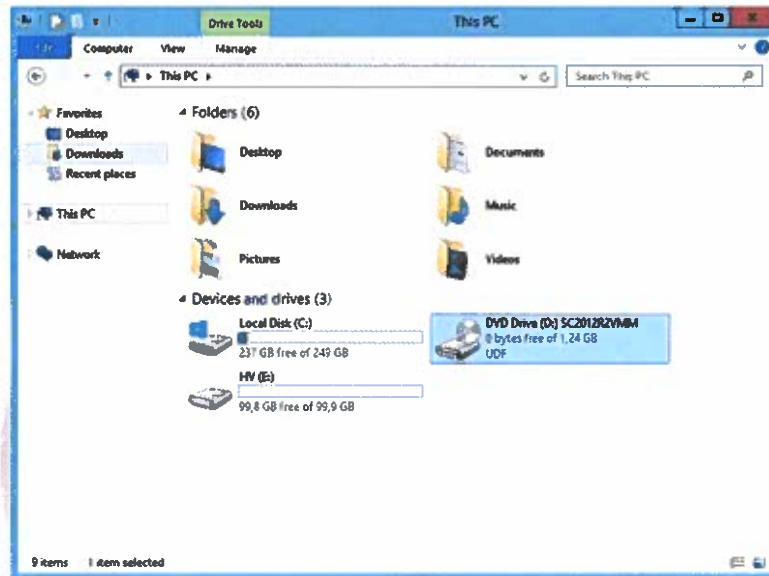


Figura 159: Instalação do SCVMM 2012 R2 1, Fonte Própria

No primeiro passo, após ter sido iniciado o assistente de instalação do “SCVMM 2012 R2”, vamos seleccionar as duas funcionalidades:

- O servidor controla todas as tarefas e processos, comunica com a base de dados, biblioteca e *hypervisors* que irão ser agregados ao sistema;
- Consola que nos permite a gestão dos recursos de uma forma centralizada e permite gerir e verificar toda a plataforma. Esta consola pode ser instalada localmente neste servidor ou remotamente noutro servidor.

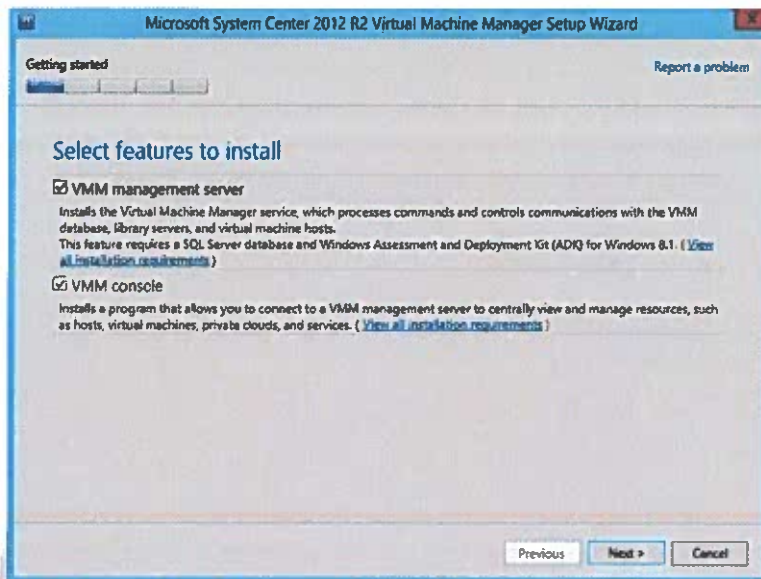


Figura 160: Instalação do SCVMM 2012 R2 2, Fonte Própria

Vamos instalar este produto da Microsoft com licenciamento, pelo que vamos introduzir a chave do produto.

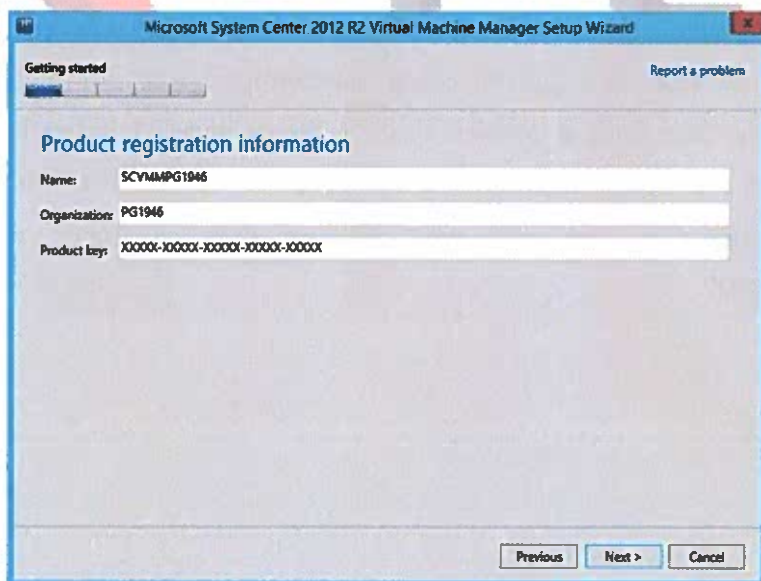


Figura 161: Instalação do SCVMM 2012 R2 3, Fonte Própria

É recomendado que selecionemos a opção “On” de modo a que sejam feitas todas as atualizações para um comportamento da aplicação mais estável e adaptado ao resto dos sistemas com que irá interagir.



Figura 162: Instalação do SCVMM 2012 R2 4, Fonte Própria

No próximo passo, é feita a configuração do motor de base de dados, que foi instalado anteriormente. O servidor onde está instalado o SQL Server e onde será criada a base de dados do SCVMM será o servidor SQLPG1946. As credenciais com as quais o SCVMM se irá ligar à instância da base de dados é a que foi criada anteriormente para este efeito com o nome “sql_scvmm”. Iremos selecionar a instância do SQL Server onde irá ficar a base de dados do SCVMM, que será a já criada anteriormente com o nome “MSSQLPG1946”. Vamos deixar selecionada a opção para criar uma nova base de dados e iremos dar-lhe o nome “SCVMMPG1946BD”.

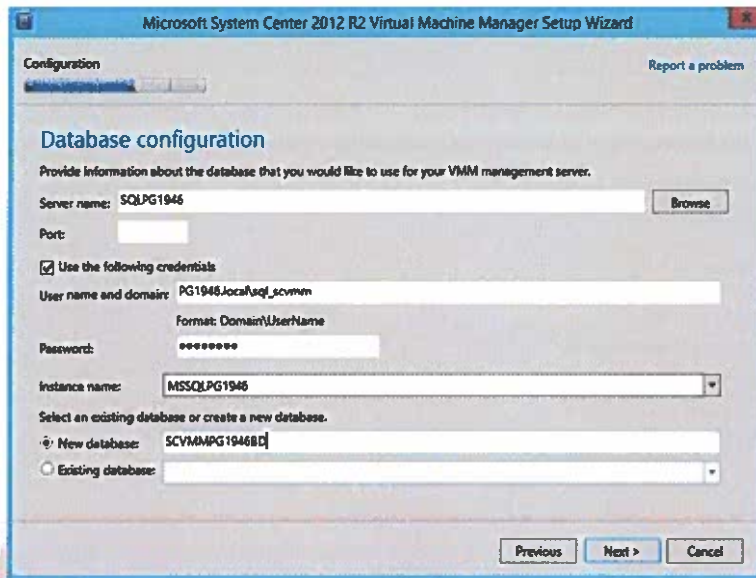


Figura 163: Instalação do SCVMM 2012 R2 5, Fonte Própria

No próximo quadro no campo “Domain Account” vamos configurar a conta de domínio, que criámos anteriormente, com a qual o serviço SCVMM irá correr. Selecionamos a opção “Store my keys in Active Directory” de modo a guardar as chaves de encriptação de informação das palavras-chave das contas e as informações das licenças dos Sistemas operativos, que serão utilizadas para configurações de máquinas virtuais. Por defeito, o SCVMM encripta alguns dados na base de dados, como por exemplo as credenciais e palavra-passe da conta Runas. Esta encriptação de dados está associada ao computador no qual o SCVMM está instalado e à conta de serviço que está a usar. Assim, no caso de ser necessário mover a instalação do SCVMM para outro servidor, os dados encriptados não serão mantidos. Nesse cenário, teremos de voltar a introduzir manualmente estes dados, como as credenciais, palavras-chave e chaves de produto das máquinas virtuais, de modo a corrigir os objetos criados no SCVMM.

Com a gestão de chaves distribuída, o armazenamento dessas chaves de encriptação é feito no “Active Directory Domain Services (AD DS)”. Com esta opção, caso seja necessário mover a instalação do SCVMM para outro servidor, os dados encriptados serão mantidos, pois o outro servidor terá acesso às referidas chaves de encriptação no “Active Directory Domain Services (AD DS)”.

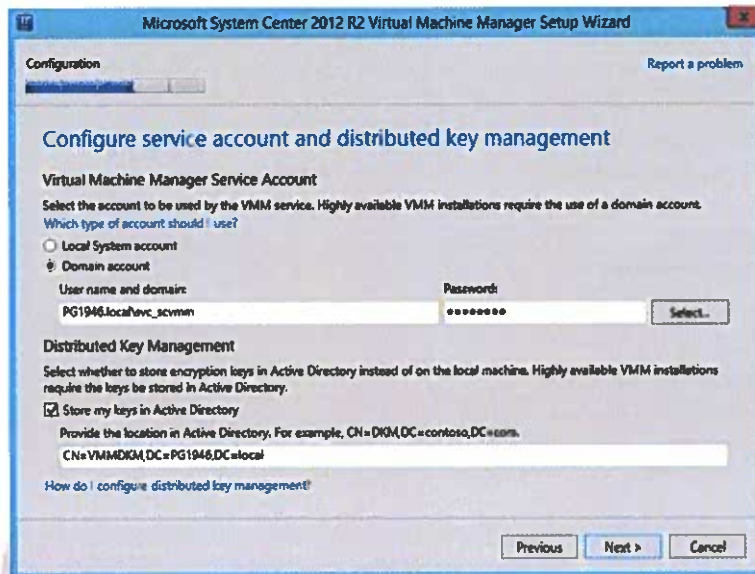


Figura 164: Instalação do SCVMM 2012 R2 6, Fonte Própria

No quadro que se segue, é dada a opção para criar uma nova biblioteca ou indicarmos a localização de uma já existente, neste caso iremos escolher a opção de criar uma nova biblioteca, à qual daremos o nome “SCVMMPG1946Library”.

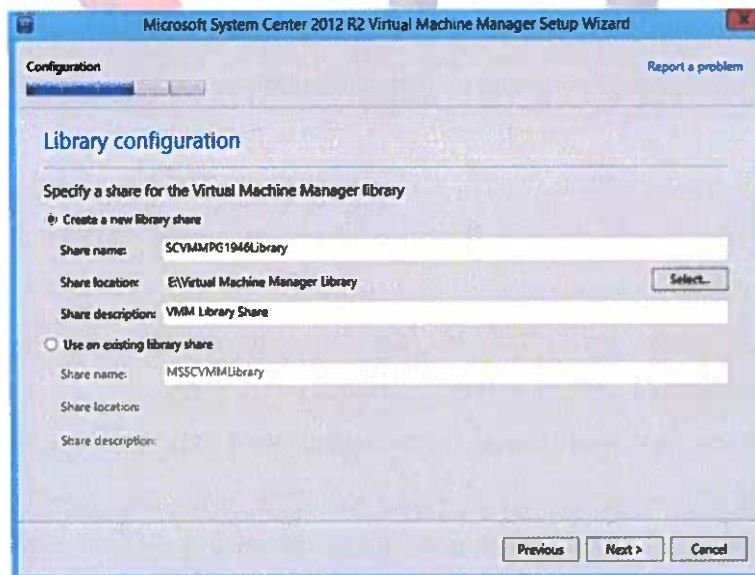


Figura 165: Instalação do SCVMM 2012 R2 7, Fonte Própria

Por fim, no último quadro temos um resumo com todas as definições e configurações com que será feita a instalação do SCVMM 2012 R2.

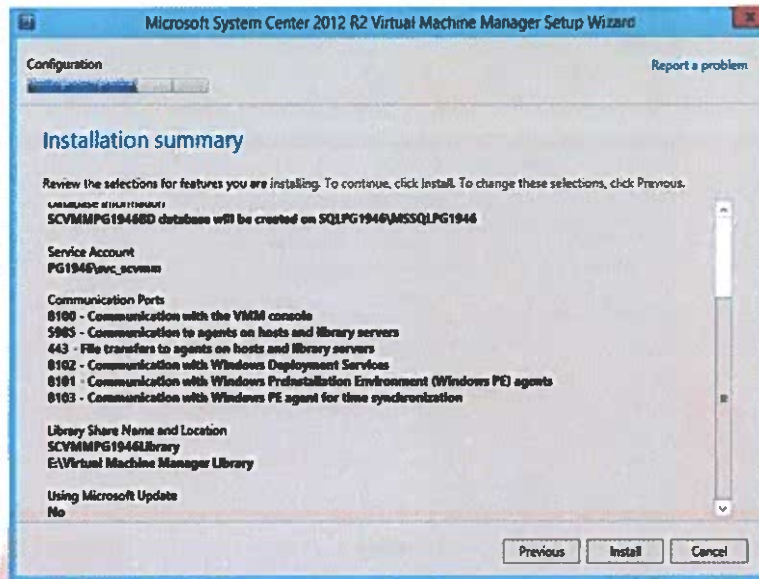


Figura 166: Instalação do SCVMM 2012 R2 8, Fonte Própria

Após terminar a instalação do SCVMM 2012 R2, poderemos verificar no *desktop* que já temos um atalho criado para o SCVMM 2012 R2.



Figura 167: Instalação do SCVMM 2012 R2 9, Fonte Própria

4.2.3.4.3 Configuração do SCVMM 2012 R2

Adicionar o *host* Hyper-V ao SCVMM

Vamos iniciar a configuração do SCVMM 2012 R2 adicionando o *host* Hyper-V. Seleccionamos no quadro de baixo do lado esquerdo o “Fabric”, em seguida no quadro de cima do lado esquerdo seleccionamos “All Hosts”, por fim escolhemos a opção “Add Hyper-V Hosts and Clusters”.

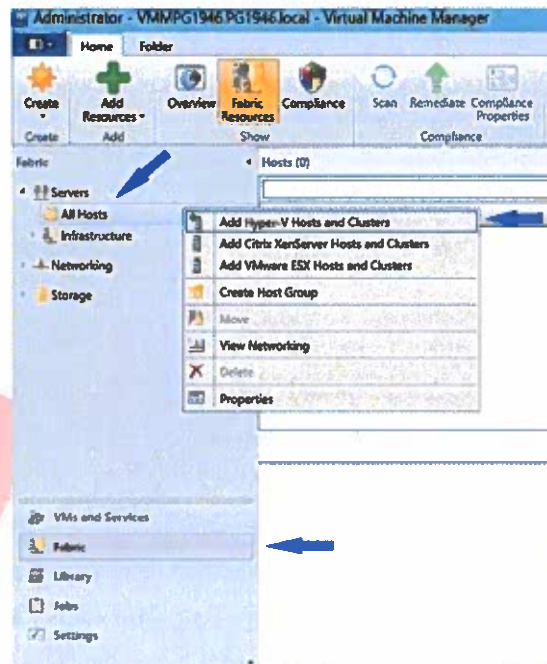


Figura 168: Adicionar *host* Hyper-V no SCVMM 1, Fonte Própria

Na janela inicial do assistente de configuração “Add Resource”, com o título “Resource Location”, é onde iremos adicionar o *host* Hyper-V. Selecionamos a opção “Windows Server computers in a trusted Active Directory domain”.

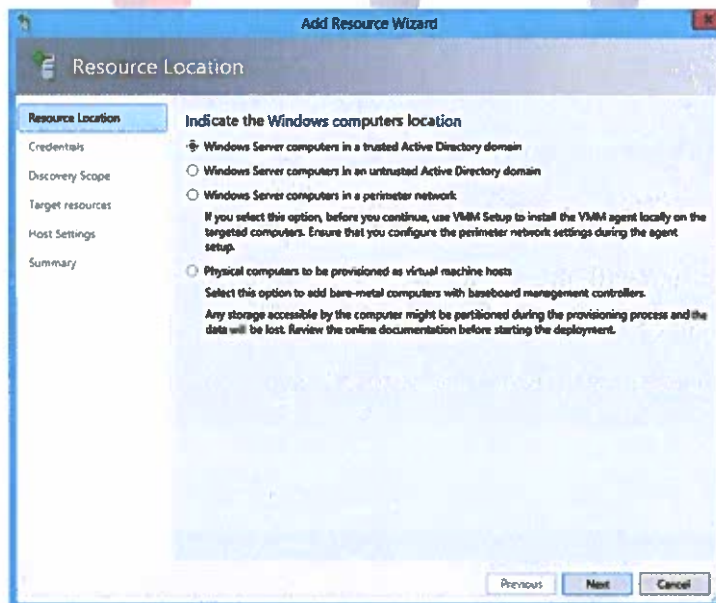


Figura 169: Adicionar *host* Hyper-V no SCVMM 2, Fonte Própria

No passo seguinte será onde iremos adicionar as credenciais com as quais o SCVMM vai comunicar com o *host*. Selecionamos a opção “Use na existing Ran As account” e carregamos no botão “Browse”.

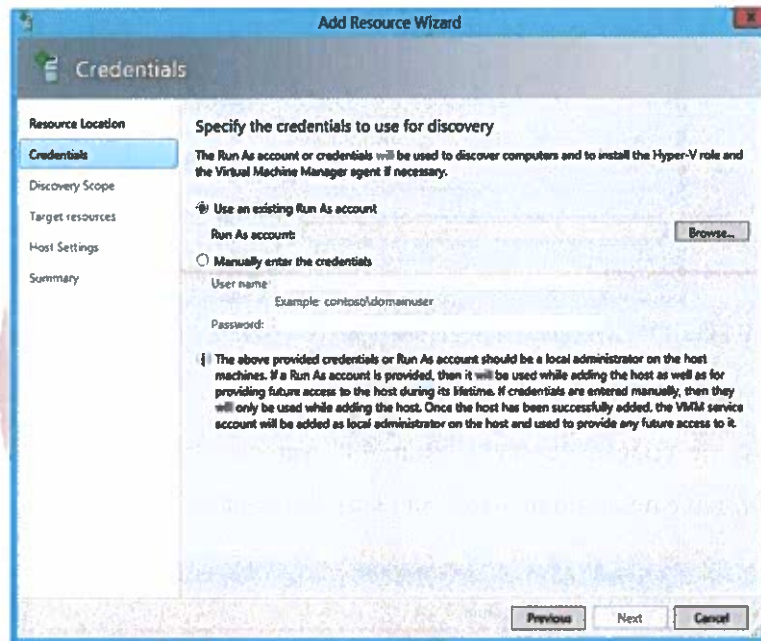


Figura 170: Adicionar *host* Hyper-V no SCVMM 3, Fonte Própria

Vamos adicionar a conta que com a qual as operações necessárias de futuro, possam ser realizadas utilizando as permissões necessárias. Vamos então carregar no botão “Create Run As Account” e adicionamos a conta.

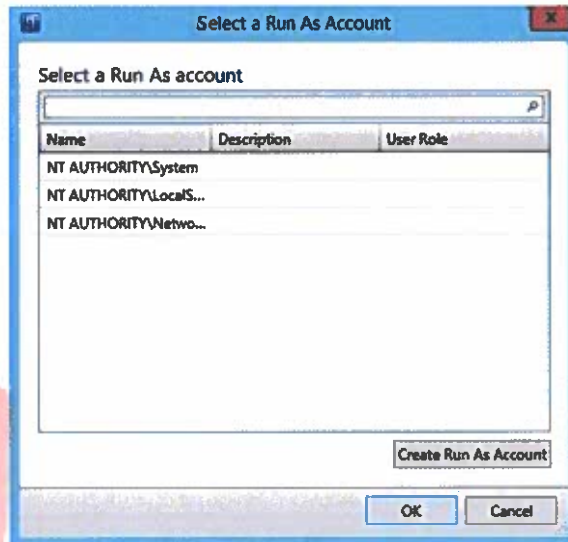


Figura 171: Adicionar *host* Hyper-V no SCVMM 4, Fonte Própria

No assistente de criação da conta “Run As”, vamos adicionar a conta `runas_scvmm`, configurada anteriormente no domínio, e inserimos a palavra-passe da conta.

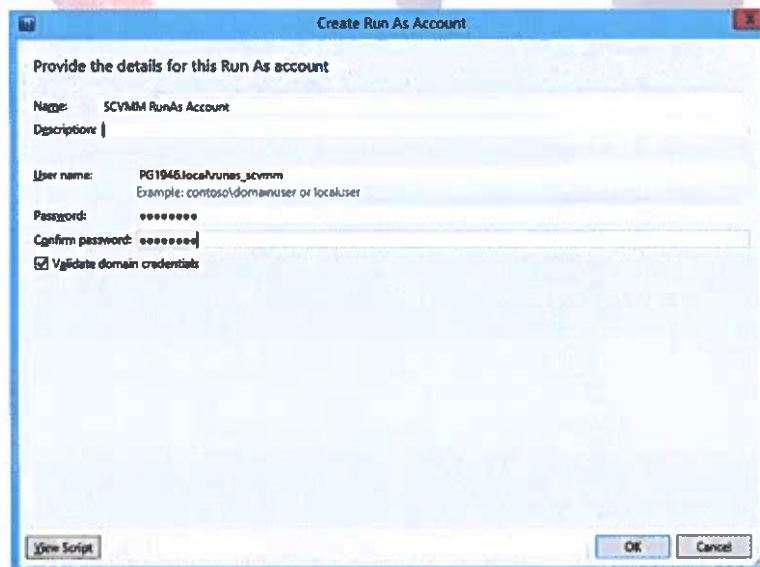


Figura 172: Adicionar *host* Hyper-V no SCVMM 5, Fonte Própria

No quadro “Select a Run As Account” já temos a conta que acabámos de adicionar. Vamos seleccionar a conta e carregar no botão “OK”.



Figura 173: Adicionar *host* Hyper-V no SCVMM 6, Fonte Própria

Voltando ao assistente de configuração “Add Resource” podemos verificar que a nossa conta já se encontra inserida como no campo “Run As Account”.

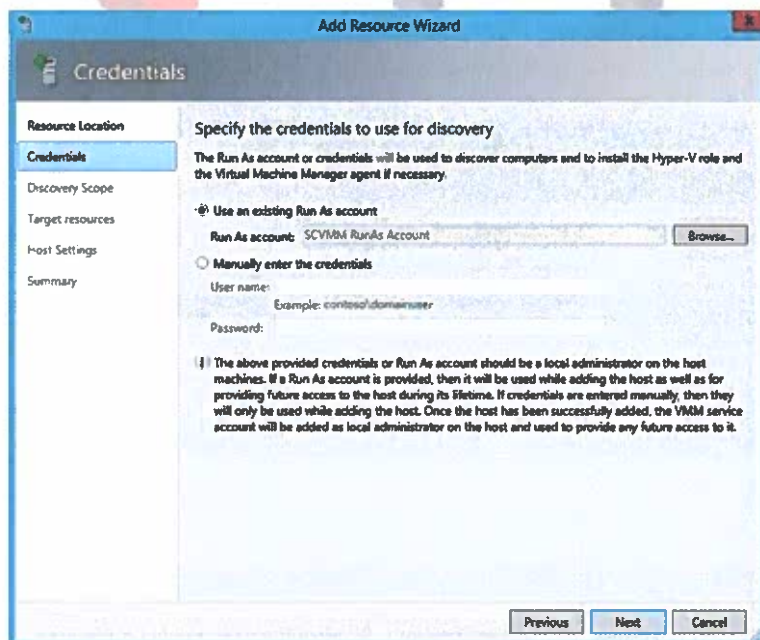


Figura 174: Adicionar *host* Hyper-V no SCVMM 7, Fonte Própria

O próximo passo, na caixa de texto com o nome “Computer names”, vamos introduzir o nome do nosso servidor *hypervisor* que será o *host* do SCVMM, neste caso será o mesmo servidor no qual estamos a instalar o SCVMM 2012 R2, VMMPG1946.

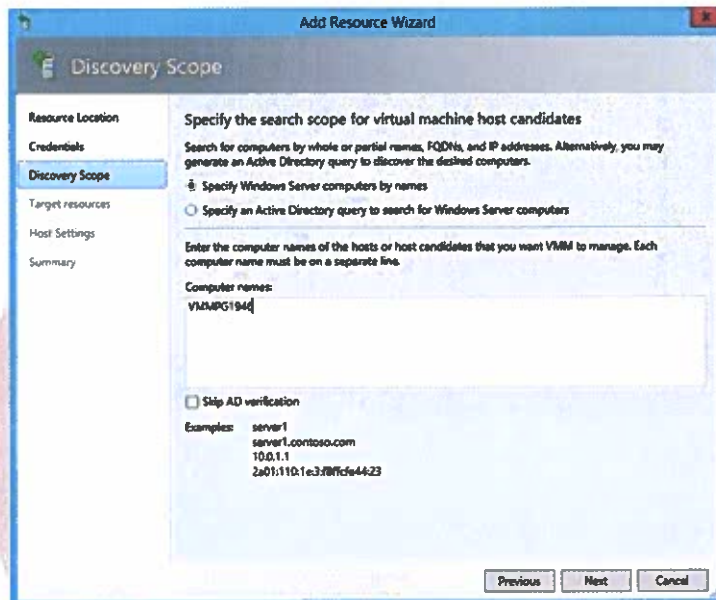


Figura 175: Adicionar *host* Hyper-V no SCVMM 8, Fonte Própria

O *hypervisor* inserido anteriormente é detetado pelo SCVMM, vamos então seleccioná-lo.

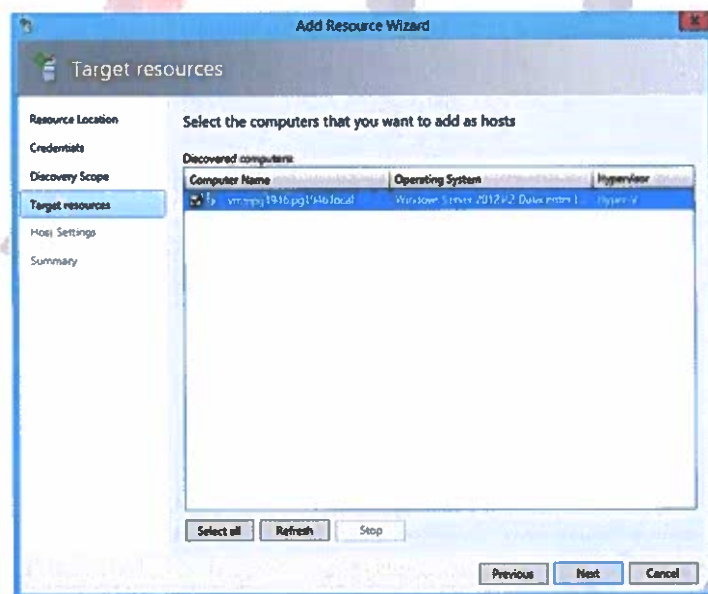


Figura 176: Adicionar *host* Hyper-V no SCVMM 9, Fonte Própria

Na secção “Hosts Settings”, verificamos o campo “Host group”, que é uma forma do servidor SCVMM gerir os seus *hosts* que estejam ou não agrupados em *clusters*. Vamos manter o “Host group” padrão, neste caso “All Hosts”.

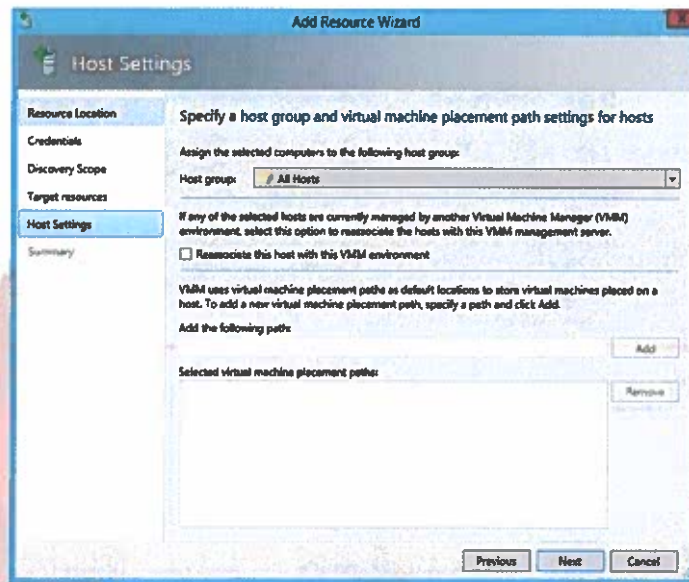


Figura 177: Adicionar *host* Hyper-V no SCVMM 10, Fonte Própria

Por fim, a última parte do assistente de configuração mostra-nos um resumo das configurações e opções selecionadas.

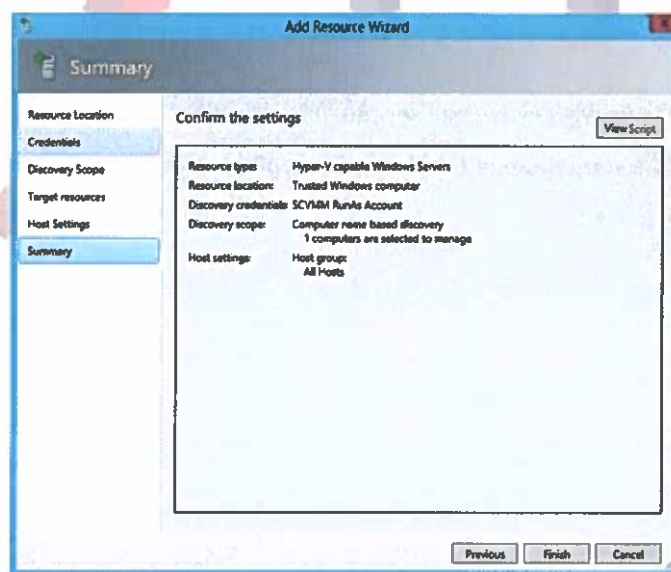


Figura 178: Adicionar *host* Hyper-V no SCVMM 11, Fonte Própria

Configurar a Rede do Hypervisor

Selecionamos no quadro de baixo do lado esquerdo a secção “VMs and Services”, em seguida no quadro de cima do lado esquerdo selecionamos o nosso *host* VMMPG1946, por fim escolhemos a opção “Properties”.

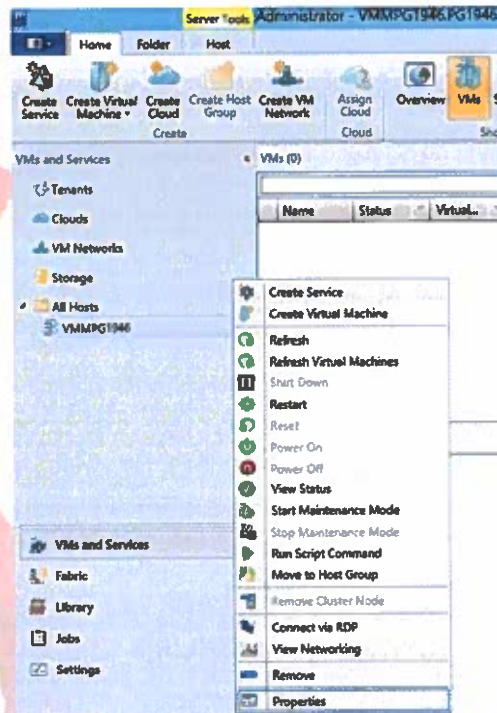


Figura 179: Configuração Rede do Hypervisor 1, Fonte Própria

Nas Propriedades do *hypervisor*, na secção “Hardware”, vamos estabelecer a ligação entre o interface de rede LAN e a rede lógica LAN. Iremos repetir os mesmos passos para a interface de rede NAT e a rede lógica NAT.

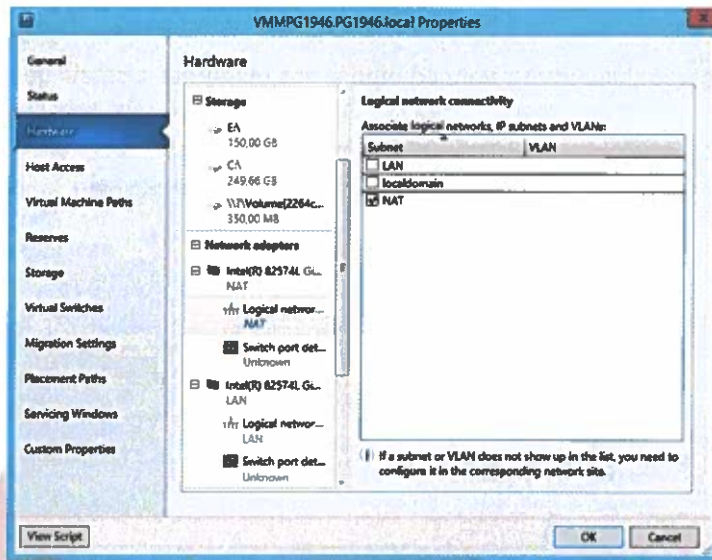


Figura 180: Configuração Rede do Hypervisor 2, Fonte Própria

Em seguida vamos à secção “Virtual Switch” e criamos dois *switchs* virtuais e vamos associá-los a cada uma das interfaces de rede do servidor, estes *switchs* virtuais irão interligar as interfaces lógicas do Hyper-V com as interfaces de rede do servidor onde ele está instalado. Vamos criar o *switch* virtual com o nome “V-LAN” e associá-lo à interface de rede LAN. Iremos repetir os mesmos passos para criar outro *switch* virtual com o nome “V-NAT” e associá-lo à interface de rede NAT.

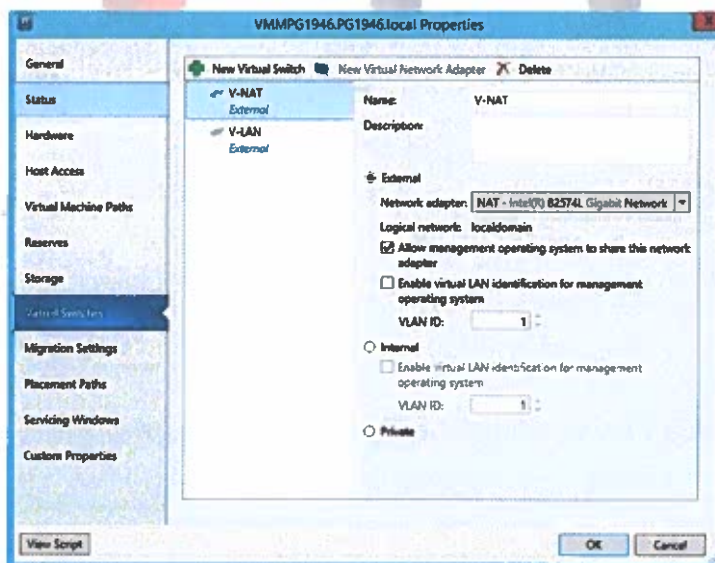


Figura 181: Configuração Rede do Hypervisor 3, Fonte Própria

Dentro do menu do “VMs and Services”, podemos ver na secção “VM Networks” as duas redes virtuais que estão já disponíveis para adicionar nas máquinas virtuais que iremos criar mais à frente.

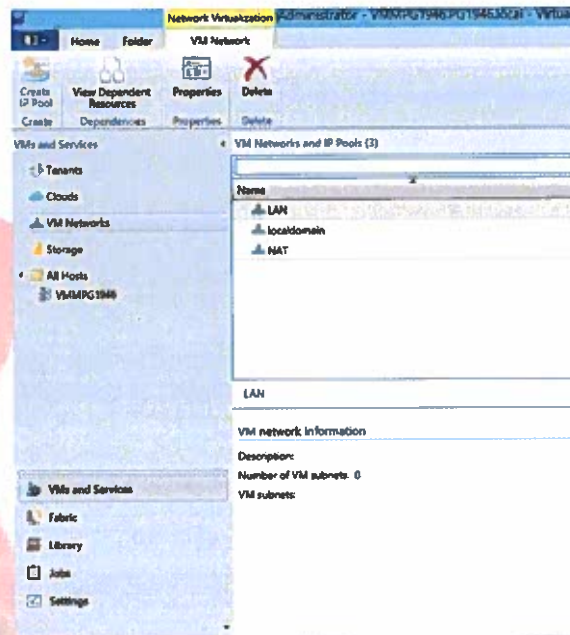


Figura 182: Configuração Rede do Hypervisor 4, Fonte Própria

Organizar Biblioteca do SCVMM

De modo a organizarmos a biblioteca do SCVMM, vamos criar algumas pastas. Vamos ao menu dentro do “Library”, selecionamos a nossa biblioteca e escolhemos a opção “Explore”.

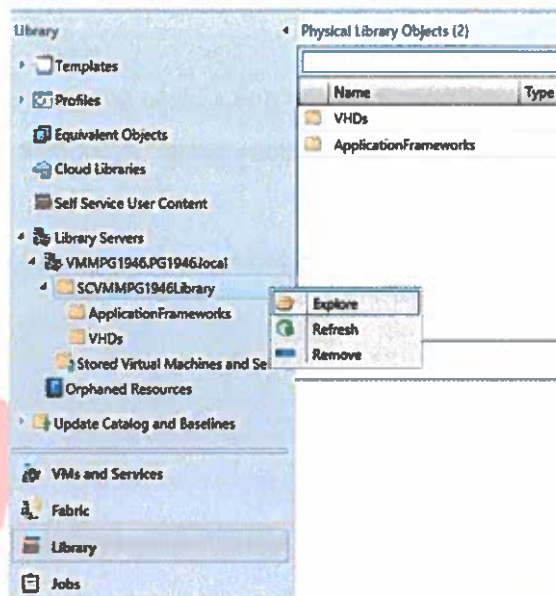


Figura 183: Organizar Biblioteca do SCVMM 1, Fonte Própria

Iremos criar duas novas pastas na nossa biblioteca, às quais iremos dar o nome “ISO” e “Biblioteca Template”.

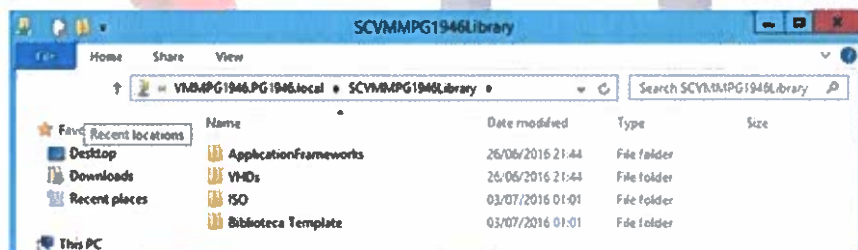


Figura 184: Organizar Biblioteca do SCVMM 2, Fonte Própria

Dentro da pasta que acabamos de criar com o nome “ISO”, iremos colocar a imagem do Sistema Operativo a instalar nas máquinas cliente.

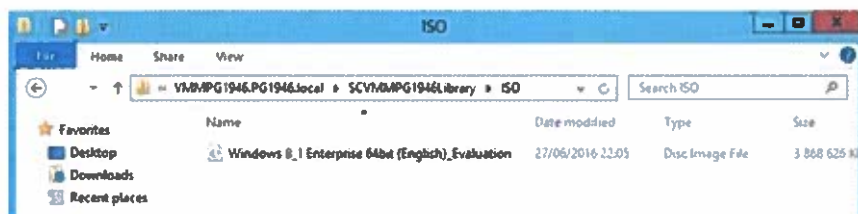


Figura 185: Organizar Biblioteca do SCVMM 3, Fonte Própria

Voltando ao SCVMM, podemos verificar que a pasta que criamos com o nome “ISO” já está visível dentro da biblioteca no SCVMM. A outra pasta “Biblioteca Template” por ainda não ter conteúdo o SCVMM não a coloca visível.

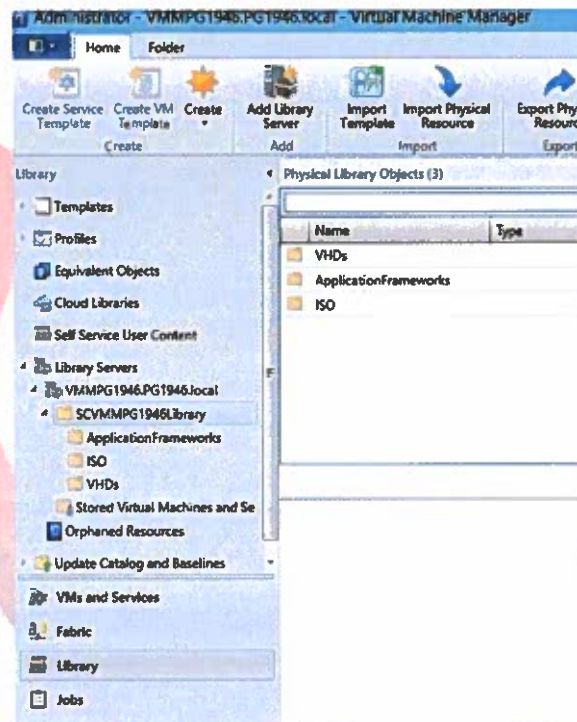


Figura 186: Organizar Biblioteca do SCVMM 4, Fonte Própria

Criar uma Máquina Virtual para *Template*

Selecionando o nosso *host* dentro do menu do “VMs and Services”, escolhemos a opção “Create Virtual Machine”.

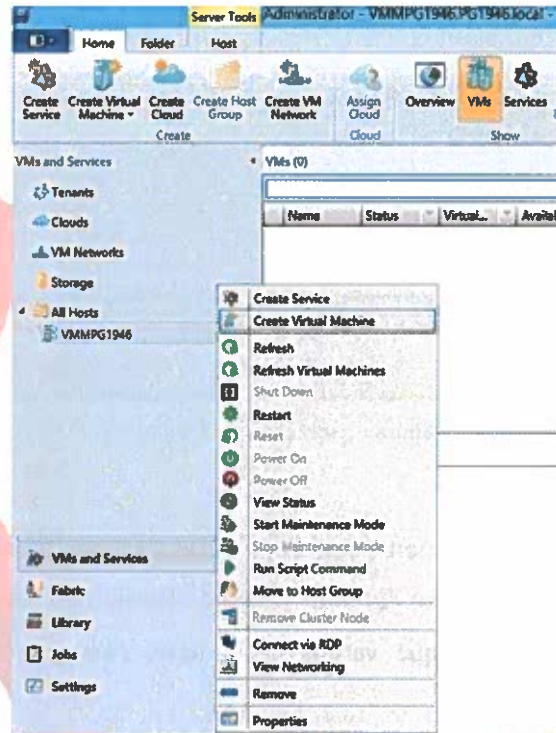


Figura 187: Criar Máquina Virtual para *Template* 1, Fonte Própria

Já iniciado o assistente de configuração “Create Virtual Machine”, vamos selecionar a opção “Create the new virtual machine with a blank virtual hard disk”.

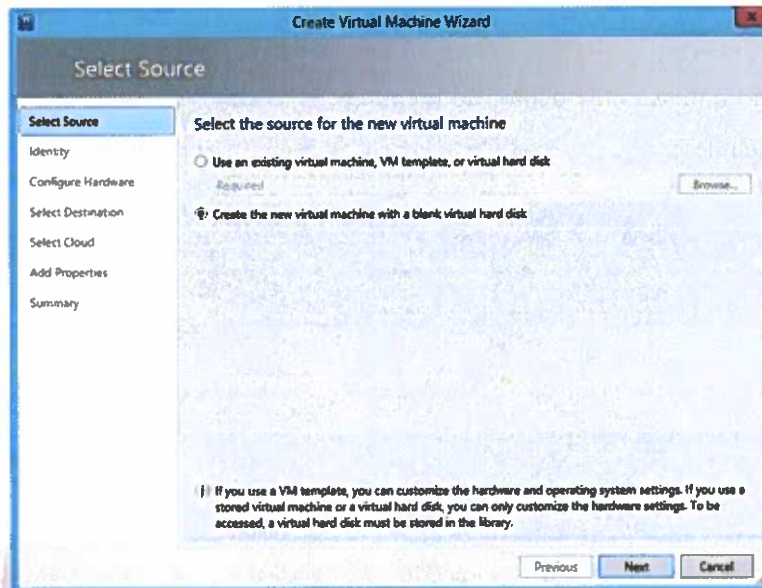


Figura 188: Criar Máquina Virtual para *Template 2*, Fonte Própria

Em seguida, vamos inserir o nome que iremos dar a esta máquina virtual que será usada para criar o nosso *template*, neste caso daremos o nome “Biblioteca Template”. Vamos também selecionar a geração da máquina que vamos criar, neste caso para efeitos de laboratório, escolhamos a geração 1.

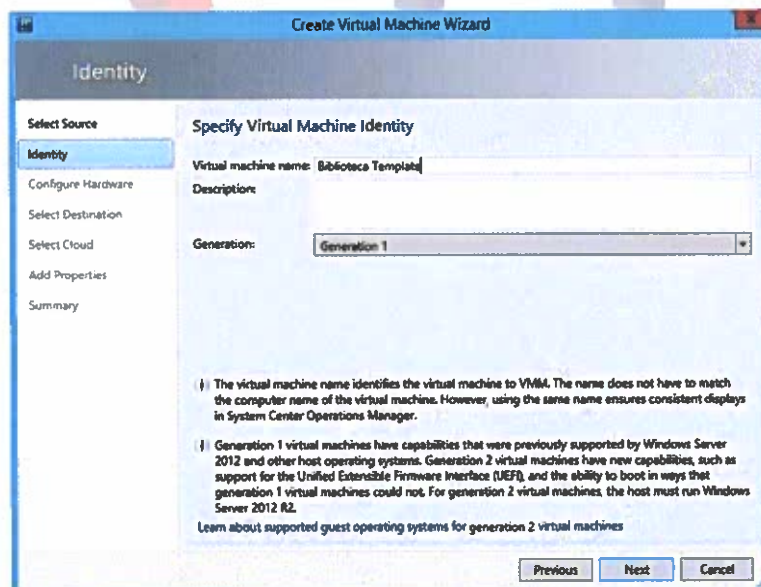


Figura 189: Criar Máquina Virtual para *Template 3*, Fonte Própria

No passo seguinte, vamos selecionar o *hypervisor* que vamos utilizar para criar esta máquina virtual, selecionamos então a opção “Hyper-V”

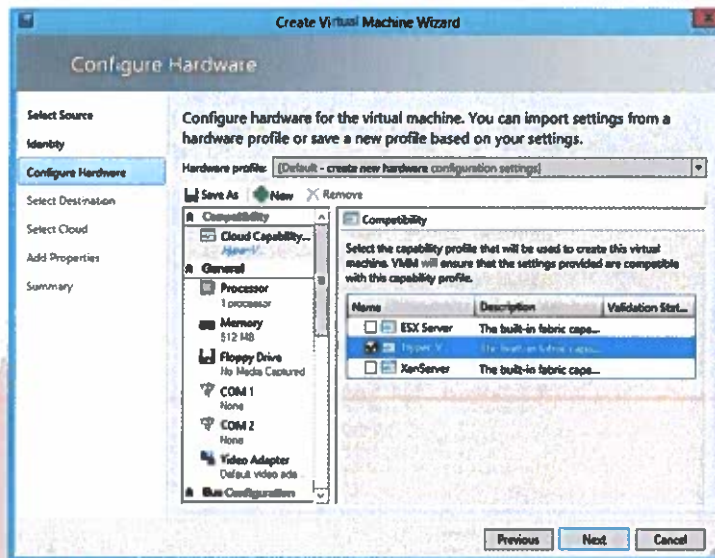


Figura 190: Criar Máquina Virtual para *Template 4*, Fonte Própria

Vamos agora definir a memória a atribuir à máquina, por uma questão de economizar recursos no *host* e perante o tipo de utilização que esta máquina terá, escolhemos a opção de atribuição memória de forma dinâmica.

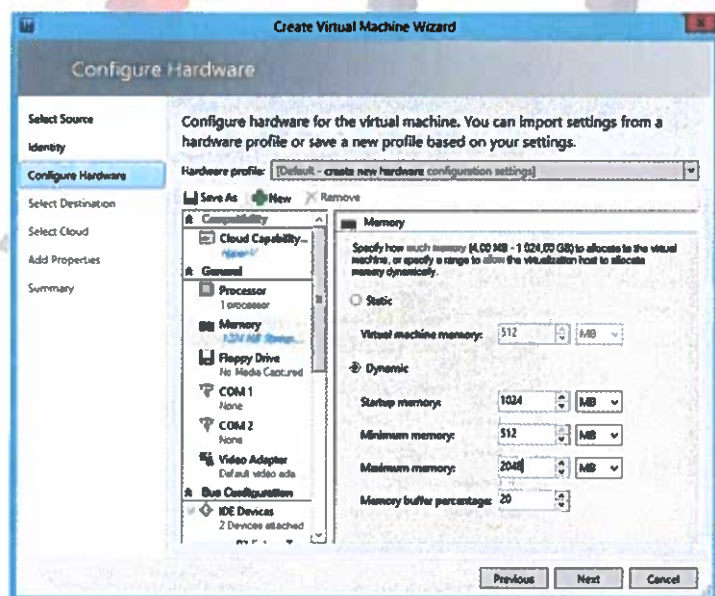


Figura 191: Criar Máquina Virtual para *Template 5*, Fonte Própria

No “Virtual DVD Drive”, iremos selecionar a imagem, que adicionamos anteriormente à biblioteca do SCVMM, do Sistema Operativo Windows 8.1 Enterprise na versão *Evaluation*, que tem um período experimental de 90 dias completamente funcional, sem necessidade de ativação de licença.

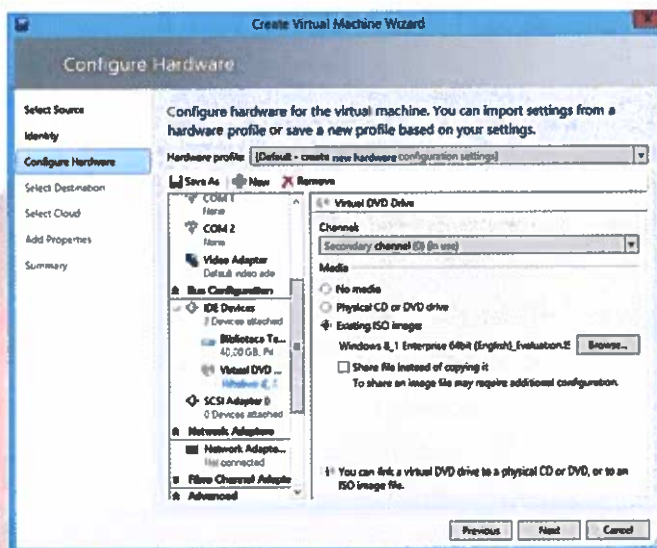


Figura 192: Criar Máquina Virtual para *Template 6*, Fonte Própria

Na parte das interfaces de rede da máquina virtual, já temos por defeito uma interface de rede que vamos associar à rede lógica LAN. Vamos adicionar mais uma interface de rede.

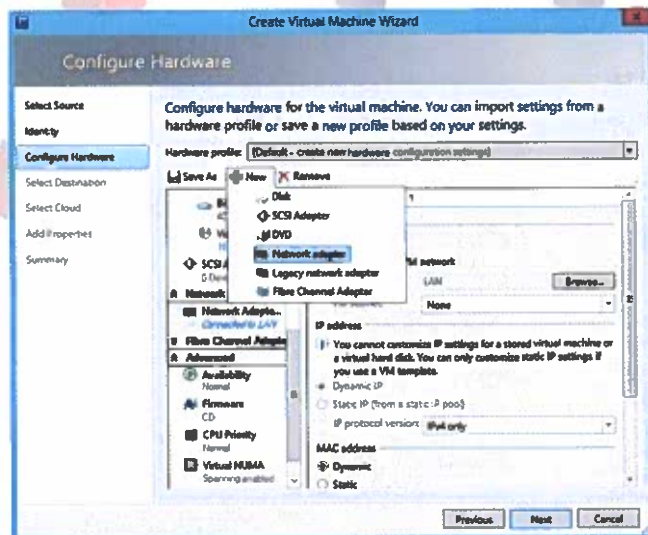


Figura 193: Criar Máquina Virtual para *Template 7*, Fonte Própria

A nova interface de rede que adicionámos, será associada à rede lógica NAT, para que deste modo a máquina tenha acesso à internet de para efetuarmos as atualizações do Windows.

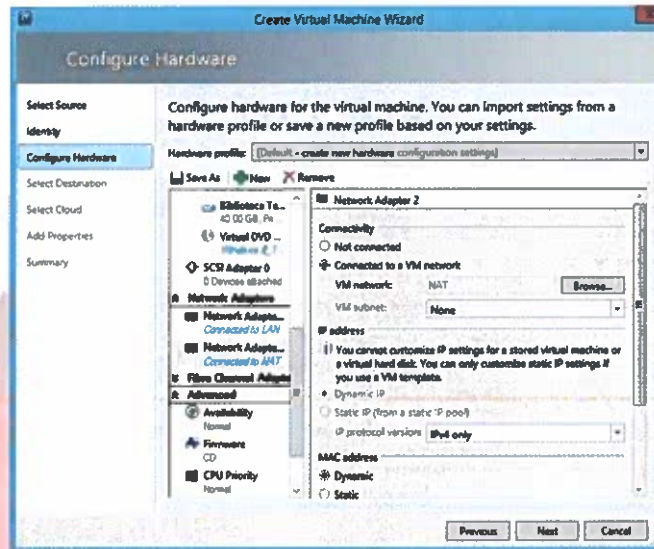


Figura 194: Criar Máquina Virtual para *Template 8*, Fonte Própria

Vamos seleccionar a localização onde a máquina virtual ficará guardada, para o cenário do nosso laboratório vamos escolher a opção “Place the virtual machine on a host”.

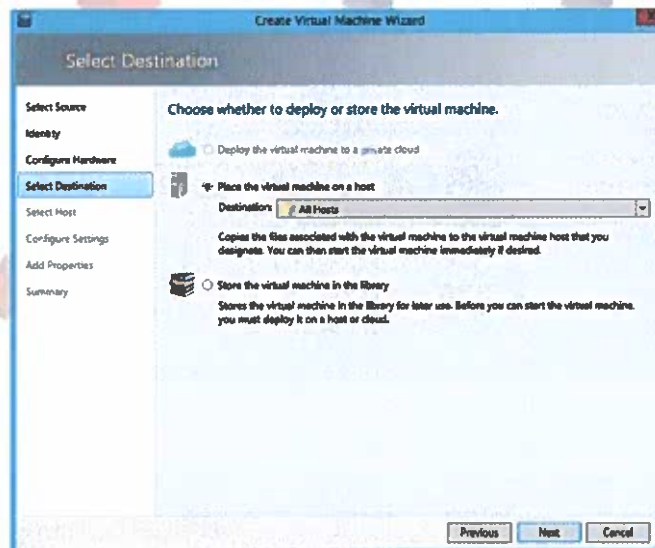


Figura 195: Criar Máquina Virtual para *Template 9*, Fonte Própria

Vamos seleccionar o *host* onde ficará alojada a máquina virtual, no presente laboratório como já referido anteriormente, o *host* será o próprio servidor onde está instalado o SCVMM.

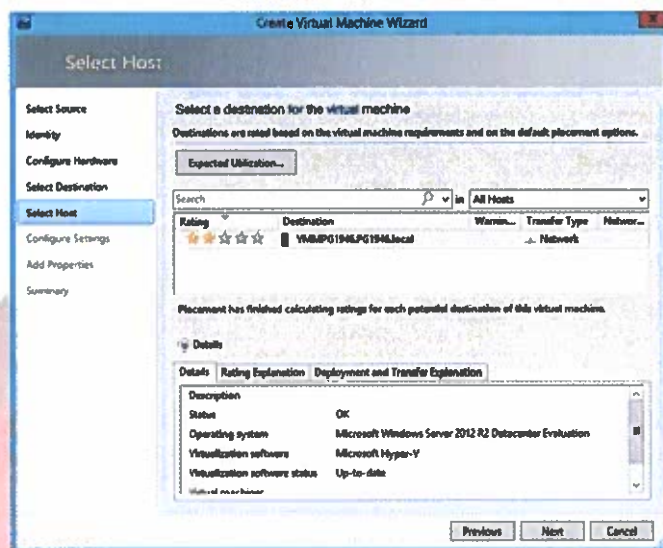


Figura 196: Criar Máquina Virtual para *Template 10*, Fonte Própria

Vamos seleccionar a localização onde ficarão guardados os ficheiros de configuração da nossa máquina virtual.

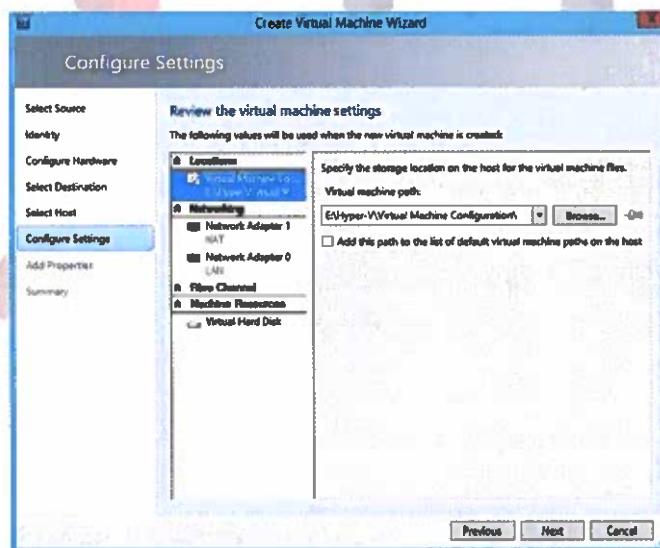


Figura 197: Criar Máquina Virtual para *Template 11*, Fonte Própria

Em seguida selecionamos a localização onde ficarão guardados os discos virtuais da nossa máquina virtual.

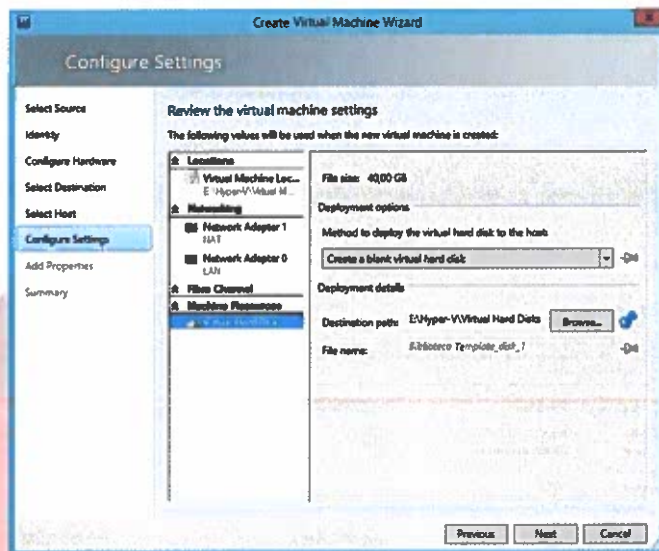


Figura 198: Criar Máquina Virtual para *Template 12*, Fonte Própria

Chegando ao fim do assistente de configuração, podemos ver o quadro com um resumo das configurações e opções selecionadas para a criação da máquina virtual.

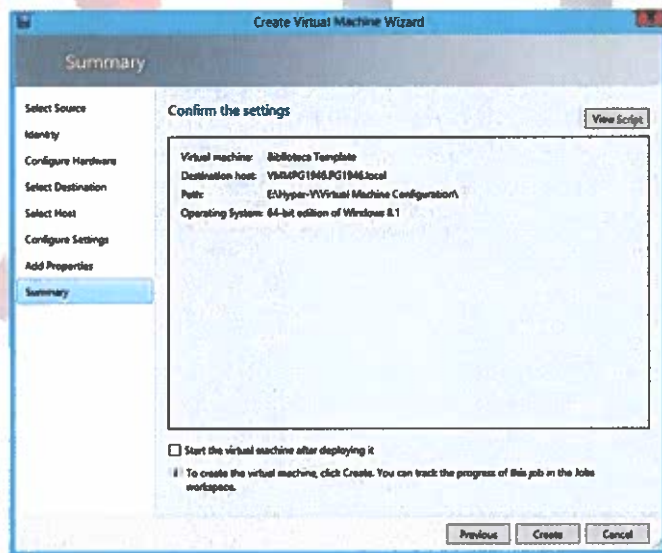


Figura 199: Criar Máquina Virtual para *Template 13*, Fonte Própria

Terminando a criação da máquina virtual, poderemos verificar dentro do nosso *host* a máquina que acabámos de criar.

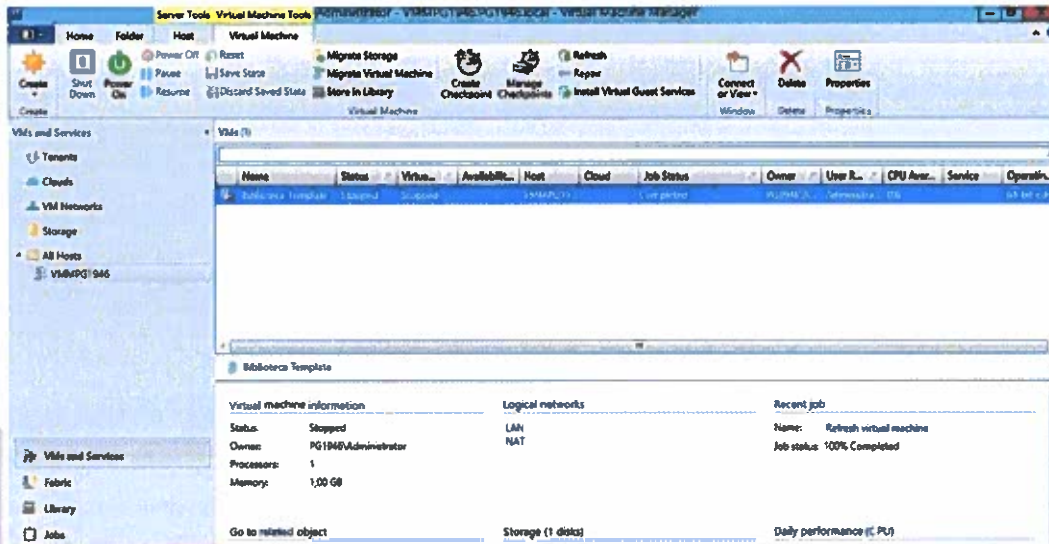


Figura 200: Criar Máquina Virtual para *Template 14*, Fonte Própria

Através do SCVMM, vamos agora ligar a máquina virtual que acabámos de criar.

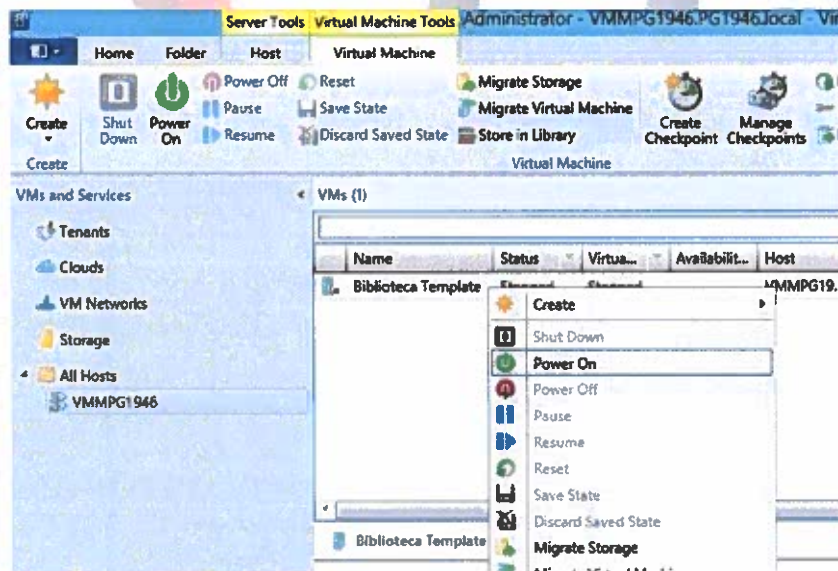


Figura 201: Criar Máquina Virtual para *Template 15*, Fonte Própria

Em seguida, vamos conectar-nos à máquina para que seja feita a instalação do Windows 8.1 Enterprise.

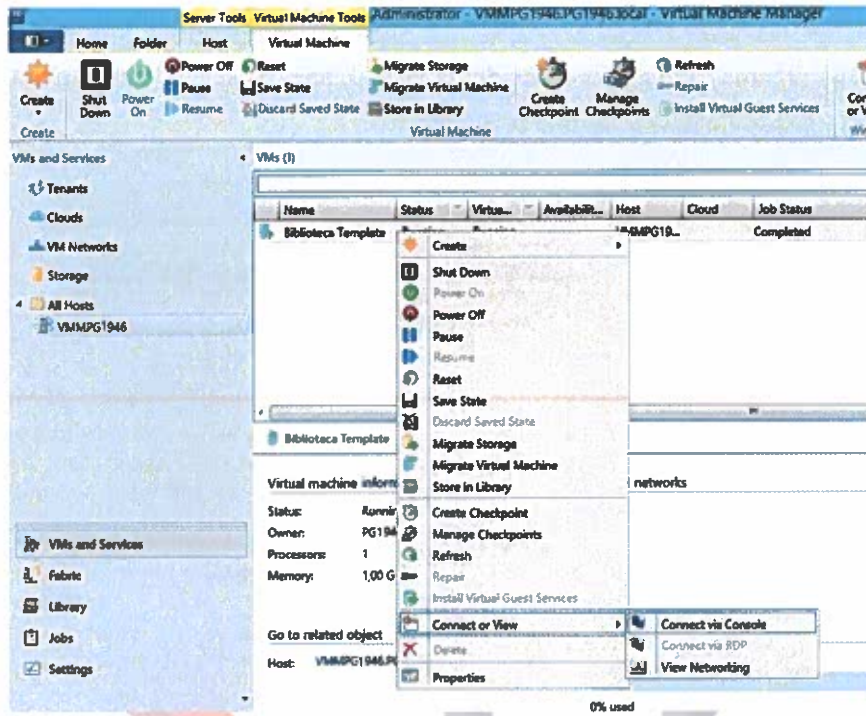


Figura 202: Criar Máquina Virtual para *Template 16*, Fonte Própria

Criar um *Template* para uma Máquina Virtual

Dentro do nosso *host*, como já verificado no passo anterior, temos as nossas máquinas virtuais criadas. Vamos agora criar um *template* a partir da máquina virtual que criámos. Primeiro teremos de efetuar algumas alterações na máquina virtual, para tal seleccionamos a nossa máquina virtual e em seguida escolhemos a opção “Properties”

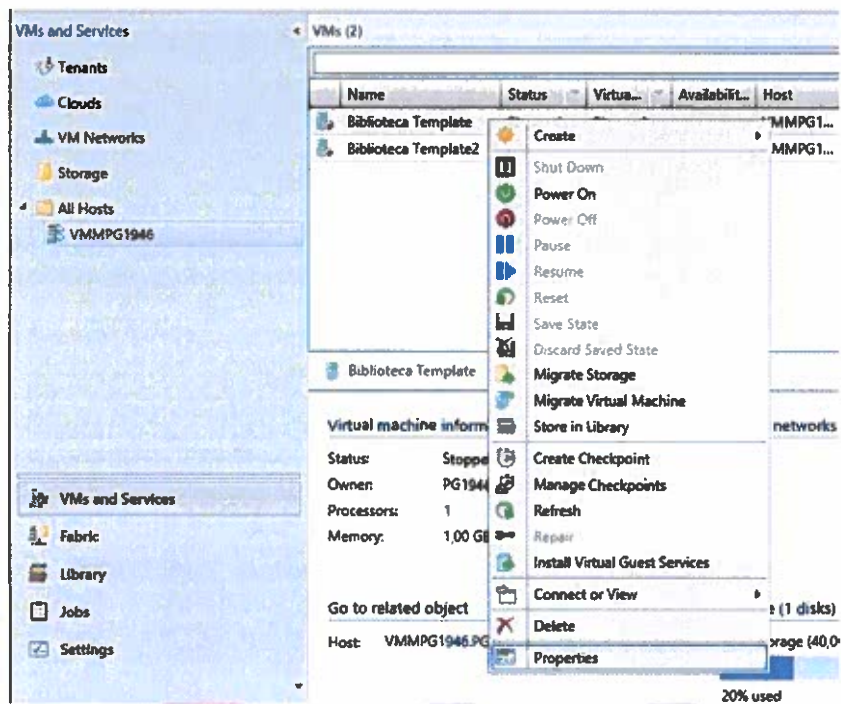


Figura 203: Criar um *Template* para uma Máquina Virtual 1, Fonte Própria

Na criação da nossa máquina virtual, tinha sido inserido na unidade virtual de DVD a imagem do Sistema Operativo para que fosse instalado no arranque da máquina, para a criação de novas máquinas virtuais a partir do *template*, teremos de remover essa imagem da unidade virtual de DVD. Nas propriedades da máquina virtual que criámos, vamos à secção “Hardware Configuration” e no “Virtual DVD Drive” vamos seleccionar a opção “No media”.

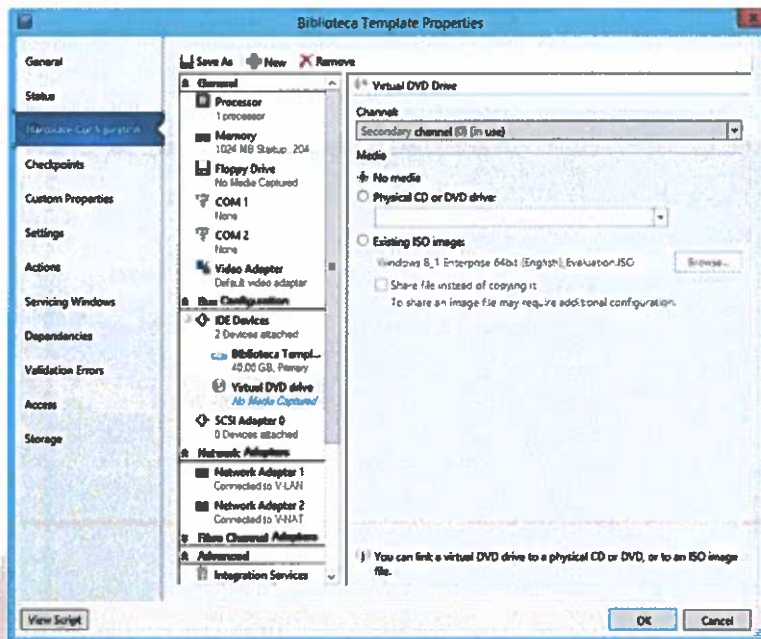


Figura 204: Criar um *Template* para uma Máquina Virtual 2, Fonte Própria

Vamos iniciar agora a criação do *template* a partir da máquina virtual criada. Selecionamos a nossa máquina virtual e sem seguida selecionamos a opção “Create” > “Create VM Template”.

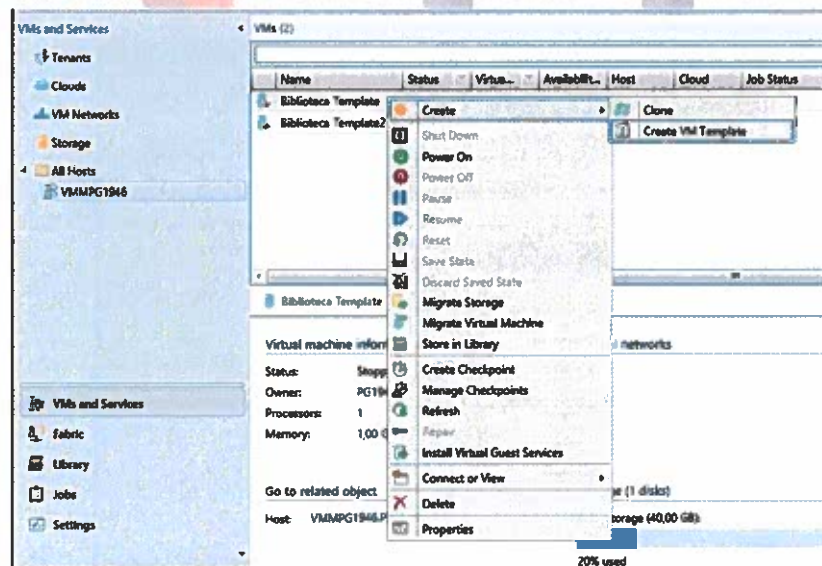


Figura 205: Criar um *Template* para uma Máquina Virtual 3, Fonte Própria

O primeiro passo é darmos o nome de identificação à nossa máquina, neste caso vamos dar o nome “PG1946VM”.

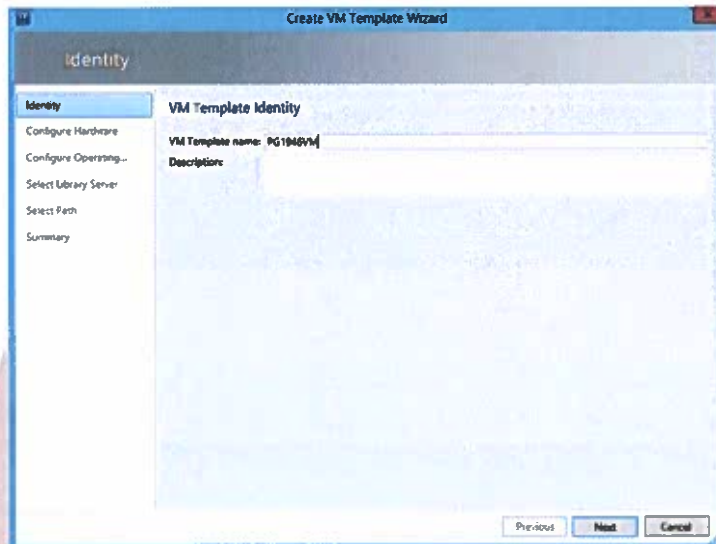


Figura 206: Criar um *Template* para uma Máquina Virtual 4, Fonte Própria

Na secção do lado esquerdo, na opção “Configure Operating System”, seleccionamos a secção “Operating system” e escolhemos a opção “64-bit edition of Windows 8.1”.

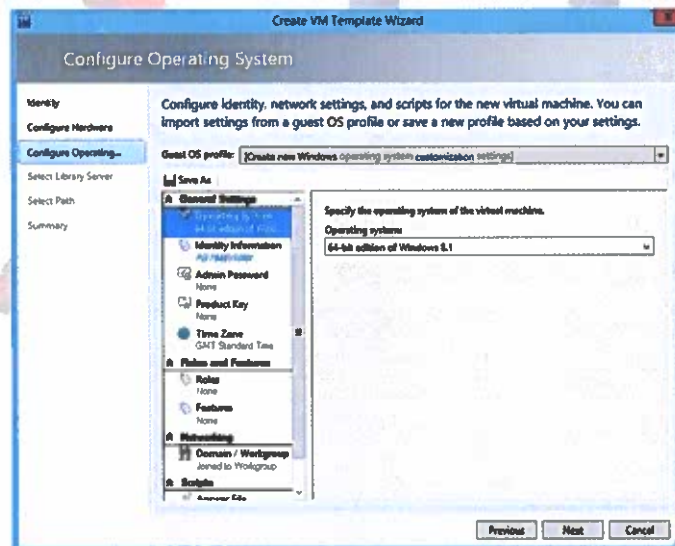


Figura 207: Criar um *Template* para uma Máquina Virtual 5, Fonte Própria

Na secção “Identity Information”, iremos definir o nome que será dado a cada máquina criada a partir deste *template*, de modo a que obtermos um nome em sequência iremos no final do nome colocar “###”, desta forma o nome das máquinas virtuais irá variar a sua terminação entre “01” e “99”. Neste caso daremos o nome “PG1946VM##”.

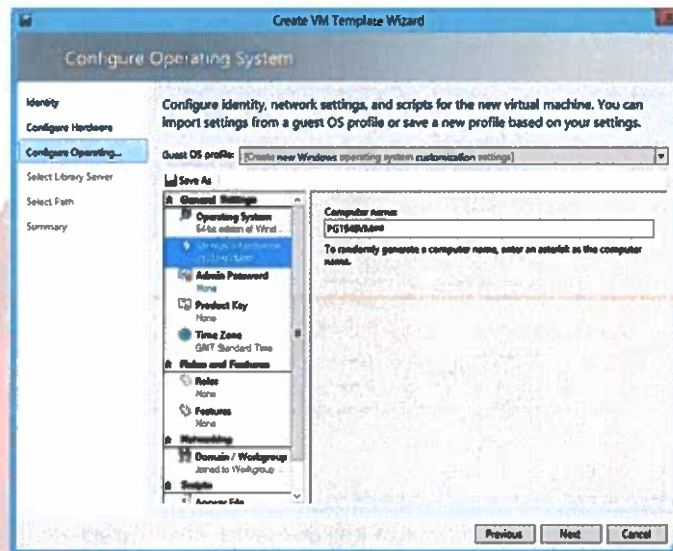


Figura 208: Criar um *Template* para uma Máquina Virtual 6, Fonte Própria

Na secção “Admin Password”, iremos definir as credenciais da conta local de administrador. Vamos escolher a opção “Specify the password of the local administrator account”.

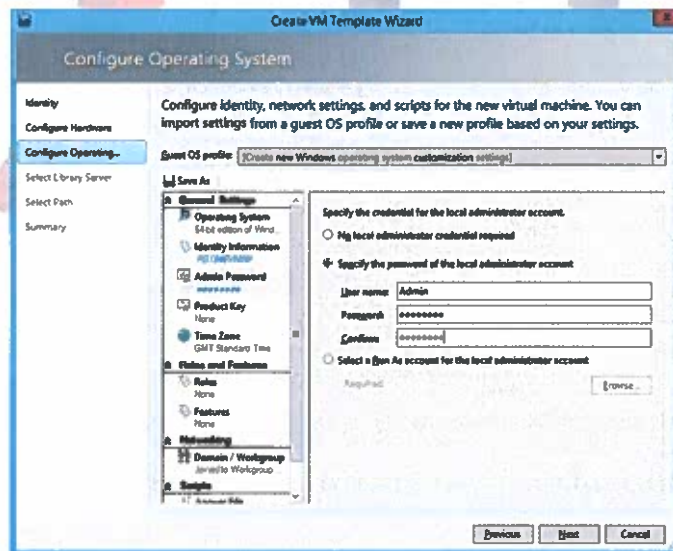


Figura 209: Criar um *Template* para uma Máquina Virtual 7, Fonte Própria

Em seguida, na secção “Domain / Workgroup”, no campo “Domain” vamos inserir o nome do nosso domínio “PG1946.local” e no campo “Specify credentials to use for joining the domain” iremos colocar as credenciais do Administrador do domínio.

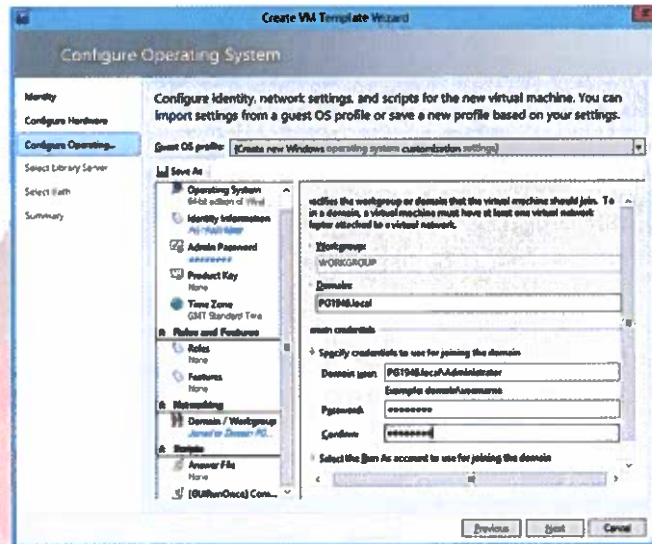


Figura 210: Criar um *Template* para uma Máquina Virtual 8, Fonte Própria

No passo que se segue, vamos seleccionar o servidor onde está a nossa biblioteca.

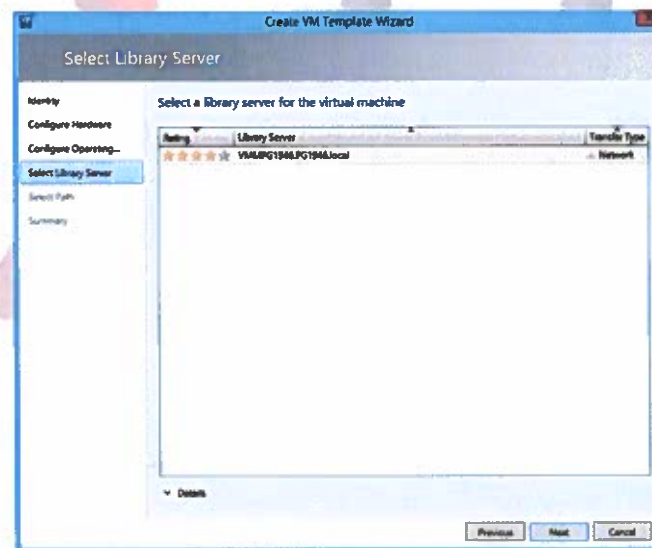


Figura 211: Criar um *Template* para uma Máquina Virtual 9, Fonte Própria

Em seguida, vamos definir a localização onde ficará guardo o *template* que estamos a criar, vamos selecionar a localização da pasta que adicionámos anteriormente à biblioteca do SCVMM, como o nome “Biblioteca Template”.

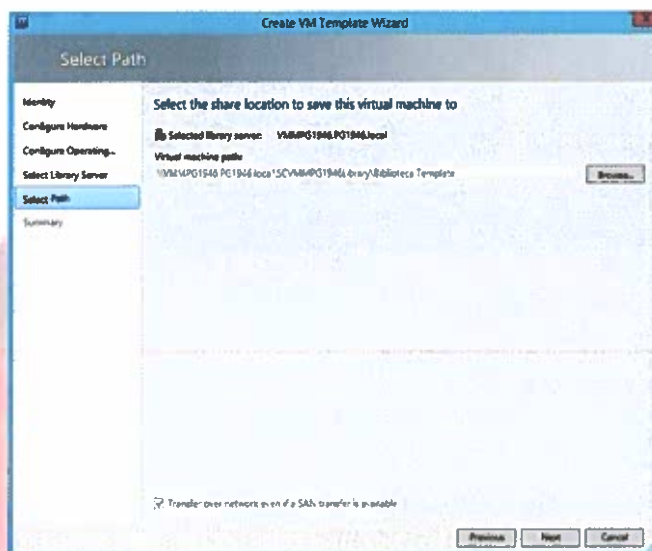


Figura 212: Criar um *Template* para uma Máquina Virtual 10, Fonte Própria

No final do assistente de configuração, teremos um quadro com o resumo das configurações e opções selecionadas para a criação do *template*.

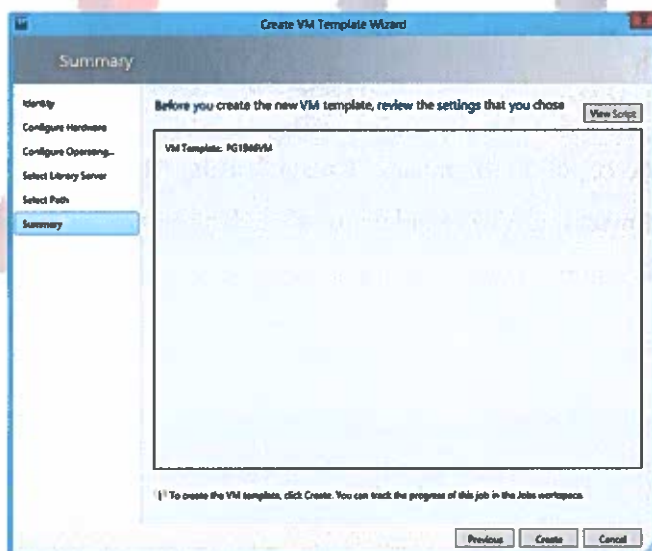


Figura 213: Criar um *Template* para uma Máquina Virtual 11, Fonte Própria

Criar Máquina Virtual a partir de um *Template*

Dentro da secção “VMs and Services”, na *ribbon* carregamos no botão “Create Virtual Machine” e selecionamos a opção “Create Virtual Machine”.

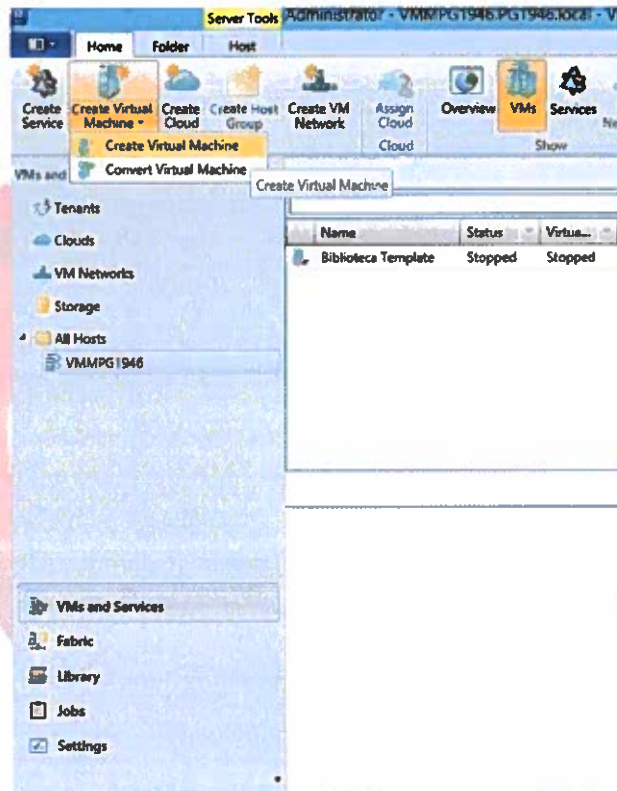


Figura 214: Criar Máquina Virtual a partir de um *Template* 1, Fonte Própria

Iniciado o assistente de configuração “Create Virtual Machine”, selecionamos a opção “Use na existing virtual machine, VM template, or virtual hard disk”. Para adicionarmos o nosso *template* criado no passo anterior como aparece na imagem, vamos carregar no botão “Browse”.

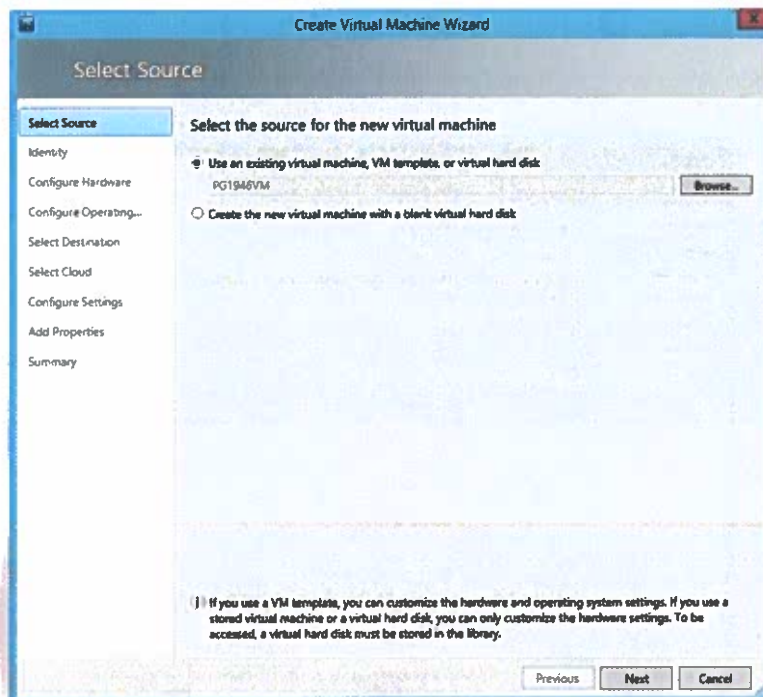


Figura 215: Criar Máquina Virtual a partir de um *Template 2*, Fonte Própria

Selecionamos o *template* criado no passo anterior e carregamos no botão “OK”. Desta forma fica adicionado o nosso *template*.

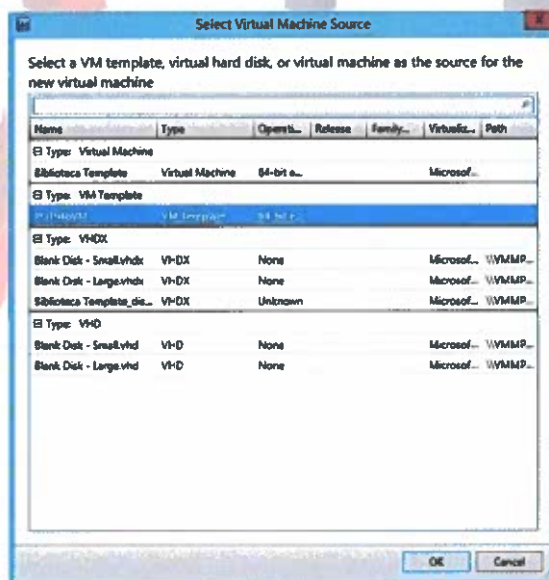


Figura 216: Criar Máquina Virtual a partir de um *Template 3*, Fonte Própria

Na secção “Identity”, não vamos preencher o nome da máquina virtual, pois como foi definido na criação do *template* este nome será atribuído automaticamente.

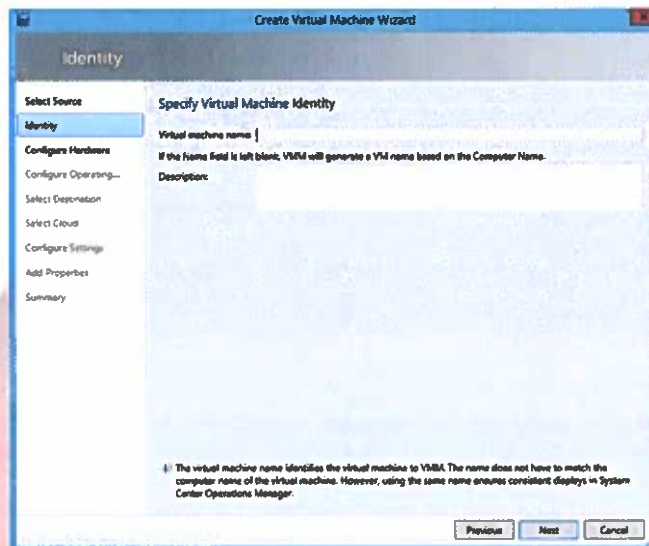


Figura 217: Criar Máquina Virtual a partir de um *Template* 4, Fonte Própria

Na secção “Configure Hardware”, vamos deixar os valores padrão, pois já foram definidos na criação do *template*.

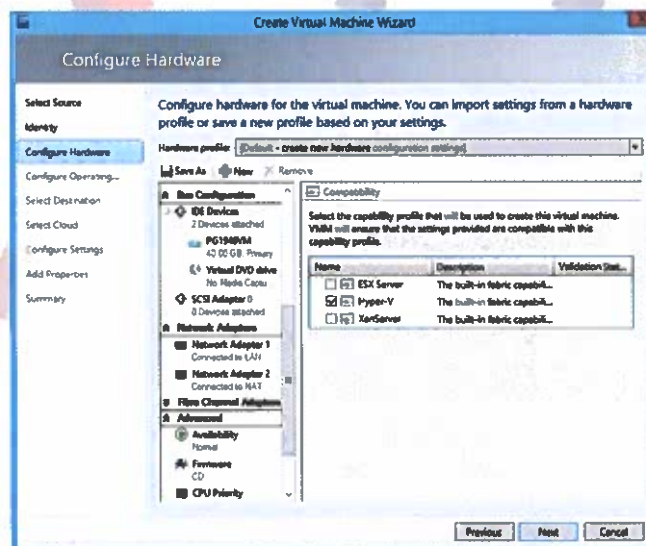


Figura 218: Criar Máquina Virtual a partir de um *Template* 5, Fonte Própria

Na secção seguinte, podemos verificar que foi atribuído o nome à nossa máquina de forma automática, tendo sido a primeira a ser criada, ficou com o nome “PG1946VM01”.

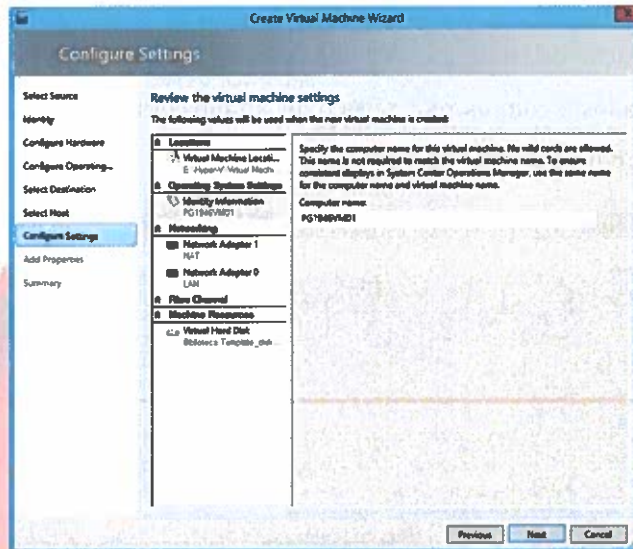


Figura 219: Criar Máquina Virtual a partir de um *Template 6*, Fonte Própria

Antes de ser criada a máquina virtual, temos a secção “Summary” na qual poderemos verificar o resumo das configurações e opções seleccionadas para a criação desta máquina virtual.

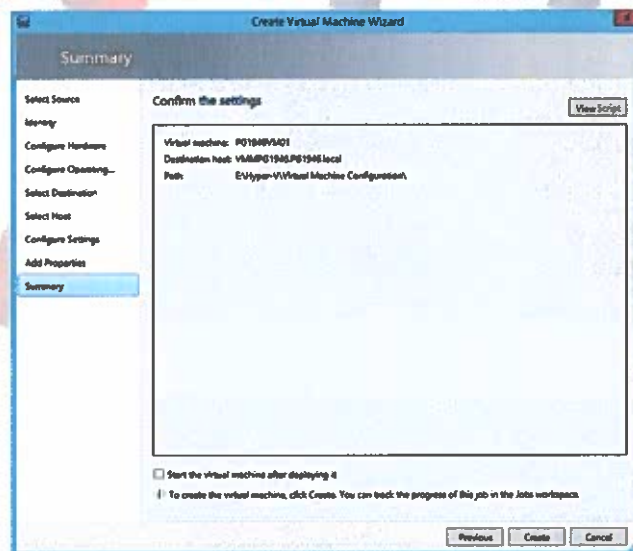


Figura 220: Criar Máquina Virtual a partir de um *Template 7*, Fonte Própria

Após criada a máquina virtual a partir do *template*, podemos inicia-la, tal como já explicado num dos passos anteriores. Podemos ver o resultado das *Group Policy Objects* configuradas numa das secções anteriores, o *desktop* da máquina cliente criada com o fundo personalizado conforme o perfil de acesso à aplicação da Biblioteca Virtual, bem como o *Addobe Reader* e a aplicação da Biblioteca Virtual instalados e com os seus respetivos atalhos criados no *desktop*, temos também o atalho no *desktop* criado para o OneDrive.

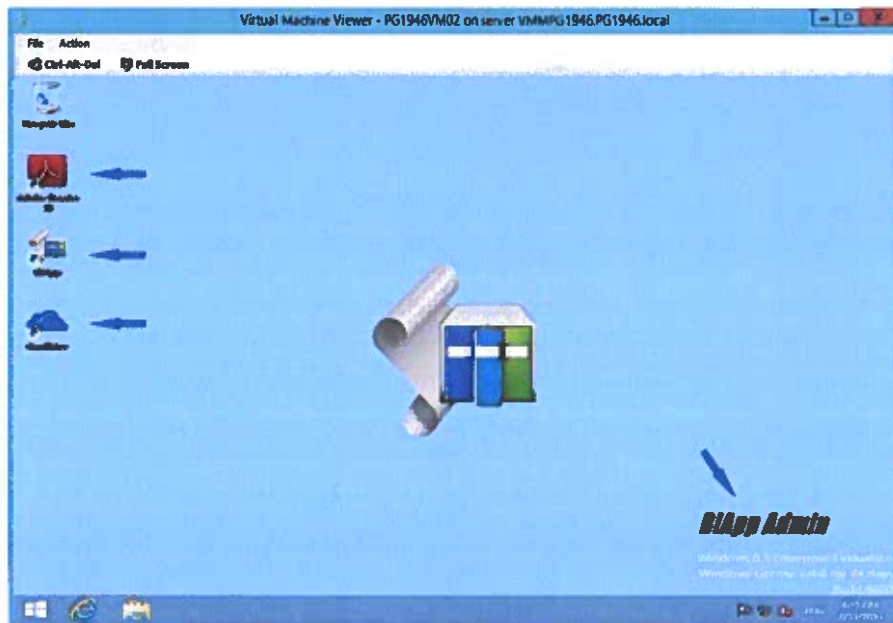


Figura 221: *Desktop* da Máquina Cliente criada no SCVMM, Fonte Própria

4.3 Aplicação Biblioteca Virtual

A Aplicação da Biblioteca Virtual que criámos foi desenvolvida em C# e com interface gráfico desenvolvido em WPF.

A aplicação irá comunicar com uma base de dados, que estará alojada no SQL Server 2012, dentro da instância que criámos com o nome “MSSQLPG1946”, no servidor SQLPG1946.

As comunicações entre a aplicação e a base de dados irão ser realizadas através de objetos ADO.NET.

4.3.1 Criar a Base de Dados da Aplicação

Vamos dar início à criação da base de dados da Aplicação da Biblioteca Virtual. Para tal vamos seleccionar a pasta com o nome “Databases”, dentro da instancia “MSSQLPG1946”, que criámos na instalação do SQL Server 2012, e escolhemos a opção “New Database”.

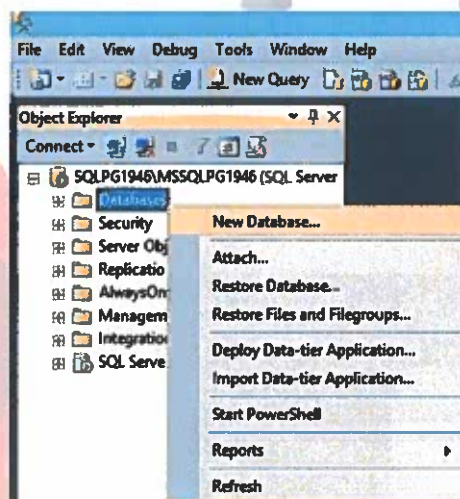


Figura 222: Criar Base de Dados 1, Fonte Própria

No campo “Database name” vamos colocar o nome que iremos dar à base de dados da Aplicação Biblioteca Virtual, neste caso daremos o nome “BiAppBD”.

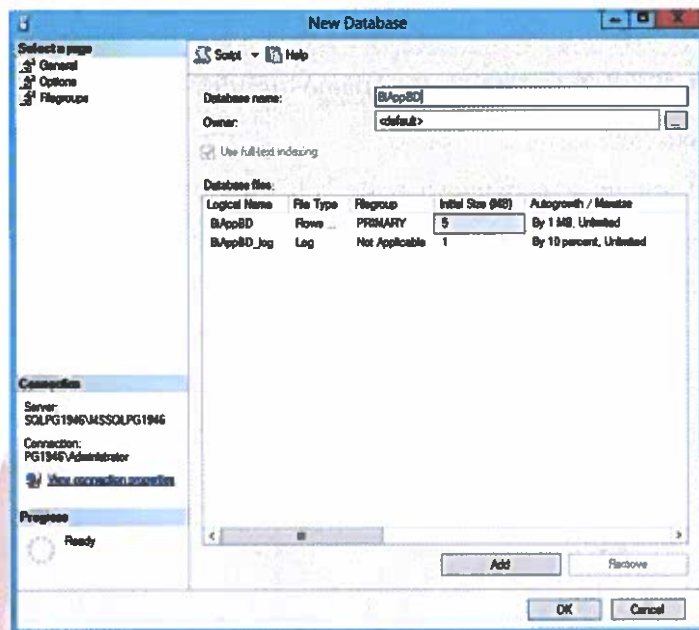


Figura 223: Criar Base de Dados 2, Fonte Própria

Aqui podemos ver a base de dados que acabámos de criar dentro da instância “MSSQLPG1946”, juntamente com a base de dados que criámos através do SCVMM com o nome “SCVMMPG1946BD”.

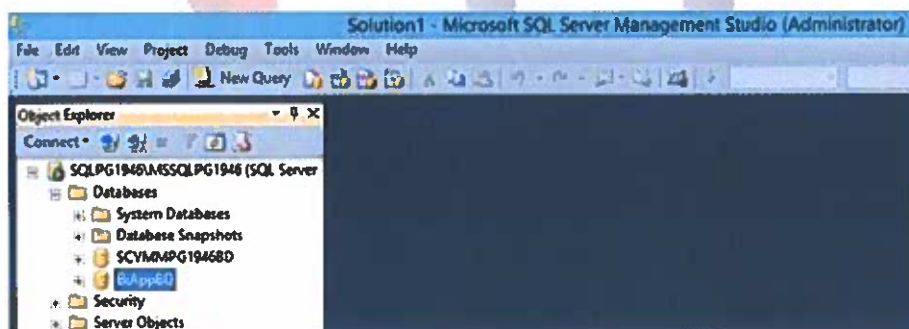


Figura 224: Criar Base de Dados 3, Fonte Própria

4.3.1.1 Criar Tabelas na Base de Dados

Vamos criar as tabelas necessárias para o funcionamento da Aplicação da Biblioteca Virtual, que iremos criar mais à frente.

Dentro da base de dados “BiAppBD”, que criámos no passo anterior, vamos seleccionar a pasta com o nome “Tables” e em seguida escolhemos a opção “New Table”.

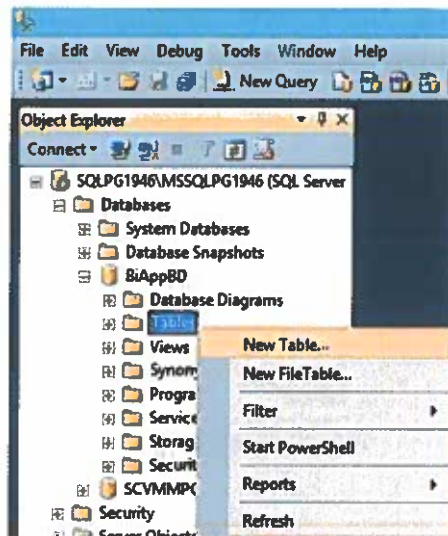


Figura 225: Criar Tabelas na Base de Dados, Fonte Própria

4.3.1.1.1 Criar Tabela para os Documentos

Em primeiro lugar vamos criar uma tabela para os Documentos, à qual iremos dar o nome “Documents”. Esta tabela terá nove colunas em que a chave primária desta tabela será o “Id” e como tal será um campo obrigatório. As colunas com os nomes “AuthorId” e “PublisherId” serão as chaves estrangeiras das tabelas “Author” e “Publisher”, respectivamente, que iremos criar mais à frente, estas serão também campos obrigatórios. Na coluna com o nome “Title” estrá o título do documento e será mais um campo que definimos como obrigatório. Na coluna seguinte, com o nome “CoverBytes”, será onde ficará guardada a capa do documento. Em seguida, temos a coluna com o nome “ContentBytes”, onde ficará guardado o documento em formato PDF. Na coluna “Resume”, irá ficar guardado o resumo do documento. Por último, na coluna “DateDoc” irá ficar a data de publicação do documento.

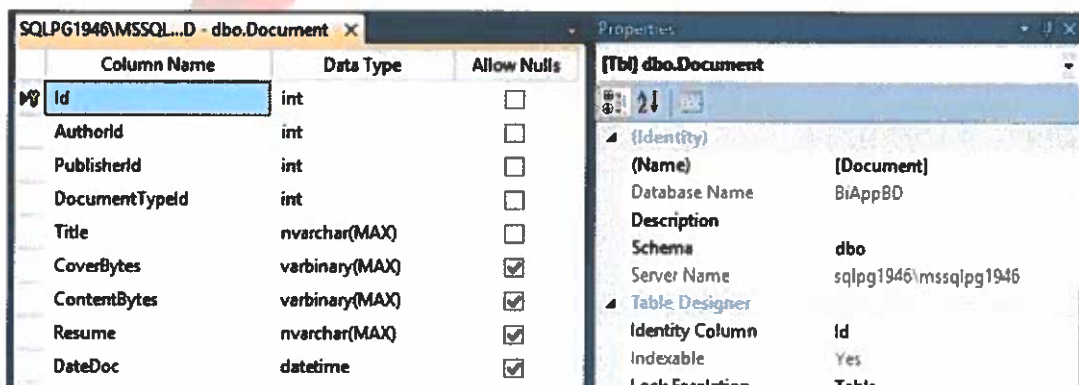


Figura 226: Criar Tabela Documentos, Fonte Própria

No SQL Server, na tabela “Documents”, podemos verificar na pasta “Keys” as chaves estrangeiras que criámos e mencionámos na criação da tabela.

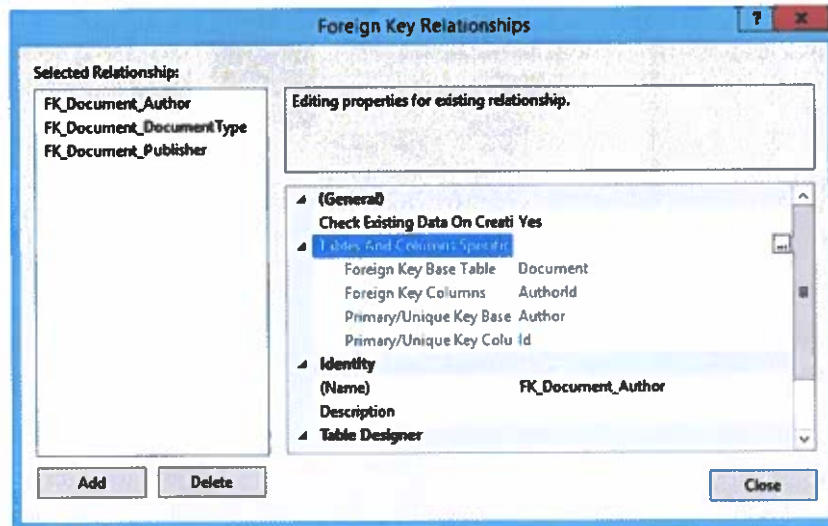


Figura 227: Chaves Estrangeiras, Tabela Documentos, Fonte Própria

4.3.1.1.2 Criar Tabela para as Editoras

A próxima tabela que iremos criar será a tabela para a Editoras, à qual iremos chamar “Publisher”. Esta tabela terá cinco colunas, em que a sua chave primária será a coluna “Id” e por esse motivo será um campo obrigatório. A coluna com o nome “Name”, irá conter o nome da editora e definimos o mesmo como sendo obrigatório. Na coluna com o nome “Address”, irá ficar guardada a morada da editora. Na coluna seguinte, com o nome “Email”, ficará o endereço de correio eletrónico da editora. Por último, na coluna “PhoneNumber”, ficará armazenado o número de telefone da editora.

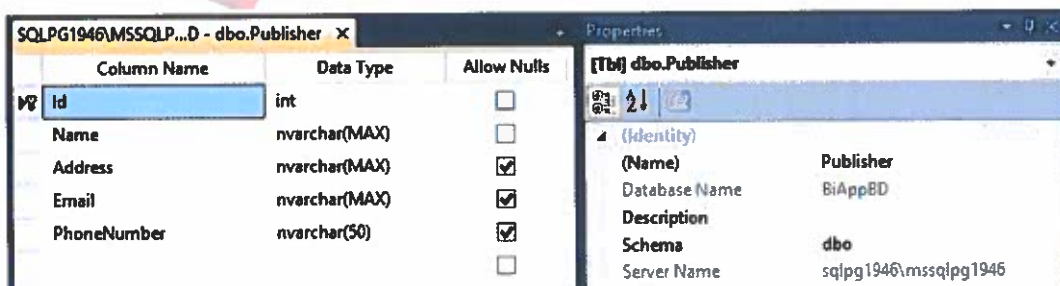


Figura 228: Criar Tabela Editoras, Fonte Própria

4.3.1.1.3 Criar Tabela para os Autores

A tabela seguinte, será a tabela que irá armazenar os dados dos Autores, à qual iremos dar o nome “Authors”. Nesta tabela iremos criar quatro colunas, em que a coluna “Id”, será a sua chave primária e como nas restantes tabelas por esse motivo será um campo obrigatório. A coluna “Name”, irá conter o nome do autor, definimos que este campo não poderá estar vazio. Na coluna “Nationality”, ficará a nacionalidade do autor, em que foi selecionada a opção de não poder conter valores nulos. A última coluna, será a “Birthdate”, onde irá ficar armazenada a data de nascimento do autor.

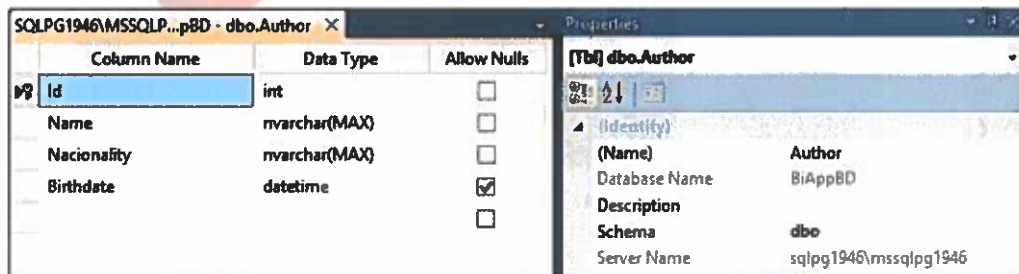


Figura 229: Criar Tabela Autores, Fonte Própria

4.3.1.1.4 Criar Tabela para os Tipos de Documentos

Por último, iremos criar uma tabela com o nome “DocumentType”, que irá guardar a informação com os diversos tipos de documentos. Nesta tabela, teremos apenas duas colunas, em que a sua chave principal está na coluna “Id”, a qual foi definida como sendo um campo obrigatório e a outra coluna terá o nome “Description”, que irá conter a descrição apenas com o tipo de documento, sendo a única outra coluna para além da chave primária, definimos a mesma como não permitindo valores nulos.

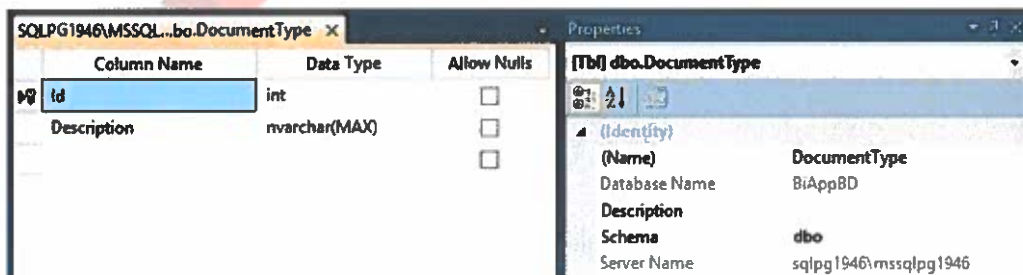


Figura 230: Criar Tabela Tipos de Documentos, Fonte Própria

4.3.1.1.5 Resumo das Tabelas Criadas na Base de Dados

No SQL Server, podemos verificar dentro da base de dados da Aplicação, na pasta “tables”, todas as tabelas que acabámos de criar.

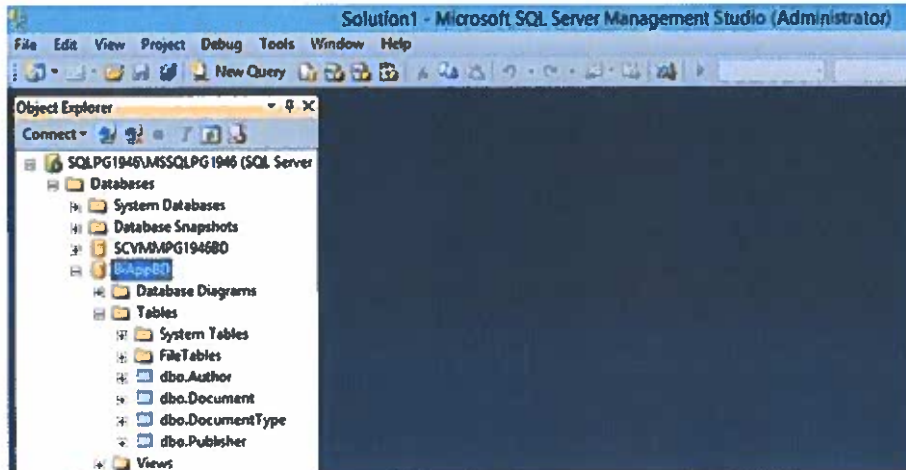


Figura 231: Resumo das Tabelas criadas, Fonte Própria

4.3.1.2 Diagrama e Correlação de Tabelas da Base de Dados

No SQL Server, podemos ver o diagrama da base de dados, em que conseguimos ver a correlação das tabelas e as suas chaves primárias e estrangeiras.

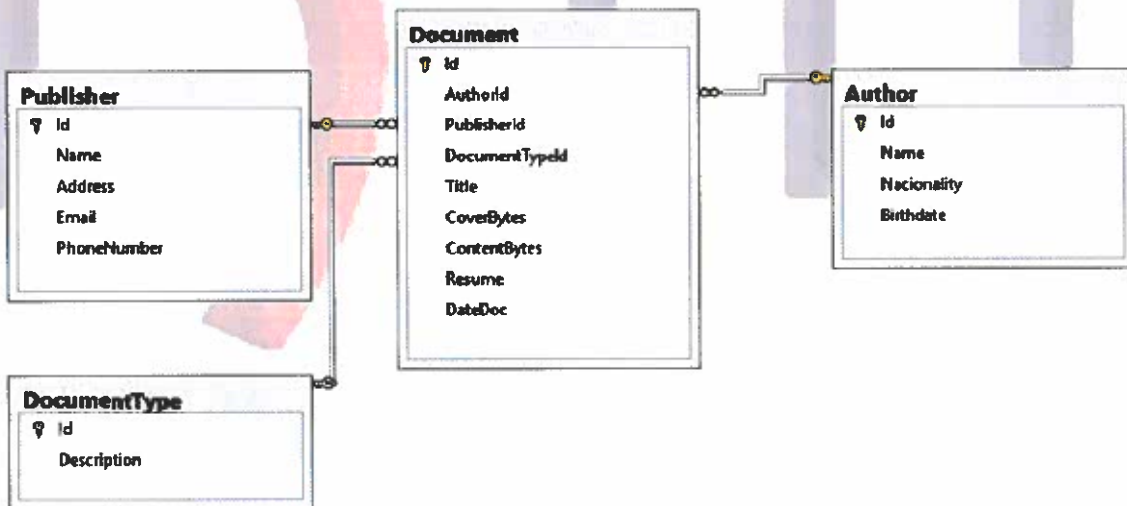
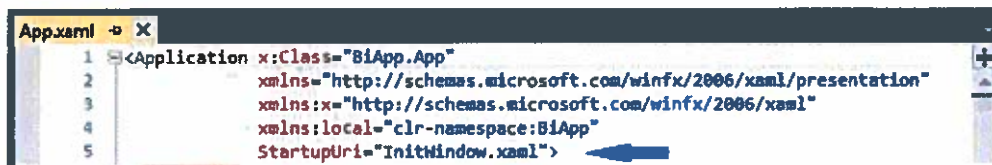


Figura 232: Diagrama das Tabelas da Base de Dados, Fonte Própria

4.3.2 Autenticação na Aplicação através da *Active Directory* (AD DS)

Quando é iniciada a aplicação através do *shortcut* no *desktop* da máquina cliente, o ficheiro que dá início à aplicação é o ficheiro *App.xaml*. Este ficheiro contém os *templates* utilizados para a criação do *layout* das várias interfaces gráficas da aplicação. É também este o ficheiro que indica qual a interface gráfica de arranque da aplicação, neste caso aponta para o ficheiro de *layout* *InitWindow.xaml*.



```
1 <Application x:Class="BiApp.App"
2     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
3     xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4     xmlns:local="clr-namespace:BiApp"
5     StartupUri="InitWindow.xaml">
```

Figura 233: Autenticação na Aplicação através da AD 1, Fonte Própria

A interface gráfica criada pelo ficheiro de *layout* *InitWindow.xaml*, fornece ao utilizador o estado da autenticação das credenciais na *Active Directory*.

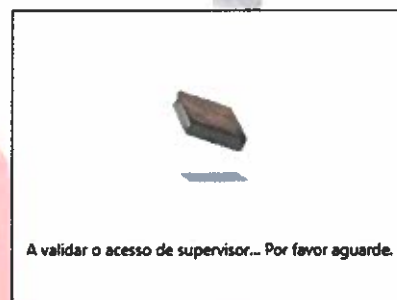
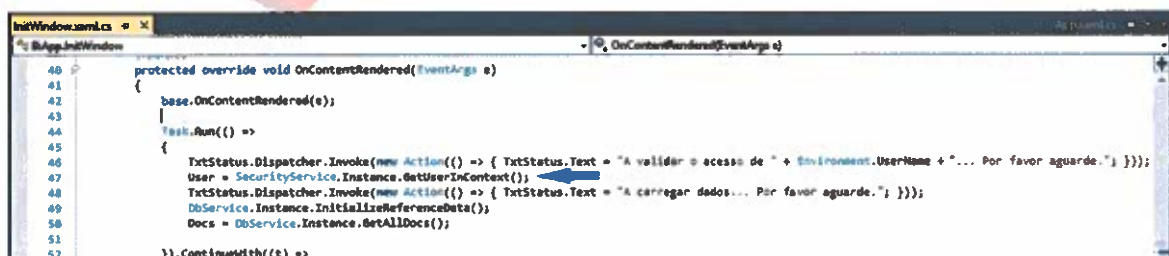


Figura 234: Interface Gráfica Inicial 1, Fonte Própria

Enquanto o utilizador está a visualizar esta interface, em *background* na classe *InitWindow.xaml.cs* está a ser chamada a classe *SecurityService.cs*, que vai validar as credenciais do utilizador.



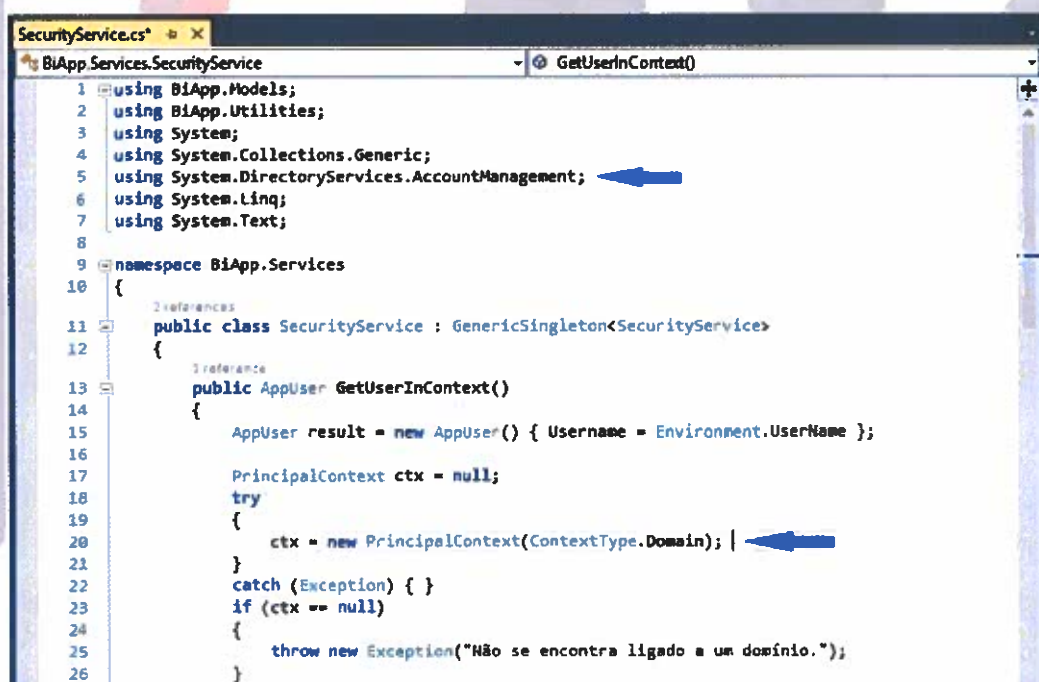
```
40 protected override void OnContentRendered(EventArgs e)
41 {
42     base.OnContentRendered(e);
43     Task.Run(() =>
44     {
45         TxtStatus.Dispatcher.Invoke(new Action(() => { TxtStatus.Text = "A validar o acesso de " + Environment.UserName + "... Por favor aguarde." }));
46         User = SecurityService.Instance.GetUserInContext();
47         TxtStatus.Dispatcher.Invoke(new Action(() => { TxtStatus.Text = "A carregar dados... Por favor aguarde." }));
48         DbService.Instance.InitializeReferenceData();
49         Docs = DbService.Instance.GetAllDocs();
50     });
51     ContinuumWith(() =>
```

Figura 235: Autenticação na Aplicação através da AD 2, Fonte Própria

Como referido anteriormente a classe SecurityService.cs faz a validação do acesso do utilizador à aplicação. Esta validação é feita com base nas credenciais com as quais o utilizador fez o login na máquina cliente e através da qual está a tentar aceder à aplicação.

Para conseguirmos validar o acesso do utilizador, através das credenciais com que fez o login na máquina cliente, recorreremos ao *namespace* “System.DirectoryServices.AccountManagement”, que nos permite o acesso e manipulação de objetos de utilizadores, computadores e grupos de segurança na *Active Directory* (AD DS).

A validação das credenciais do utilizador para acesso à aplicação serão feitas por três fases, a primeira é a de verificar se consegue chegar ao domínio, onde a máquina de onde está a ser feito o acesso à aplicação está ligada, pois poderá não estar num domínio, esta primeira validação é feita através do objeto “PrincipalContext”. (Microsoft Corporation, n.d.-h)



```
SecurityService.cs
BIApp.Services.SecurityService
GetUserInContext()
1 using BIApp.Models;
2 using BIApp.Utilities;
3 using System;
4 using System.Collections.Generic;
5 using System.DirectoryServices.AccountManagement;
6 using System.Linq;
7 using System.Text;
8
9 namespace BIApp.Services
10 {
11     public class SecurityService : GenericSingleton<SecurityService>
12     {
13         public AppUser GetUserInContext()
14         {
15             AppUser result = new AppUser() { Username = Environment.UserName };
16
17             PrincipalContext ctx = null;
18             try
19             {
20                 ctx = new PrincipalContext(ContextType.Domain);
21             }
22             catch (Exception) { }
23             if (ctx == null)
24             {
25                 throw new Exception("Não se encontra ligado a um domínio.");
26             }
27         }
28     }
29 }
```

Figura 236: Autenticação na Aplicação através da AD 3, Fonte Própria

A segunda fase de validação, passa pela verificação do nome de utilizador com o qual foi feito o login na máquina cliente, vai verificar se o mesmo está no domínio, esta validação é realizada através do objeto “UserPrincipal”. (Microsoft Corporation, n.d.-h)

```

SecurityService.cs
BiApp.Services.SecurityService
- GetUserInfoContext()
39     using (ctx)
40     {
41         result.DomainName = ctx.ConnectedServer;
42         |
43         UserPrincipal user = UserPrincipal.FindByIdentity(ctx, result.Username);
44     }

```

Figura 237: Autenticação na Aplicação através da AD 4, Fonte Própria

A última fase de validação é a de verificação se os grupos de segurança definidos para terem acesso à aplicação existem na *Active Directory*, esta validação é feita através do objeto *GroupContext*. (Microsoft Corporation, n.d.-h)

Após a validação da existência dos grupos de segurança na *Active Directory*, é feita a validação de qual o perfil a que pertencem, se pertencem ao grupo de administração da aplicação, ou ao grupo de utilizadores da aplicação.

```

SecurityService.cs
BiApp.Services.SecurityService
- GetUserInfoContext()
46     GroupPrincipal adminGroup = GroupPrincipal.FindByIdentity(ctx, ConfigHelper.Instance.GroupNameAdmin);
47     if (adminGroup == null)
48     {
49         throw new Exception("Não foi encontrado o grupo de acesso " + ConfigHelper.Instance.GroupNameAdmin);
50     }
51     GroupPrincipal normalGroup = GroupPrincipal.FindByIdentity(ctx, ConfigHelper.Instance.GroupNameNormal);
52     if (normalGroup == null)
53     {
54         throw new Exception("Não foi encontrado o grupo de acesso " + ConfigHelper.Instance.GroupNameNormal);
55     }
56     if (user != null)
57     {
58         if (user.IsMemberOf(adminGroup))
59         {
60             result.CurrentAccessMode = AccessMode.Admin;
61         }
62         else if (user.IsMemberOf(normalGroup))
63         {
64             result.CurrentAccessMode = AccessMode.Normal;
65         }
66     }
67 }

```

Figura 238: Autenticação na Aplicação através da AD 5, Fonte Própria

De modo a saber quais os grupos de segurança definidos para cada tipo de acesso à aplicação, foi criada a classe *ConfigHelper.cs*, onde com o método “get” vai ao *App.config*, através do *AppSettings* para obter os grupos de segurança que foram definidos no *App.config*.

Estes grupos de acesso foram criados para dar privilégios diferentes aos utilizadores e aos administradores da aplicação, tal como o *layout* da aplicação que é apresentado a cada um deles, é necessário que a aplicação mediante o tipo de acesso crie o *layout* correto e diferenciado.

```

8 namespace BiApp.Utilities
9 {
10     public class ConfigHelper : GenericSingleton<ConfigHelper>
11     {
12         private const string CipherKey = "PG1946";
13         public string GroupNameAdmin
14         {
15             get
16             {
17                 string encrypted = ConfigurationManager.AppSettings["GroupNameAdmin"] ?? string.Empty;
18                 string decrypted = StringCipher.Decrypt(encrypted, CipherKey);
19                 return decrypted;
20             }
21         }
22
23         public string GroupNameNormal
24         {
25             get
26             {
27                 string encrypted = ConfigurationManager.AppSettings["GroupNameNormal"] ?? string.Empty;
28                 string decrypted = StringCipher.Decrypt(encrypted, CipherKey);
29                 return decrypted;
30             }
31         }
32     }
33 }

```

Figura 239: Autenticação na Aplicação através da AD 6, Fonte Própria

Os nomes dos grupos de segurança que terão acesso à aplicação, foram previamente inseridos no código no AppSettings do App.config. O grupo de acesso com privilégios de utilizador é o “GroupNameNormal” e foi definido que o grupo de segurança na *Active Directory* para este perfil é o já criado anteriormente para o efeito, “AppUser”. Para os acessos com privilégios de administração na aplicação é o “GroupNameAdmin” e o grupo de segurança na *Active Directory* que definimos para este perfil foi o “AppAdmin”. Como se pode ver na imagem abaixo o nome dos grupos de segurança da *Active Directory* que cada um dos grupos da aplicação está a receber está encriptado por questões de segurança, iremos falar deste processo de encriptação e desencriptação mais à frente.

```

11 <appSettings>
12 <add key="GroupNameAdmin" value="60d00f429a8130f3e383c7b0c4e4e238af70f5a682549557d6pY3jy1M1e6f7WAZcdMlOn1s4Y14dQq;0b0PLn1g02yeF02r8cz4Kv0fAYs0fYoyd0B0bL" />
13 <add key="GroupNameNormal" value="7a67lr370pLIVest0z008t0d0e%Cl.7009CZ/nat5u000u0d0e000F0u000ch00Lg3u000r00r365Kg1s5u11Fu0000P0C0/00007g0U92R26v0b0d0p0v0H000r0" />
14 </appSettings>

```

Figura 240: Autenticação na Aplicação através da AD 7, Fonte Própria

Na classe SecurityService.cs foi criada uma exceção, para que no caso de algum dos grupos de segurança definidos no App.config não exista na *Active Directory*, é devolvida uma mensagem de erro ao utilizador a indicar que o grupo de segurança não foi encontrado.

```
SecurityService.cs
BiApp.Services.SecurityService
GetUserInContext()
47 GroupPrincipal adminGroup = GroupPrincipal.FindByIdentity(ctx, ConfigHelper.Instance.GroupNameAdmin);
48 if (adminGroup == null)
49     throw new Exception("Não foi encontrado o grupo de acesso " + ConfigHelper.Instance.GroupNameAdmin);
50 GroupPrincipal normalGroup = GroupPrincipal.FindByIdentity(ctx, ConfigHelper.Instance.GroupNameNormal);
51 if (normalGroup == null)
52     throw new Exception("Não foi encontrado o grupo de acesso " + ConfigHelper.Instance.GroupNameNormal);
```

Figura 241: Autenticação na Aplicação através da AD 8, Fonte Própria

Passando todas as validações anteriores é retomada a classe InitWindow.xaml.cs e após a aplicação tentar estabelecer a ligação à base de dados, é validado se o tipo de acesso é “Unknown”, caso seja devolve uma mensagem de erro ao utilizador a indicar que as suas credenciais não pertencem a nenhum dos grupos de segurança com acesso à aplicação. Em caso de não ser “Unknown” dá início à interface gráfica principal da aplicação “MainWindow” e fecha esta interface “InitWindow”.

```
InitWindow.xaml.cs
BiApp.InitWindow
OnCurrentWindowLoaded@vstf1p.d
67 else
68 {
69     if (User.CurrentAccessMode == AccessMode.Unknown)
70     {
71         MessageBox.Show("O seu utilizador não pertence a nenhum grupo (ou acesso à aplicação ", "Acesso negado", MessageBoxButton.OK, MessageBoxImage.Error);
72         this.Close();
73     }
74     MainWindow main = new MainWindow(User, Docs, DbService.Instance.Publishers, DbService.Instance.Authors, DbService.Instance.DocumentTypes);
75     main.Show();
76     this.Close();
77 }
```

Figura 242: Autenticação na Aplicação através da AD 9, Fonte Própria

Para diferenciar os diferentes tipos de acesso à aplicação foi criada a classe Enums.cs e criada uma lista de constantes com o nome o “AccessMode”, para diferenciar os modos de acesso entre “Admin”, “Normal” e “Unknown”.

```
Enums.cs
BiApp.Models.EditionMode
Edit
References
8 public enum AccessMode
9 {
10     Unknown,
11     Normal,
12     Admin
13 }
```

Figura 243: Autenticação na Aplicação através da AD 10, Fonte Própria

4.3.3 Estabelecer Ligação da Aplicação à Base de Dados

Após a validação de acesso das credenciais de ligação à aplicação através da *Active Directory*, na interface gráfica do ficheiro de *layout* *InitWindow.xaml* o utilizador tem a indicação de que os dados estão a ser carregados.

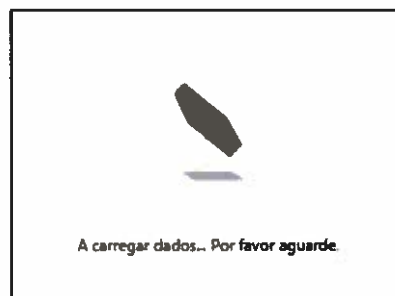


Figura 244: Interface Gráfica Inicial 2, Fonte Própria

Em seguida enquanto o utilizador continua a visualizar a interface gráfica do ficheiro de *layout* *InitWindow.xaml* com a nova mensagem, em background será estabelecida a ligação da aplicação à base de dados criada anteriormente, chamando a classe *DbService.cs*.



Figura 245: Estabelecer Ligação com a Base de Dados 1, Fonte Própria

A classe *DbService.cs* irá carregar na memória os valores de referência das tabelas editora, autor e tipos de documento.


```
30 public void InitializeReferenceData()
31 {
32     DataTable dt = DbManager.Instance.GetDataTable("select * from DocumentType");
33     if (dt != null && dt.Rows.Count > 0)
34     {
35         DocumentTypes = new List<DocumentType>();
36         foreach (DataRow r in dt.Rows)
37             DocumentTypes.Add(FillDocumentType(r));
38     }
39     dt = DbManager.Instance.GetDataTable("select * from Author");
40     if (dt != null && dt.Rows.Count > 0)
41     {
42         Authors = new List<Author>();
43         foreach (DataRow r in dt.Rows)
44             Authors.Add(FillAuthor(r));
45     }
46     dt = DbManager.Instance.GetDataTable("select * from Publisher");
47     if (dt != null && dt.Rows.Count > 0)
48     {
49         Publishers = new List<Publisher>();
50         foreach (DataRow r in dt.Rows)
51             Publishers.Add(FillPublisher(r));
52     }
53 }
```

Figura 246: Estabelecer Ligação com a Base de Dados 2, Fonte Própria

Na classe DbService.cs é chamada a classe DbManager.cs que cria dois novos objetos ADO.NET, uma “SqlConnection” e um “SqlDataAdapter”. O “SqlDataAdapter” é apenas criado após ter sido estabelecida a ligação à base de dados com sucesso, utilizando a “ConnectionString” obtida pelo objeto “SqlConnection”. (Microsoft Corporation, n.d.-f)

```
11 public class DbManager : GenericSingleton<DbManager>
12 {
13     public DataTable GetDataTable(string selectStatement)
14     {
15         DataTable dt = new DataTable();
16
17         using (SqlConnection con = new SqlConnection(ConfigHelper.Instance.MyConnectionString))
18         using (SqlDataAdapter adp = new SqlDataAdapter(selectStatement, con))
19         {
20             adp.Fill(dt);
21         }
22         return dt;
23     }
24 }
```

Figura 247: Estabelecer Ligação com a Base de Dados 3, Fonte Própria

O objeto “SqlConnection” utiliza a classe ConfigHelper.cs para aceder ao App.config e obter a “ConnectionString” definida para estabelecer a ligação à base de dados.

```

50 public string MyConnectionString
51 {
52     get
53     {
54         string encrypted = ConfigurationManager.ConnectionStrings["MyConnectionString"].ConnectionString ?? string.Empty;
55         string decrypted = StringCipher.Decrypt(encrypted, CipherKey);
56         return decrypted;
57     }
58 }

```

Figura 248: Estabelecer Ligação com a Base de Dados 4, Fonte Própria

No App.config podemos verificar a nossa “ConnectionString” para ligação à base de dados da aplicação. Tal como foi acontece nos grupos de segurança referentes a cada grupo de acesso à aplicação, também a “ConnectionString” se encontra encriptada por questões de segurança. Foi utilizada a “ConnectionString” com o conteúdo seguinte: “Data Source=SQLPG1946\MSSQLPG1946;Initial Catalog=BiAppBD;Trusted_Connection=yes”. O “Data Source” é referente ao nome do servidor onde se encontra alojada a base de dados e depois da barra o nome da instância na qual se encontra a base de dados. O “Initial Catalog” identifica o nome da nossa base de dados. Foi escolhido o tipo de autenticação à base de dados com autenticação através da *Active Directory*, para tal colocámos o parâmetro “Trusted Connection” como verdadeiro. (Microsoft Corporation, n.d.-f)

```

17 <connectionStrings>
18 <add name="MyConnectionString" connectionString="1#9D48Q2ubLrd107hoczahVWu00Luhew01q008D716EFP7wfu0m-3/WAdcn311p4M8u2160a3184a1P13c369007yF88v4u8Ca/
19 s1YR8Ab+V81+u0t59Z16ysM8d7dps7sP2Z16u8Lrtk05u1s9yk148A1UB3api1x717m811BFxagt7C8g8wv3P270y03C978v086IyfSKA=" providerName="System.Data.SqlClient" />
20 </connectionStrings>

```

Figura 249: Estabelecer Ligação com a Base de Dados 5, Fonte Própria

4.3.4 Layout das Interfaces Gráficas

4.3.4.1 Interface Gráfica Principal

Nesta secção vamos descrever e explicar a construção e funcionamento da interface gráfica referente aos Documentos. Pelo que iremos explicar a Tab Documentos da interface gráfica principal. Todas as restantes Tabs são idênticas, apenas alterando o nome das colunas que irão ser preenchidas com o conteúdo da base de dados.

Como referido anteriormente após todas as validações de autenticação do utilizador na *Active Directory* e de ter sido estabelecida a ligação à base de dados, a interface gráfica do ficheiro de *layout* InitWindow.xaml é encerrada e é iniciada a interface gráfica do ficheiro de *layout* MainWindow.xaml. A interface gráfica é lançada mediante o tipo de acesso do utilizador, como

podemos verificar na classe MainWindow.xaml.cs, caso o perfil do utilizador não seja de administrador o interface terá alguns componentes ocultos. O perfil de utilizador apenas terá visível a Tab Documentos e o botão “Download”.

```

MainWindow.xaml.cs
BiApp.MainWindow
- MainWindow(AppUser user, List<Document> docs,
129     if(user.CurrentAccessMode != AccessMode.Admin)
130     {
131         BtnAdd.Visibility = Visibility.Collapsed;
132         BtnEdit.Visibility = Visibility.Collapsed;
133         BtnDel.Visibility = Visibility.Collapsed;
134         TabPublisher.Visibility = Visibility.Collapsed;
135         TabAuthors.Visibility = Visibility.Collapsed;
136         TabDocumentTypes.Visibility = Visibility.Collapsed;
137         LblProfile.Text = "Perfil de utilizador: Normal";
138     }
139     else
140     {
141         LblProfile.Text = "Perfil de utilizador: Admin";
142     }
143
144     LblUser.Text = "Nome de utilizador: " + User.Username;
145     this.DataContext = this;
146 }
    
```

Figura 250: Layout Interface Gráfica Principal 1, Fonte Própria

Para o perfil de utilizador o interface gráfico do ficheiro de *layout* MainWindow.xaml terá o seguinte formato.

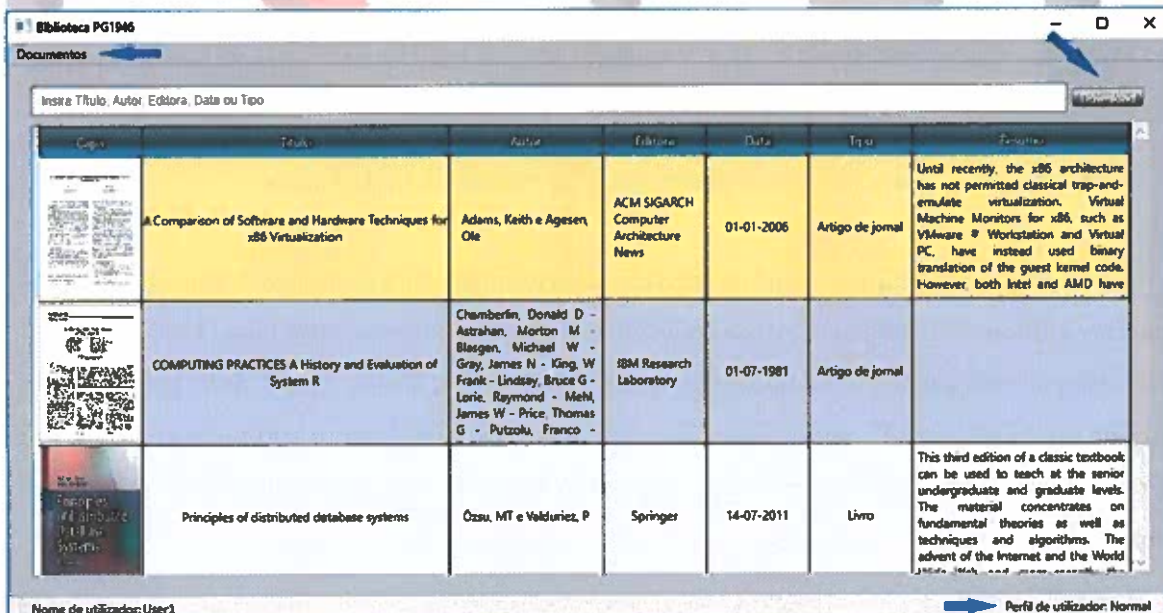


Figura 251: Layout Interface Gráfica Principal 2, Fonte Própria

4.3.4.1.1 Interface Gráfica na Tab Documentos

No caso do perfil do utilizador ser de administrador, o interface gráfico terá todos os componentes visíveis. As restantes Tabs apenas alteram a “DataGrid” onde está contida a tabela que é importada da base de dados. A caixa de texto para pesquisa e os botões no canto superior direito irão manter-se sempre independentemente da Tab que seja selecionada. Deste modo todas as restantes interfaces gráficas Editoras, Autores e Tipos de Documentos, são construídas da mesma forma, pelo que só iremos mostrar o seu *layout* para o administrador.

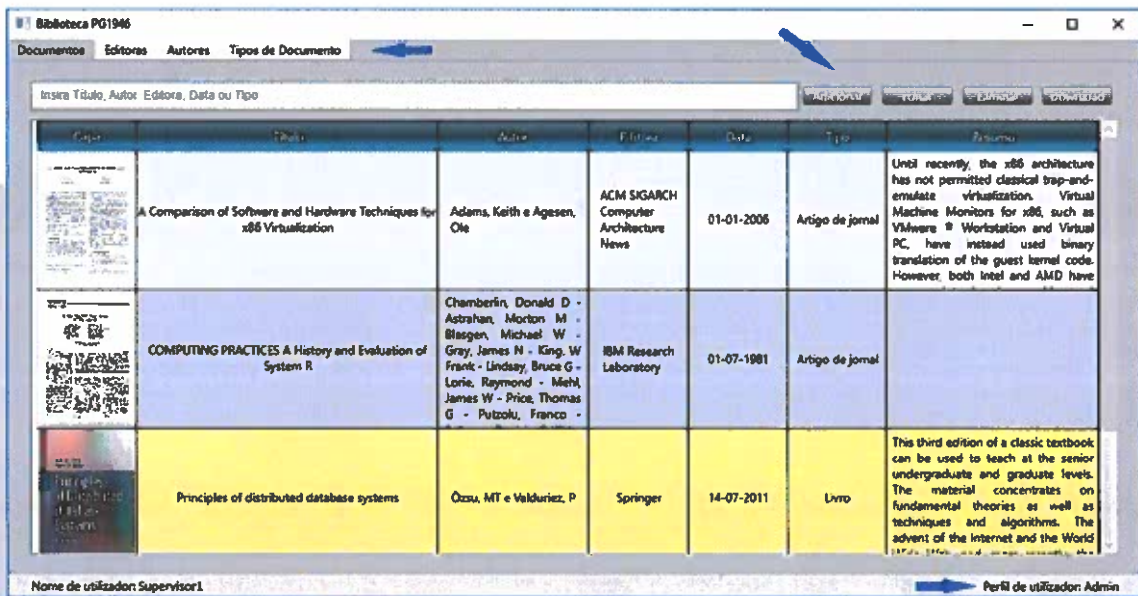


Figura 252: *Layout* Interface Gráfica Principal 3, Fonte Própria

Os interfaces gráficos como já referido anteriormente foram desenvolvidos em WPF. O interface gráfico do ficheiro de *layout* MainWindow.xaml é composto por uma “Grid” principal onde estão inseridos todos os componentes da interface gráfica. Dentro desta “Grid” principal está inserido um “TabControl”, onde dentro deste temos os “TabItem”, que correspondem a cada Tab que verificamos no interface gráfico da aplicação e nos permitem a visualização dos vários conteúdos de cada tabela da base de dados. (Microsoft Corporation, 2011)



Figura 253: *Layout* Interface Gráfica Principal 4, Fonte Própria

Dentro de cada “TabItem” temos um “DockPanel”, onde dentro deste foi criada uma nova “Grid”. Na primeira linha da “Grid” temos na primeira coluna uma “TextBox” que será uma caixa de pesquisa na tabela.

```

50 <TextBox Grid.Column="0" Name="TextSearch"
51     Text="{Binding Path=SearchQuery,
52     Mode=TwoWay,
53     Delay=500,
54     UpdateSourceTrigger=PropertyChanged}"
55     Tag="Insira o Título, Autor, Editora, Data ou Tipo de Documento"
56     Style="{StaticResource MyWaterMarkStyle}" />

```

Figura 254: Layout Interface Gráfica Principal 5, Fonte Própria

Na mesma primeira linha da “Grid”, na segunda coluna temos um “StackPanel” onde estão todos os botões inseridos.

```

57 <StackPanel Grid.Column="1" Name="FlButtons" Orientation="Horizontal" HorizontalAlignment="Right"
58     <Button Name="BtnAdd" Content="Adicionar" Click="BtnAdd_Click" Margin="5" Foreground="White" Style="{DynamicResource StyleButton}" />
59     <Button Name="BtnEdit" Content="Editar" Click="BtnEdit_Click" Margin="5" IsEnabled="False" Style="{DynamicResource StyleButton}" />
60     <Button Name="BtnDel" Content="Eliminar" Click="BtnDel_Click" Margin="5" IsEnabled="False" Style="{DynamicResource StyleButton}" />
61     <Button Name="BtnDownload" Content="Download" Click="BtnDownload_Click" Margin="5" IsEnabled="False" Style="{DynamicResource StyleButton}" />
62 />
63 />
64 />

```

Figura 255: Layout Interface Gráfica Principal 6, Fonte Própria

Na quarta linha temos um “ScrollViewer” para que após estar carregado o conteúdo da tabela, possamos ter um “scroll” para podermos visualizar todo o conteúdo da tabela. Dentro deste “ScrollViewer” criámos uma “DataGrid” que será onde estará todo o conteúdo das nossas tabelas, obtido da base de dados. Nesta “DataGrid” foram criadas as respetivas colunas de cada tabela da base de dados.

```

66 <ScrollViewer Name="ScrollContainer" Grid.Row="3" Margin="0,0,0,20">
67     <DataGrid x:Name="gridDocs" AutoGenerateColumns="False"
68         AlternationCount="2"
69         PreviewMouseWheel="gridDocs_PreviewMouseWheel"
70         ColumnHeaderStyle="{StaticResource HeaderStyle}"
71         RowStyle="{StaticResource RowStyleWithAlternation}"
72         CellStyle="{StaticResource CenterCellStyle}"
73         ItemsSource="{Binding Docs, UpdateSourceTrigger=PropertyChanged}"
74         CanUserAddRows="False"
75         CanUserDeleteRows="False"
76         IsReadOnly="True"
77         SelectedCellsChanged="gridDocs_SelectedCellsChanged"
78         MouseDoubleClick="gridDocs_MouseDoubleClick">
79         <DataGrid.Columns>
80             <DataGridTemplateColumn Header="Capa" Width="100" ...>
81             <DataGridTextColumn Header="Titulo" Width="300" Binding="{Binding Title }" ...>
82             <DataGridTextColumn Header="Autor" Width="150" ...>
83             <DataGridTextColumn Header="Data" Width="100" Binding="{Binding DateDoc, Converter={StaticResource DateConverter}}" />
84             <DataGridTextColumn Header="Tipo" Width="100" Binding="{Binding DocumentType.Description}" />
85             <DataGridTemplateColumn Header="Resumo" Width="200" ...>
86         />
87     />
88 />
89 />
90 />

```

Figura 256: Layout Interface Gráfica Principal 7, Fonte Própria

Foi criada uma outra “Grid” dentro da “Grid” principal, e colocada a sua posição no fundo da “Grid” principal, de modo que na interface gráfica principal seja visualizado no fundo da mesma. Dentro desta “Grid” foi criado um “StatusBar” com dois “StatusBarItem”. Um dos “StatusBarItem” foi criado com o posicionamento encostado à esquerda, que nos dará o nome do utilizador com o qual foi feita a autenticação na aplicação e o outro “StatusBarItem” foi criado com o seu posicionamento encostado à direita, que nos dará a informação de qual o perfil com que foi autenticado.

```

346 <Grid VerticalAlignment="bottom">
347   <StatusBar>
348     <StatusBar.ItemsPanel>
349       <ItemsPanelTemplate>
350     </StatusBar.ItemsPanel>
351     <StatusBarItem>
352       <TextBlock Name="LblUser" Margin="20 0 0 0">User</TextBlock>
353     </StatusBarItem>
354     <StatusBarItem Grid.Column="1" HorizontalContentAlignment="Right">
355       <TextBlock Name="LblProfile" Margin="0 0 20 0">Profile</TextBlock>
356     </StatusBarItem>
357   </StatusBar>
358 </Grid>
359 </Grid>
370
371 </Window>

```

Figura 257: Layout Interface Gráfica Principal 8, Fonte Própria

4.3.4.1.2 Interface Gráfica na Tab Editoras

Sendo selecionada a Tab Editoras na interface gráfica do ficheiro de *layout* MainWindow.xaml, esta será a visualização que o administrador terá dela.

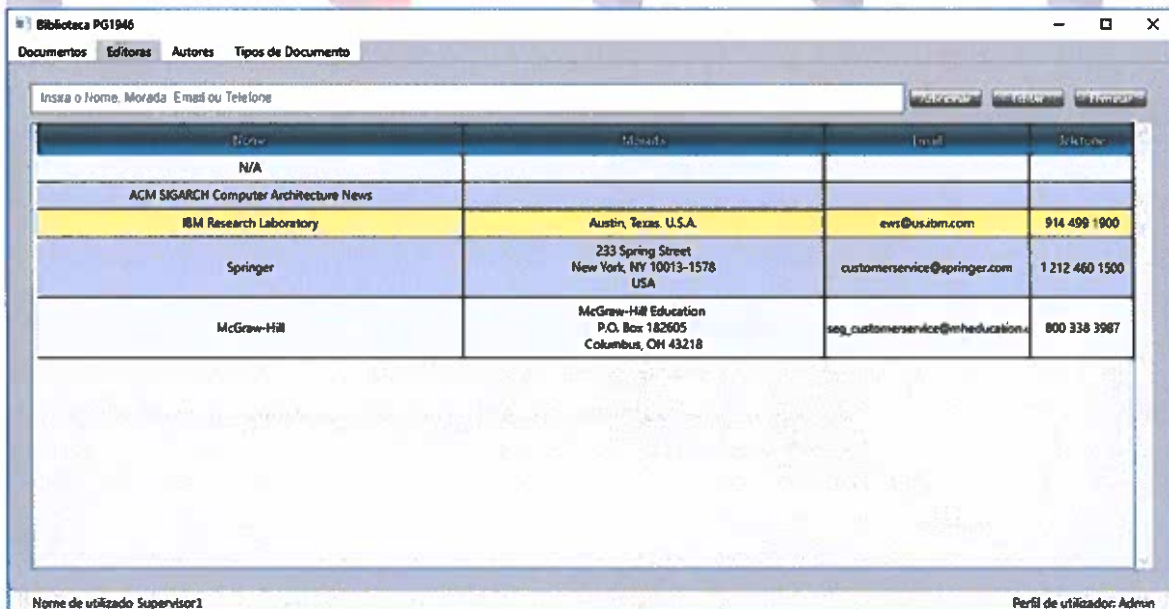


Figura 258: Layout Interface Gráfica Principal 9, Fonte Própria

4.3.4.1.3 Interface Gráfica na Tab Autores

Na interface gráfica do ficheiro de *layout* MainWindow.xaml, seleccionando a Tab Autores o administrador irá obter a seguinte visualização.

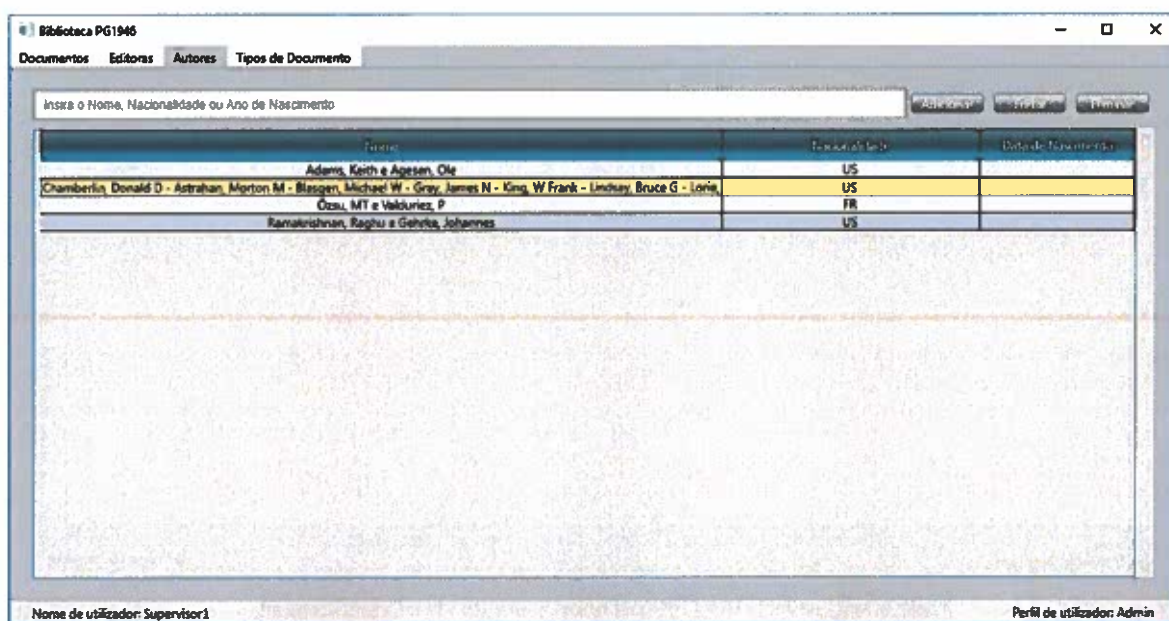


Figura 259: Layout Interface Gráfica Principal 10, Fonte Própria

4.3.4.1.4 Interface Gráfica na Tab Tipos de Documento

A interface gráfica do ficheiro de *layout* MainWindow.xaml, quando o administrador seleccionar a Tab Tipos de Documentos, terá a seguinte visualização dessa Tab.

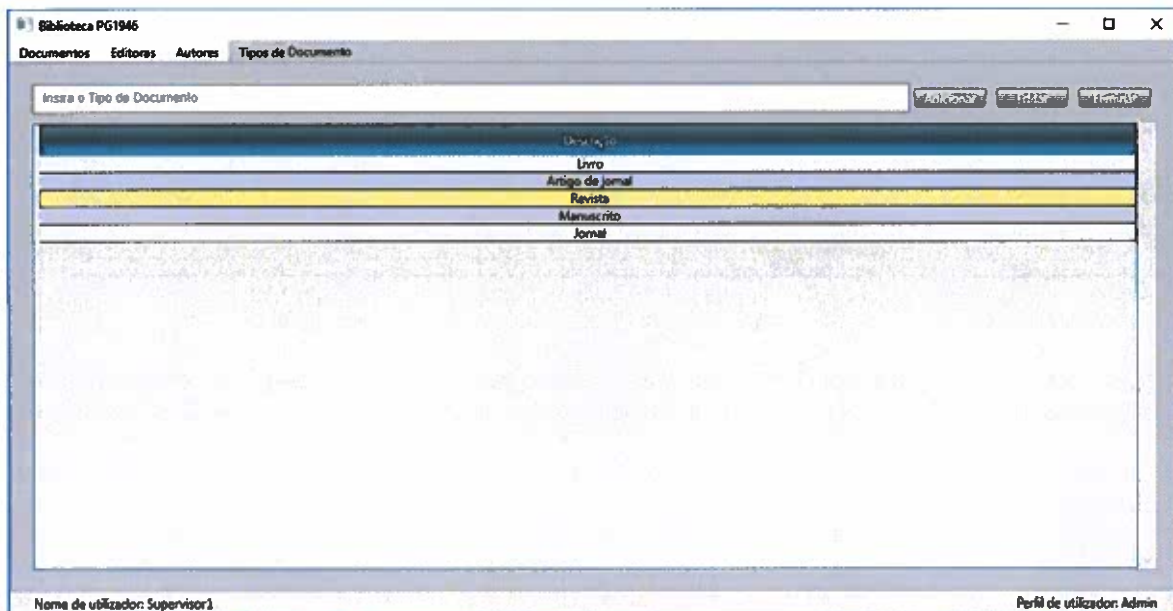


Figura 260: *Layout* Interface Gráfica Principal 11, Fonte Própria

4.3.4.2 Interface Gráfica Adicionar e Editar

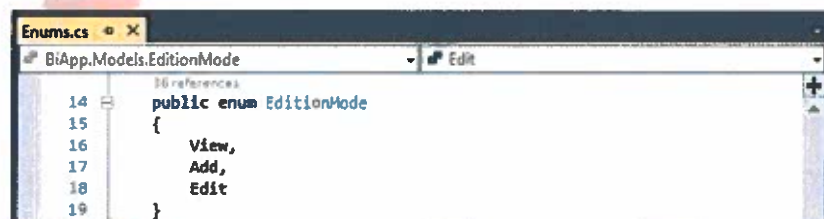
Vamos explicar nesta secção as interfaces gráficas para editar e adicionar Documentos, todas as restantes interfaces gráficas Editoras, Autores e Tipos de Documentos, são construídas da mesma forma, pelo que só iremos mostrar o seu *layout* para o administrador.

Como já explicado anteriormente, apenas os utilizadores com perfil de administração da aplicação, terão acesso às interfaces gráficas para adicionar e editar, pois os botões não estão visíveis aos utilizadores com o perfil de utilizador. Uma vez que foi associado o evento “MouseDownClick” na interface gráfica principal, este evento tem como função chamar a interface gráfica de edição, como tal na função que este evento chama, irá ser validado o tipo de acesso à aplicação.

4.3.4.2.1 Interface Gráfica Documentos

Quando o administrador da aplicação carrega no botão “Adicionar”, “Editar” ou faz duplo “click” em cima de uma das linhas da tabela, é iniciada uma nova interface gráfica do ficheiro de *layout* DocumentWindow.xaml.

Na interface gráfica do ficheiro de *layout* DocumentWindow.xaml, a base será a mesma para o modo de editar ou adicionar, diferindo apenas o conteúdo que dentro de cada um dos seus campos. Para diferenciar o modo com que é gerado o ficheiro de *layout* DocumentWindow.xaml, na classe Enums.cs, já referida atrás para diferenciar o modo de acesso à aplicação, nesta classe foi criada uma lista de constantes “EditionMode”, que irá diferenciar o modo com que a interface será gerada.



```
Enums.cs
BiApp.Models.EditionMode
16 references
14 public enum EditionMode
15 {
16     View,
17     Add,
18     Edit
19 }
```

Figura 261: Interface Gráfica Adicionar/Editar Documentos 1, Fonte Própria

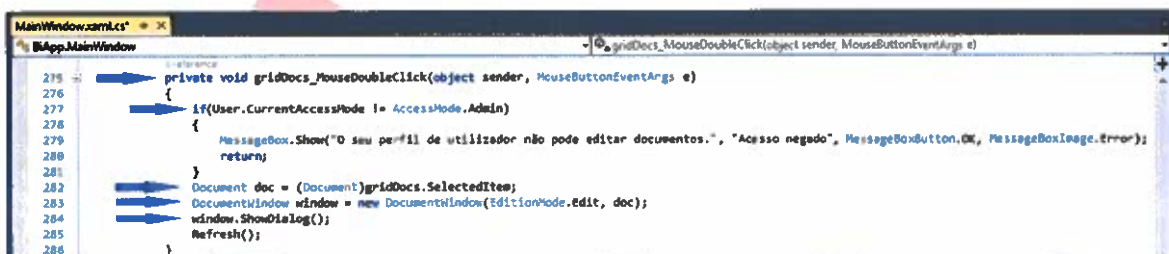
Quando carrega no botão “Editar”, podemos ver numa das imagens anteriores do *layout* dos botões que associado a este botão existe um evento, é chamado o evento “Click” que por sua vez chamada a função “BtnEditP_Click”. Esta função obtém o *item* que foi selecionado antes de ser chamado o evento “Click”, em seguida é chamada a classe DocumentWindow.xaml.cs juntamente com o ficheiro de *layout* DocumentWindow.xaml e é passada à classe DocumentWindow.xaml.cs o conteúdo do *item* que estava selecionado.



```
MainWindow.xaml.cs
BiApp.MainWindow
1 reference
188 private void BtnEdit_Click(object sender, RoutedEventArgs e)
189 {
190     try
191     {
192         Document doc = (Document)gridDocs.SelectedItem;
193         DocumentWindow window = new DocumentWindow(EditionMode.Edit, doc);
194         window.ShowDialog();
195         Refresh();
196     }
```

Figura 262: Interface Gráfica Editar Documentos 1, Fonte Própria

Quando é feito o duplo “click” em cima de uma das linhas da tabela é chamado o evento “MouseDownClick”, na imagem com o “DataGrid” do ficheiro de *layout* MainWindow.xaml podemos ver este evento associado ao “DataGrid”. Este evento “MouseDownClick” chama a função “gridDocs_MouseDoubleClick”. Como mencionado anteriormente, esta classe começa por validar o tipo de acesso à aplicação e caso não seja de administração da aplicação devolve uma mensagem de erro, dizendo que o perfil não tem permissões para edição. Após a validação anterior, esta função tal como a função anterior, “BtnEditP_Click”, irá efetuar o mesmo processo para iniciar a interface gráfica de edição.



```
275 private void gridDocs_MouseDoubleClick(object sender, MouseButtonEventArgs e)
276 {
277     if (User.CurrentAccessMode != AccessMode.Admin)
278     {
279         MessageBox.Show("O seu perfil de utilizador não pode editar documentos.", "Acesso negado", MessageBoxButton.OK, MessageBoxImage.Error);
280         return;
281     }
282     Document doc = (Document)gridDocs.SelectedItem;
283     DocumentWindow window = new DocumentWindow(EditMode.Edit, doc);
284     window.ShowDialog();
285     Refresh();
286 }
```

Figura 263: Interface Gráfica Editar Documentos 2, Fonte Própria

Após ser chamado o ficheiro de *layout* Documentwindow.xaml em modo editar, o utilizador visualiza a interface gráfica no modo de edição, com o conteúdo do *item* que havia sido selecionado na tabela da interface gráfica principal.

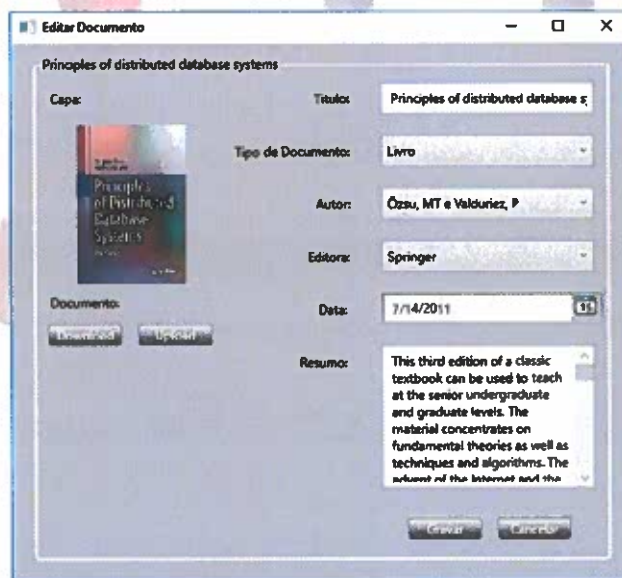


Figura 264: Interface Gráfica Editar Documentos 3, Fonte Própria

Quando é escolhido o botão “Adicionar”, podemos ver numa das imagens anteriores do *layout* dos botões que associado a este botão existe um evento, é chamado o evento “Click” que por sua vez chamada a função “BtnAdd_Click”. Esta função obtém chamada a classe DocumentWindow.xaml.cs juntamente com o ficheiro de *layout* DocumentWindow.xaml, em modo adicionar.



```
172 private void BtnAdd_Click(object sender, RoutedEventArgs e)
173 {
174     try
175     {
176         DocumentWindow window = new DocumentWindow(EditionMode.Add);
177         bool? result = window.ShowDialog();
178         if (result.HasValue && result.Value)
179             Refresh();
180     }
}
```

Figura 265: Interface Gráfica Adicionar Documentos 1, Fonte Própria

Ao ser iniciada a interface gráfica do ficheiro de *layout* DocumentWindow.xaml no modo adicionar, o administrador terá a seguinte visualização da mesma.

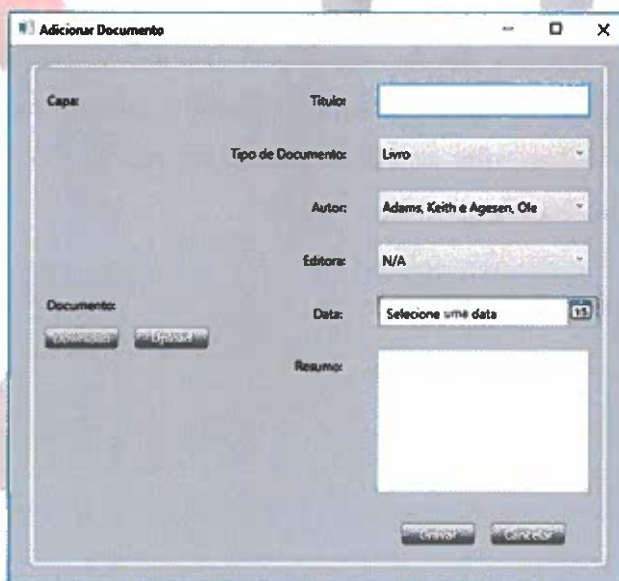


Figura 266: Interface Gráfica Adicionar Documentos 2, Fonte Própria

Como podemos verificar a base da interface gráfica do ficheiro de *layout* DocumentWindow.xaml é idêntica, pelo que vamos explicar o seu *layout*.

Este ficheiro de *layout* DocumentWindow.xaml é composto por uma “GroupBox” que no canto superior esquerdo terá um título, este título será o Título do documento em questão, pois este campo título da “GroupBox” está associado ao conteúdo da caixa de texto Título.

```

18 |
19 | <GroupBox Margin="20 10 20 20" Header="{Binding ElementName=TxtTitle, Path=Text}">
20 |     <Grid...>
116 | </GroupBox>
117 |

```

Figura 267: Interface Gráfica Adicionar/Editar Documentos 2, Fonte Própria

Dentro da “GroupBox” criámos uma “Grid” e dentro desta teremos todo o conteúdo de cada campo da tabela Documentos. Para dar indicação ao administrador a que conteúdo corresponde cada campo e o seu nome foram criadas “TextBlock”. Para permitir ao administrador inserir o conteúdo em cada campo foram criadas “TextBox”, estas “TextBox” no caso da interface gráfica estar no modo adicionar estarão todos vazios, no caso de estar no modo editar estarão preenchidos com o conteúdo do item selecionado na interface gráfica principal. Uma vez que os campos Tipo de Documento, Autor e Editora provêm de outras tabelas da base de dados, foram criadas “ComboBox” em cada um destes campos, em que nestas estará a Descrição no caso do Tipo de Documento, no caso do Autor e da Editora estará o nome, desta “ComboBox” o administrador poderá escolher as disponíveis na base de dados, quer em modo de adicionar, quer em modo de editar. De modo a exibir a imagem da cada da capa de Documento que está a ser inserido ou editado criámos um campo do tipo “Image”.

```

37 | <TextBlock Text="Título:" Grid.Column="1" Grid.Row="0" VerticalAlignment="Center".../>
39 | <TextBox Name="TxtTitle" VerticalContentAlignment="Center" Grid.Column="2" Grid.Row="0" Height="30" Margin="10".../>
49 | <TextBlock Text="Tipo de Documento:" Grid.Column="1" Grid.Row="1" VerticalAlignment="Center" HorizontalAlignment="Right".../>
51 | <ComboBox Name="CmbDocumentType" Grid.Column="2" Grid.Row="1" VerticalContentAlignment="Center" Height="30" Margin="10".../>
54 | <TextBlock Text="Autor:" Grid.Column="1" Grid.Row="2" VerticalAlignment="Center" HorizontalAlignment="Right".../>
56 | <ComboBox Name="CmbAuthor" Grid.Column="2" Grid.Row="2" VerticalContentAlignment="Center" Height="30" Margin="10".../>
59 | <TextBlock Text="Editora:" Grid.Column="1" Grid.Row="3" VerticalAlignment="Center" HorizontalAlignment="Right".../>
61 | <ComboBox Name="CmbPublisher" Grid.Column="2" Grid.Row="3" VerticalContentAlignment="Center" Height="30" Padding="5, 5, 5, 5".../>
64 | <TextBlock Text="Data:" Grid.Column="1" Grid.Row="4" VerticalAlignment="Center" HorizontalAlignment="Right" Margin="0, 10, 10, 17".../>
66 | <TextBlock Text="Resumo:" Grid.Column="1" Grid.Row="5" VerticalAlignment="Center" HorizontalAlignment="Right" Margin="0, 10, 10, 17".../>
68 | <TextBlock x:Name="TxtResume" Text="{Binding Doc.Resume}" ScrollViewer.HorizontalScrollBarVisibility="Disabled".../>
77 | <TextBlock Text="Capa:" Grid.Row="0" VerticalAlignment="Center" HorizontalAlignment="Right" Margin="0, 18, 13, 17" Height="16".../>
79 | <Image Name="ImgCover" Source="{Binding Doc.CoverBytes, NotifyOnSourceUpdated=True, Converter={StaticResource imgConverter}}".../>
87 | <TextBlock Text="Documento:" Grid.Row="4" VerticalAlignment="Center" HorizontalAlignment="Right" Margin="0, 10, 99, 24" Height="18".../>

```

Figura 268: Interface Gráfica Adicionar/Editar Documentos 3, Fonte Própria

Foram criados dois “StackPanel” de modo a colocarmos os botões desta interface gráfica. Foi criado um “StackPanel” onde foram colocados os botões referentes ao documento, um para “Download” e outro para “Upload”. No outro “StackPanel” criado foram colocados os botões “Gravar” e “Cancelar”.

```

90 <StackPanel Orientation="Horizontal" Grid.Row="4" Margin="0,24,0,25" Grid.RowSpan="2" >
91 <Button x:Name="BtnDownload" Content="Download" Height="20" Width="75" Margin="5" Click="BtnDownload_Click"...
93 <Button x:Name="BtnUpload" Content="Upload" Height="20" Width="75" Margin="5" Click="BtnUpload_Click".../>
95 </StackPanel>
96 <StackPanel Orientation="Horizontal" Grid.Column="2" Grid.Row="7" HorizontalAlignment="Center"
97 VerticalAlignment="Bottom" Height="30" Margin="29,0,29,-29" Width="170">
98 <Button x:Name="BtnSave" Content="Gravar" Height="20" Width="75" Margin="5" Click="BtnSave_Click".../>
100 <Button x:Name="BtnCancel" Content="Cancelar" Height="20" Width="75" Margin="5" Click="BtnCancel_Click".../>
102 </StackPanel>

```

Figura 269: Interface Gráfica Adicionar/Editar Documentos 4, Fonte Própria

De modo a facilitar a introdução da data e de modo a que seja introduzida no formato correto, correspondente ao definido na base de dados, foi criado um “DatePicker”. Este “DatePicker” após selecionada a data pretendida, coloca essa mesma data no formato pretendido, no campo “Data”.

```

105 <DatePicker HorizontalAlignment="Left"
106 x:Name="Dtpick"
107 Margin="10,10,0,0"
108 VerticalAlignment="Top"
109 SelectedDate="{Binding Doc.DateDoc, Mode=TwoWay, UpdateSourceTrigger=PropertyChanged}"
110 Grid.Row="4" Width="200" Height="30" RenderTransformOrigin="0.506,1.1" Grid.Column="2">
111 <DatePicker.Resources>
112 <Style TargetType="{x:Type DatePickerTextBox}">
113 <Setter Property="Padding" Value="2" />
114 <Setter Property="Text" Value="Selecione uma data"/>
115 <Setter Property="IsReadOnly" Value="True"/>
116 </Style>
117 </DatePicker.Resources>
118 </DatePicker>

```

Figura 270: Interface Gráfica Adicionar/Editar Documentos 5, Fonte Própria

4.3.4.2.2 Interface Gráfica Editoras

A interface gráfica do ficheiro de *layout* PublisherWindow.xaml em modo adicionar, terá a seguinte visualização para o administrador.

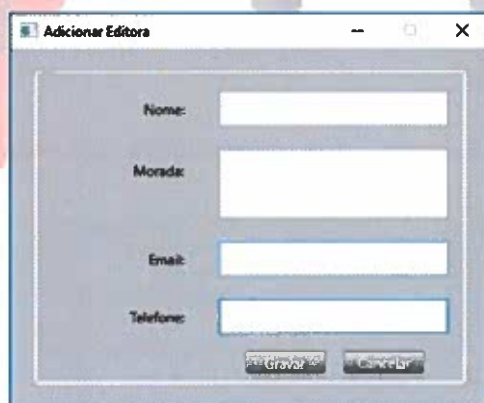
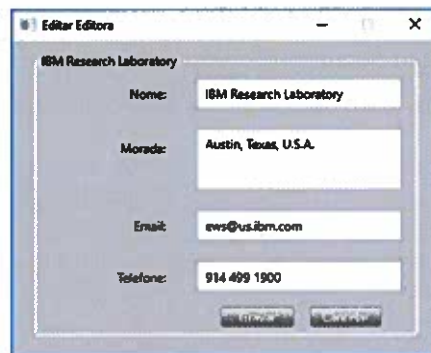


Figura 271: Interface Gráfica Adicionar Editoras, Fonte Própria

No caso da interface gráfica ser iniciada em modo editar terá a seguinte visualização.

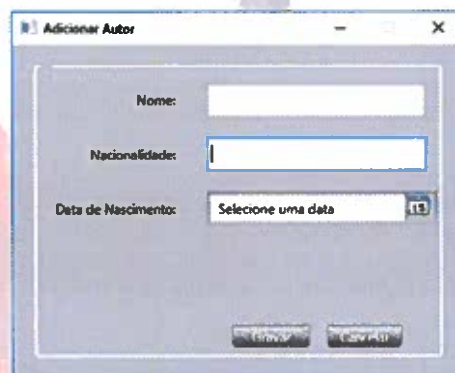


The screenshot shows a window titled "Editar Editoras" with a header "IBM Research Laboratory". It contains four text input fields: "Nome:" with the value "IBM Research Laboratory", "Morada:" with "Austin, Texas, U.S.A.", "Email:" with "ews@us.ibm.com", and "Telefone:" with "914 499 1900". At the bottom, there are two buttons labeled "OK" and "Cancelar".

Figura 272: Interface Gráfica Editar Editoras, Fonte Própria

4.3.4.2.3 Interface Gráfica Autores

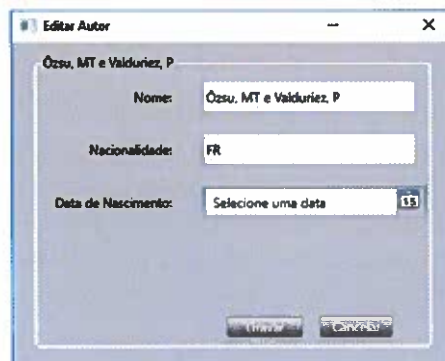
Na interface gráfica do ficheiro de *layout* AuthorsWindow.xaml em modo adicionar, ficará com a seguinte visualização ara o administrador.



The screenshot shows a window titled "Adicionar Autor". It contains three input fields: "Nome:" (empty), "Nacionalidade:" (empty), and "Data de Nascimento:" with a dropdown menu showing "Selecione uma data" and a calendar icon. At the bottom, there are two buttons labeled "OK" and "Cancelar".

Figura 273: Interface Gráfica Adicionar Autor, Fonte Própria

Sendo iniciada em modo editar, o *layout* apresentado será o seguinte.



The screenshot shows a window titled "Editar Autor" with a header "Ózsu, MT e Valduriez, P". It contains three input fields: "Nome:" with the value "Ózsu, MT e Valduriez, P", "Nacionalidade:" with "FR", and "Data de Nascimento:" with a dropdown menu showing "Selecione uma data" and a calendar icon. At the bottom, there are two buttons labeled "OK" and "Cancelar".

Figura 274: Interface Gráfica Editar Autor, Fonte Própria

4.3.4.2.4 Interface Gráfica Tipos de Documento

O ficheiro de *layout* PublisherWindow.xaml quando iniciado em modo adicionar, irá ser gerada a seguinte interface gráfica.

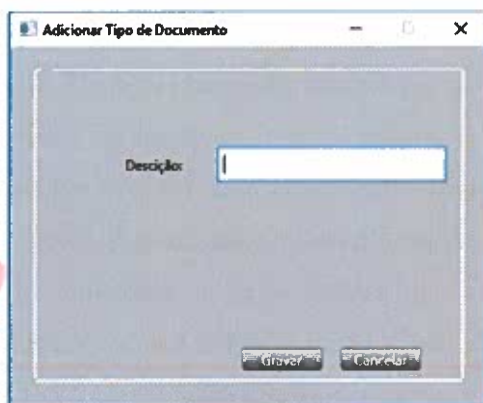


Figura 275: Interface Gráfica Adicionar Tipo de Documento, Fonte Própria

Caso a interface gráfica seja iniciada em modo editar terá a seguinte visualização.

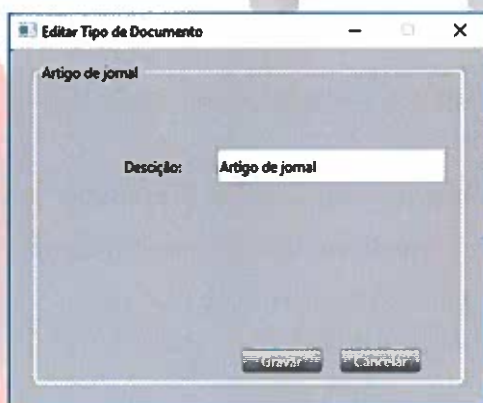


Figura 276: Interface Gráfica Editar Tipo de Documento, Fonte Própria

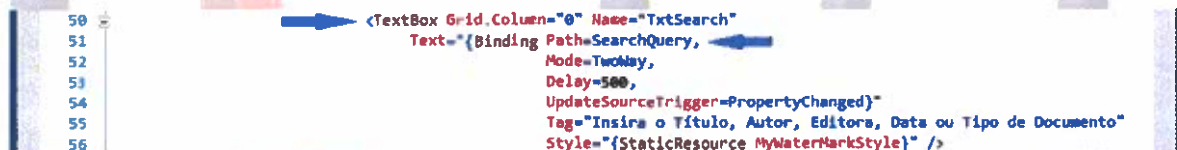
4.3.5 Funcionalidades da Aplicação

4.3.5.1 Campo de Pesquisa

Foi criada uma caixa de pesquisa em cada uma das Tab da interface gráfica do ficheiro de *layout* MainWindow.xaml, todas elas têm um funcionamento idêntico procurando as letras ou palavras introduzidas nessa caixa, no conteúdo da tabela referente à Tab selecionada.

Para que apenas seja disponibilizado o conteúdo da tabela que contenha os valores introduzidos na caixa de pesquisa, em qualquer uma das suas colunas, foram efetuadas algumas configurações nos parâmetros das propriedades desta caixa de texto.

Através do *binding*, que é um método de associação entre o ficheiro de *layout* e a classe, neste caso, entre o ficheiro de *layout* MainWindow.xaml e a classe MainWindow.xaml.cs, a propriedade "Path" vai indicar o caminho para a função com a qual será feito o *binding*, neste caso a função "SearchQuery".

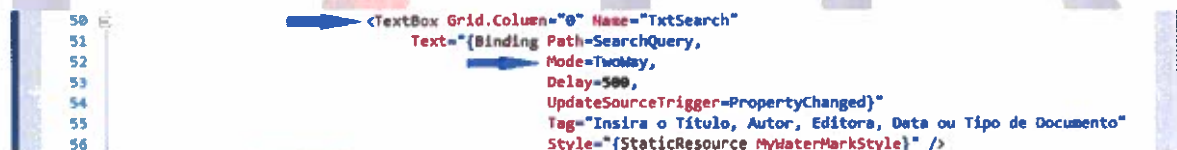


```
50 <TextBox Grid.Column="0" Name="TxtSearch"
51 Text="{Binding Path=SearchQuery,
52 Mode=TwoWay,
53 Delay=500,
54 UpdateSourceTrigger=PropertyChanged}"
55 Tag="Insira o Título, Autor, Editora, Data ou Tipo de Documento"
56 Style="{StaticResource MyWaterMarkStyle}" />
```

The screenshot shows the XAML code for a search field. A blue arrow points to the `Name="TxtSearch"` attribute. Another blue arrow points to the `Text="{Binding Path=SearchQuery,` attribute. A third blue arrow points to the `Mode=TwoWay,` attribute.

Figura 277: Campo Pesquisa 1, Fonte Própria

No *binding* da caixa de texto com o nome "TxtSearch" temos também a propriedade "Mode" com o valor "TwoWay", que determina que este *binding* irá funcionar nos dois sentidos, da MainWindow.xaml para a MainWindow.xaml.cs e vice versa.

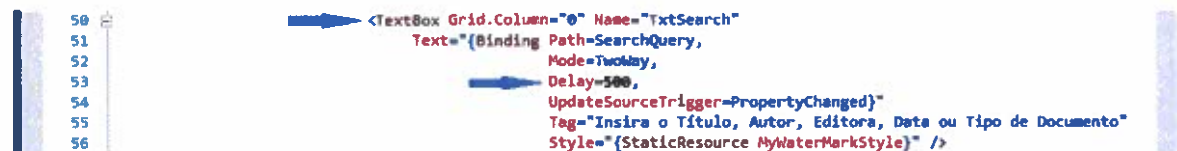


```
50 <TextBox Grid.Column="0" Name="TxtSearch"
51 Text="{Binding Path=SearchQuery,
52 Mode=TwoWay,
53 Delay=500,
54 UpdateSourceTrigger=PropertyChanged}"
55 Tag="Insira o Título, Autor, Editora, Data ou Tipo de Documento"
56 Style="{StaticResource MyWaterMarkStyle}" />
```

The screenshot shows the XAML code for a search field. A blue arrow points to the `Mode=TwoWay,` attribute. Another blue arrow points to the `Delay=500,` attribute.

Figura 278: Campo Pesquisa 2, Fonte Própria

A propriedade "Delay" irá indicar o tempo de atraso com que será atualizada a fonte do *binding*.



```
50 <TextBox Grid.Column="0" Name="TxtSearch"
51 Text="{Binding Path=SearchQuery,
52 Mode=TwoWay,
53 Delay=500,
54 UpdateSourceTrigger=PropertyChanged}"
55 Tag="Insira o Título, Autor, Editora, Data ou Tipo de Documento"
56 Style="{StaticResource MyWaterMarkStyle}" />
```

The screenshot shows the XAML code for a search field. A blue arrow points to the `Delay=500,` attribute.

Figura 279: Campo Pesquisa 3, Fonte Própria

Temos também a propriedade "UpdateSourceTrigger", à qual demos o valor "PropertyChanged", esta propriedade irá verificar se existe alguma alteração na caixa de texto e caso exista irá despoletar a função "SearchQuery".

```

50 <TextBox Grid.Column="0" Name="TxtSearch"
51       Text="{Binding Path=SearchQuery,
52       Mode=TwoWay,
53       Delay=500,
54       UpdateSourceTrigger=PropertyChanged}"
55       Tag="Insira o Título, Autor, Editora, Data ou Tipo de Documento"
56       Style="{StaticResource MyWaterMarkStyle}" />

```

Figura 280: Campo Pesquisa 4, Fonte Própria

Na função "SearchQuery" temos o método "set" que em primeiro lugar recebe o valor que está na caixa de texto "TxtSearch" e em seguida chama a função NotifyPropertyChanged.

```

MainWindow.xaml.cs*
BiApp.MainWindow
38 private string searchQuery;
39 public string SearchQuery
40 {
41     get
42     {
43         return searchQuery;
44     }
45     set
46     {
47         searchQuery = value;
48         NotifyPropertyChanged("SearchQuery");
49         Refresh();
50     }
51 }

```

Figura 281: Campo Pesquisa 5, Fonte Própria

A função "NotifyPropertyChanged" funciona com base na interface "INotifyPropertyChanged" que foi implementada na classe MainWindow.cs.

```

MainWindow.xaml.cs*
BiApp.MainWindow
26 public partial class MainWindow : Window, INotifyPropertyChanged
27 {

```

Figura 282: Campo Pesquisa 6, Fonte Própria

A função "NotifyPropertyChanged" vai verificar se o atributo "PropertyChanged" que atribuímos à propriedade "UpdateSourceTrigger" da "TxtSearch" é nulo ou não, o seu valores só

será diferente de nulo se tiver existido alguma alteração na "TxtSearch". Caso seja diferente de nulo irá voltar a função "SearchQuery".

```
MainWindow.xaml.cs* x
BiApp.MainWindow
public event PropertyChangedEventHandler PropertyChanged;
31 protected void NotifyPropertyChanged(string propertyName)
32 {
33     if (PropertyChanged != null)
34         PropertyChanged(this, new PropertyChangedEventArgs(propertyName));
35 }
```

Figura 283: Campo Pesquisa 7, Fonte Própria

Voltando à função "SearchQuery" a mesma irá chamar a função "Refresh".

```
MainWindow.xaml.cs* x
BiApp.MainWindow
private string searchQuery;
39 public string SearchQuery
40 {
41     get
42     {
43         return searchQuery;
44     }
45     set
46     {
47         searchQuery = value;
48         NotifyPropertyChanged("SearchQuery");
49         Refresh();
50     }
51 }
```

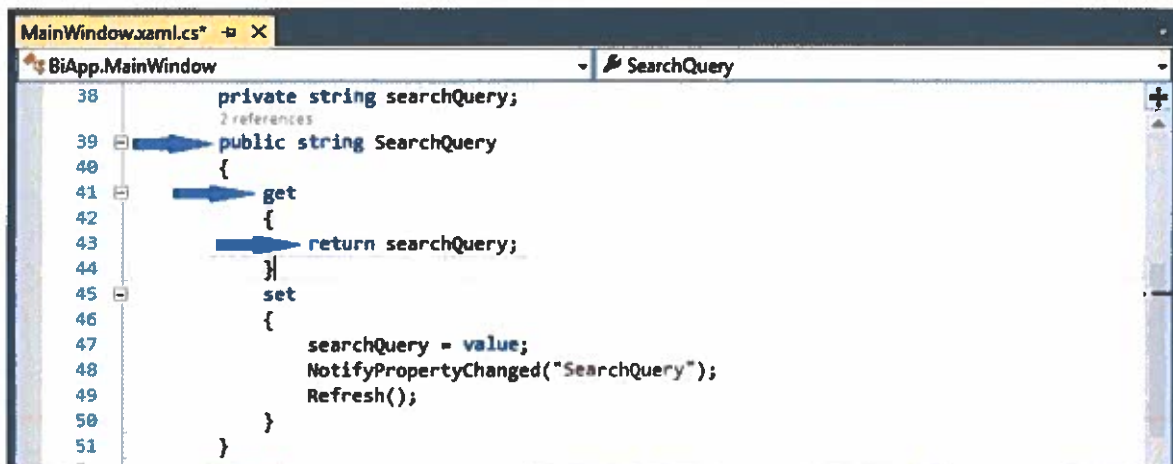
Figura 284: Campo Pesquisa 8, Fonte Própria

A função "Refresh" irá receber o conteúdo da "TxtSearch" através do método "get" da função "SearchQuery" de modo a validar se conteúdo da "TxtSearch" é nulo ou não.

```
MainWindow.xaml.cs* x
BiApp.MainWindow
11 references
247 private void Refresh()
248 {
249     Task.Run(() =>
250     {
251         if (string.IsNullOrEmpty(SearchQuery))
252         {
253             this.Docs = DbService.Instance.GetAllDocs();
254         }
255         else
256         {
257             this.Docs = DbService.Instance.SearchDocs(SearchQuery);
258         }
259         UpdateUIAsync();
260     });
261 }
```

Figura 285: Campo Pesquisa 9, Fonte Própria

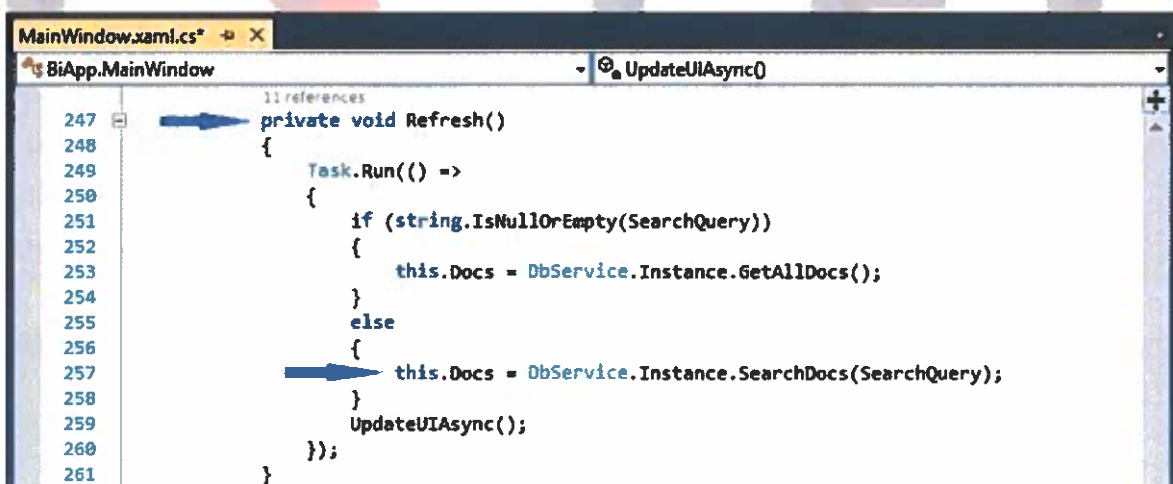
O conteúdo da “TxtSearch”, como referido é passado pelo método “get” da função “SearchQuery”.



```
MainWindow.xaml.cs
BiApp.MainWindow
38 private string searchQuery;
39 public string SearchQuery
40 {
41     get
42     {
43         return searchQuery;
44     }
45     set
46     {
47         searchQuery = value;
48         NotifyPropertyChanged("SearchQuery");
49         Refresh();
50     }
51 }
```

Figura 286: Campo Pesquisa 10, Fonte Própria

Não sendo nulo o valor passado à função "Refresh", esta irá chamar a classe DbServices.cs, que irá efetuar a função "SearchDocs" e passar-lhe o valor da "TxtSearch" para que seja feita uma nova instrução SQL.



```
MainWindow.xaml.cs
BiApp.MainWindow
247 private void Refresh()
248 {
249     Task.Run(() =>
250     {
251         if (string.IsNullOrEmpty(SearchQuery))
252         {
253             this.Docs = DbService.Instance.GetAllDocs();
254         }
255         else
256         {
257             this.Docs = DbService.Instance.SearchDocs(SearchQuery);
258         }
259         UpdateUIAsync();
260     });
261 }
```

Figura 287: Campo Pesquisa 11, Fonte Própria

Com base no conteúdo da “TxtSearch” passado pela função “Refresh”, a função “SearchDocs” da classe DbServices.cs irá passar a *string* com a nova instrução SQL, à função “GetDataTable” da classe DbManager, para que o resultado dessa instrução executada pela função “GetDataTable” seja passado ao objeto ADO.NET “DataTable”. (Microsoft Corporation, n.d.-a)

```
DbService.cs x
BiApp.Services.DbService UpdateDoc(Document doc)
1 reference
98 public List<Document> SearchDocs(string search)
99 {
100     List<Document> result = new List<Document>();
101     DataTable dt = DbManager.Instance.GetDataTable(
102         string.Format("select doc." +
103             "from Document doc " +
104             "inner join DocumentType dt " +
105             "on dt.Id = doc.DocumentTypeId " +
106             "inner join Publisher pub " +
107             "on pub.Id = doc.PublisherId " +
108             "inner join Author a " +
109             "on a.Id = doc.AuthorId " +
110             "where doc.Title like '{0}%' or " +
111             "doc.DateDoc like '{0}%' or " +
112             "doc.Resume like '{0}%' or " +
113             "dt.Description like '{0}%' or " +
114             "pub.Name like '{0}%' or " +
115             "a.Name like '{0}%", search));
116     if (dt != null && dt.Rows.Count > 0)
117     {
118         foreach (DataRow r in dt.Rows)
119             result.Add(FillDocument(r));
120     }
121     return result;
122 }
```

Figura 288: Campo Pesquisa 12, Fonte Própria

A função “GetDataTable” da classe DbManager.cs, recebe a *string* passada pela função “SearchDocs” e executa a instrução SQL, para que a base de dados devolva os conteúdos da tabela Documentos que contenham o valor passado na “TxtSearch”. Após receber os conteúdos da base de dados referentes à instrução executada, a função “GetDataTable” irá colocar esse conteúdo numa “DataTable” e devolve essa “DataTable” à função “SearchDocs”. (Microsoft Corporation, n.d.-b)

```
DbManager.cs x
BiApp.DataAccess.DbManager GetDataTable(string selectStatement)
17 references
13 public DataTable GetDataTable(string selectStatement)
14 {
15     DataTable dt = new DataTable();
16
17     using (SqlConnection con = new SqlConnection(ConfigHelper.Instance.MyConnectionString))
18     using (SqlDataAdapter adp = new SqlDataAdapter(selectStatement, con))
19     {
20         adp.Fill(dt);
21     }
22     return dt;
23 }
```

Figura 289: Campo Pesquisa 13, Fonte Própria

A função “SearchDocs” valida que a “DataTable” recebida não está vazia e envia-a à função “FillDocument”.

```

DbService.cs
BiApp.Services.DbService
UpdateDoc(Document doc)

1 reference
98 public List<Document> SearchDocs(string search)
99 {
100     List<Document> result = new List<Document>();
101     DataTable dt = DbManager.Instance.GetDataTable(
102         string.Format("select doc." +
103             "from Document doc " +
104             "inner join DocumentType dt " +
105             "on dt.Id = doc.DocumentTypeId " +
106             "inner join Publisher pub " +
107             "on pub.Id = doc.PublisherId " +
108             "inner join Author a " +
109             "on a.Id = doc.AuthorId " +
110             "where doc.Title like '{0}%' or " +
111             "doc.DateDoc like '{0}%' or " +
112             "doc.Resume like '{0}%' or " +
113             "dt.Description like '{0}%' or " +
114             "pub.Name like '{0}%' or " +
115             "a.Name like '{0}%', search));
116     if (dt != null && dt.Rows.Count > 0)
117     {
118         foreach (DataRow r in dt.Rows)
119             result.Add(FillDocument(r));
120     }
121     return result;
122 }
  
```

Figura 290: Campo Pesquisa 14, Fonte Própria

A função “FillDocument” irá preencher todas as colunas da tabela Documentos na Tab Documentos do ficheiro de *layout* MainWindow.xaml.

```

DbService.cs
BiApp.Services.DbService
SearchDocs(string search)

2 references
74 private Document FillDocument(DataRow row)
75 {
76     Document doc = new Document
77     {
78         Id = Convert.ToInt32(row["Id"]),
79         AuthorId = Convert.ToInt32(row["AuthorId"]),
80         PublisherId = Convert.ToInt32(row["PublisherId"]),
81         DocumentTypeId = Convert.ToInt32(row["DocumentTypeId"]),
82         Title = row["Title"].ToString(),
83         CoverBytes = row["CoverBytes"] == DBNull.Value ? null : (byte[])row["CoverBytes"],
84         ContentBytes = row["ContentBytes"] == DBNull.Value ? null : (byte[])row["ContentBytes"],
85         Resume = row["Resume"].ToString()
86     };
87     if (row["DateDoc"] != DBNull.Value)
88         doc.DateDoc = Convert.ToDateTime(row["DateDoc"]);
89
90     doc.Author = Authors.FirstOrDefault(a => a.Id == doc.AuthorId);
91     doc.Publisher = Publishers.FirstOrDefault(p => p.Id == doc.PublisherId);
92     doc.DocumentType = DocumentTypes.FirstOrDefault(t => t.Id == doc.DocumentTypeId);
93     return doc;
94 }
  
```

Figura 291: Campo Pesquisa 15, Fonte Própria

4.3.5.2 Eliminar Conteúdos nas Tabelas

4.3.5.2.1 Tabela Documentos

Uma das funcionalidades dada aos administradores da aplicação é a possibilidade de eliminarem conteúdos das diferentes tabelas na base de dados da aplicação.

Como já demonstrado numa das secções anteriores, na interface gráfica do ficheiro de *layout* MainWindow.xaml, o administrador da aplicação tem acesso a vários botões que lhe permitem realizar diferentes ações no conteúdo das tabelas da base de dados, aqui iremos explicar a funcionalidade do botão “Eliminar”.

O administrador irá seleccionar a linha com o conteúdo da tabela que pretende eliminar e irá carregar no botão eliminar.

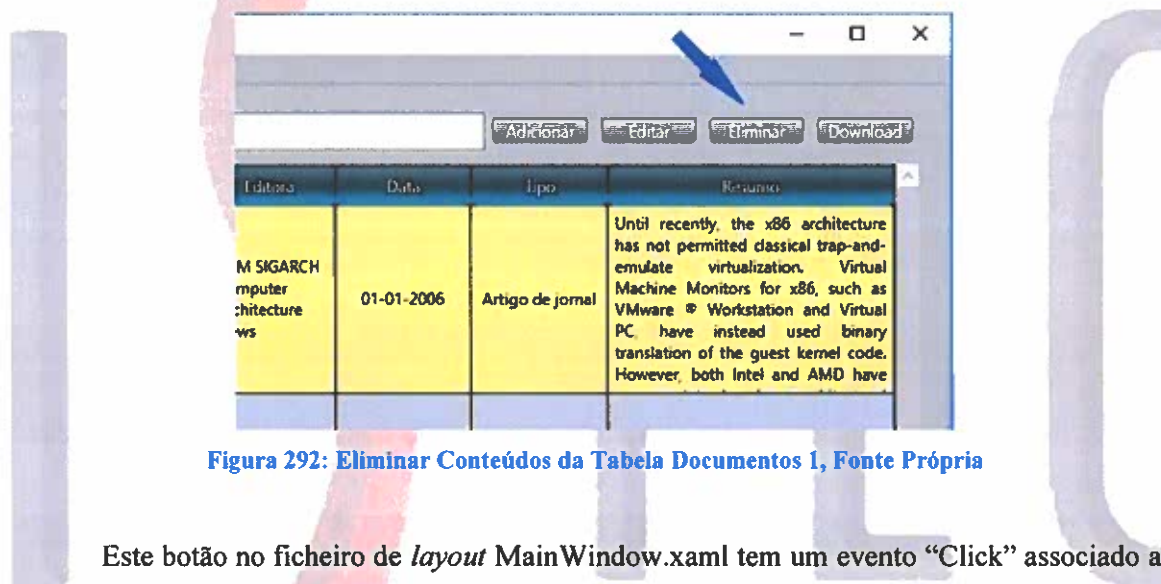


Figura 292: Eliminar Conteúdos da Tabela Documentos 1, Fonte Própria

Este botão no ficheiro de *layout* MainWindow.xaml tem um evento “Click” associado ao mesmo, este evento irá chamar a função “BtnDel_Click”.



Figura 293: Eliminar Conteúdos da Tabela Documentos 2, Fonte Própria

Chamada a função “BtnDel_Click” a mesma irá receber a identificação do conteúdo da linha seleccionada antes de ter sido carregado o botão “Eliminar”. Em seguida é enviada uma mensagem de confirmação, se o administrador pretende ou não eliminar o conteúdo seleccionado e em caso afirmativo será chamada a função “DeleteDoc” da classe DbService.cs e passado a esta a identificação do conteúdo da linha seleccionada.

```
MainWindow.xaml.cs - X
BiApp.MainWindow - BtnDownload_Click(object sender, RoutedEventArgs e)
1 reference
204 private void BtnDel_Click(object sender, RoutedEventArgs e)
205 {
206     try
207     {
208         Document doc = (Document)gridDocs.SelectedItem;
209         if (MessageBox.Show("Tem a certeza que pretende eliminar o item selecionado?", "Confirmação",
210             MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)
211         {
212             DbService.Instance.DeleteDoc(doc.Id);
213             Refresh();
214         }
215     }
216 }
```

Figura 294: Eliminar Conteúdos da Tabela Documentos, Fonte Própria

A função “DeleteDoc” da classe DbService.cs irá passar à função “ExecuteComand” da classe DbManager.cs, a instrução SQL a ser executada na base de dados.

```
DbService.cs - X
BiApp.Services.DbService - RefreshPublishers()
1 reference
211 public bool DeleteDoc(int id)
212 {
213     DbManager.Instance.ExecuteCommand("delete from Document where Id = " + id);
214     return true;
215 }
```

Figura 295: Eliminar Conteúdos da Tabela Documentos 4, Fonte Própria

A função “ExecuteComand” através do objeto ADO.NET “ExecuteNonQuery”, irá executar na base de dados a instrução SQL recebida.

```
DbManager.cs - X
BiApp.DataAccess.DbManager - ExecuteCommand(SqlCommand cmd)
4 references
25 public void ExecuteCommand(string cmdText)
26 {
27     using (SqlConnection con = new SqlConnection(ConfigHelper.Instance.MyConnectionString))
28     using (SqlCommand cmd = new SqlCommand(cmdText, con))
29     {
30         SqlParameter p = new SqlParameter();
31         con.Open();
32         cmd.ExecuteNonQuery();
33     }
34 }
```

Figura 296: Eliminar Conteúdos da Tabela Documentos 5, Fonte Própria

Voltando à função “BtnDel_Click”, após a execução da instrução SQL, a função “BtnDel_Click” chama a função “Refresh”, a qual já explicámos mais atrás o seu funcionamento, esta função irá fazer com que o conteúdo exibido ao administrador da aplicação seja atualizado.

```

204 private void BtnDel_Click(object sender, RoutedEventArgs e)
205 {
206     try
207     {
208         Document doc = (Document)gridDocs.SelectedItem;
209         if (MessageBox.Show("Tem a certeza que pretende eliminar o item selecionado?", "Confirmação",
210             MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)
211         {
212             DbService.Instance.DeleteDoc(doc.Id);
213             Refresh();
214         }
215     }

```

Figura 297: Eliminar Conteúdos da Tabela Documentos 6, Fonte Própria

4.3.5.2.2 Tabelas Editoras, Autores e Tipos de Documentos

O conteúdo das restantes tabelas da base de dados, Editoras, Autores e Tipos de Documento, estão interligadas à tabela Documentos, através de chaves estrangeiras. Nestas é necessária uma outra validação antes de ser possível eliminá-las, teremos de validar se o conteúdo que pretendemos eliminar está já a ser utilizado na tabela Documentos.

Vamos aqui demonstrar apenas o exemplo para os conteúdos da tabela Editoras, para as restantes será idêntico o processo. Tal como na Tab Documentos, o botão “Eliminar” tem um evento associado que chamada a função “BtnDeIP_Click”, esta função chamada então a função “PublisherInUse” da classe DbService.cs.

```

339 private void BtnDeIP_Click(object sender, RoutedEventArgs e)
340 {
341     try
342     {
343         Publisher pub = (Publisher)gridPubs.SelectedItem;
344         if (DbService.Instance.PublisherInUse(pub.Id))
345         {
346             MessageBox.Show("Esta Editora encontra-se associada a um Documento!\nDeve é desassociar-la do Documento para a eliminar.",
347                 "Erro", MessageBoxButton.OK, MessageBoxImage.Error);
348             return;
349         }
350         else if (MessageBox.Show("Tem a certeza que pretende eliminar a Editora selecionada?", "Confirmação",
351             MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)
352         {
353             DbService.Instance.DeletePub(pub.Id);
354             Refresh();
355         }
356     }

```

Figura 298: Eliminar Conteúdos da Tabela Editora, Autor e Tipos de Documento 1, Fonte Própria

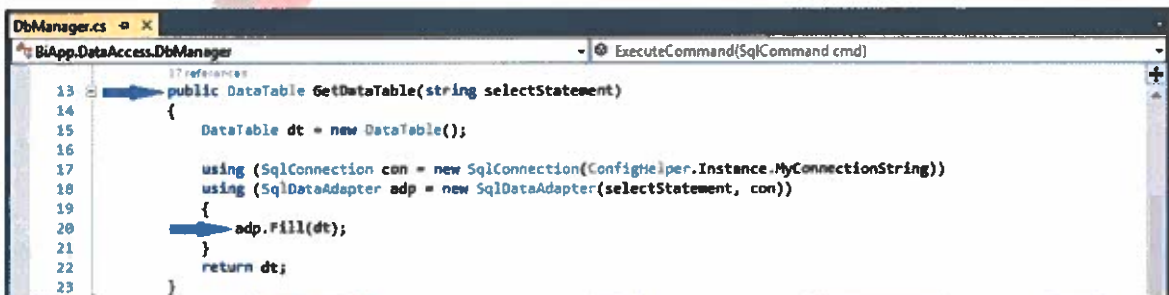
Na função “PublisherInUse”, vai passar a instrução SQL, que irá verificar se o conteúdo que se pretende eliminar existe na tabela Documentos. Esta instrução SQL é passada à função “GetDataTable” da classe DbManager.cs,



```
DbService.cs* x
BiApp.Services.DbService GetAllAut()
1 reference
359 public bool PublisherInUse(int Id)
360 {
361     DataTable dt = DbManager.Instance.GetDataTable
362     ("select a.* from Publisher a inner join Document doc on a.Id = doc.PublisherId where a.Id = " + Id);
363     return dt.Rows.Count > 0;
364 }
```

Figura 299: Eliminar Conteúdos da Tabela Editora, Autor e Tipos de Documento 2, Fonte Própria

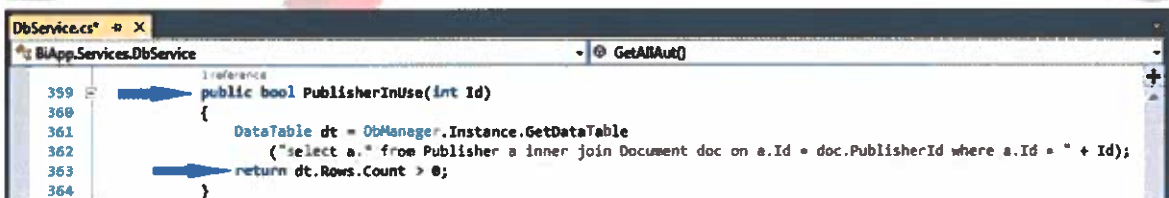
A função “GetDataTable” através do objeto ADO.NET “SqlDataAdapter” executa a instrução SQL recebida da função “PublisherInUse” e constrói uma tabela com o conteúdo recebido da execução da instrução SQL. Essa tabela é devolvida à função “PublisherInUse”. (Microsoft Corporation, n.d.-b)



```
DbManager.cs x
BiApp.DataAccess.DbManager ExecuteCommand(SqlCommand cmd)
17 references
13 public DataTable GetDataTable(string selectStatement)
14 {
15     DataTable dt = new DataTable();
16
17     using (SqlConnection con = new SqlConnection(ConfigHelper.Instance.MyConnectionString))
18     using (SqlDataAdapter adp = new SqlDataAdapter(selectStatement, con))
19     {
20         adp.Fill(dt);
21     }
22     return dt;
23 }
```

Figura 300: Eliminar Conteúdos da Tabela Editora, Autor e Tipos de Documento 3, Fonte Própria

A função “PublisherInUse” é uma função booleana pelo que devolve apenas verdadeiro ou falso. Esta função recebe a tabela enviada pela função “GetDataTable” e verifica o número de linhas da tabela recebida e caso seja superior a zero devolve o valor verdadeiro, no caso do número de linhas ser zero devolve o valor falso.



```
DbService.cs* x
BiApp.Services.DbService GetAllAut()
1 reference
359 public bool PublisherInUse(int Id)
360 {
361     DataTable dt = DbManager.Instance.GetDataTable
362     ("select a.* from Publisher a inner join Document doc on a.Id = doc.PublisherId where a.Id = " + Id);
363     return dt.Rows.Count > 0;
364 }
```

Figura 301: Eliminar Conteúdos da Tabela Editora, Autor e Tipos de Documento 4, Fonte Própria

Voltando de novo à função “BtnDelP_Click”, esta recebe o valor booleano da função “PublisherInUse” e no caso de ter recebido o valor verdadeiro envia uma mensagem ao administrador a indicar que o conteúdo que pretende eliminar está a ser utilizado na tabela

Documentos, no caso de receber o valor falso, irá executar as mesmas ações já referidas anteriormente para a tabela Documentos.



```
339 private void BtnDelP_Click(object sender, RoutedEventArgs e)
340 {
341     try
342     {
343         Publisher pub = (Publisher)gridPubs.SelectedItem;
344         if (DbService.Instance.PublisherInUse(pub.Id))
345         {
346             MessageBox.Show("Esta Editora encontra-se associada a um Documento!\nDeverá desassociá-la do Documento para a eliminar.",
347                 "Erro", MessageBoxButton.OK, MessageBoxImage.Error);
348             return;
349         }
350         else if (MessageBox.Show("Tem a certeza que pretende eliminar a Editora seleccionada?", "Confirmação",
351             MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)
352         {
353             DbService.Instance.DeletePub(pub.Id);
354             Refresh();
355         }
356     }
357 }
```

Figura 302: Eliminar Conteúdos da Tabela Editora, Autor e Tipos de Documento 5, Fonte Própria

4.3.5.3 Editar e Adicionar Conteúdos nas Tabelas

Outras das funcionalidades disponibilizadas aos administradores da aplicação, é a possibilidade de adicionarem e editarem os conteúdos das tabelas na base de dados.

Como já explicado anteriormente neste documento, quando o administrador da aplicação carrega no botão “Adicionar” ou “Editar” irá ser gerada uma nova interface gráfica para o efeito, em que a sua base é a mesma, diferindo apenas o modo como é gerada, se em modo de adicionar, ou se em modo de editar.

4.3.5.3.1 Editar Conteúdos nas Tabelas

Sendo gerada a interface gráfica em modo de editar, é passado o conteúdo da tabela da linha selecionada e os controlos da interface gráfica irão receber os respetivos conteúdos da tabela.



```
188 private void BtnEdit_Click(object sender, RoutedEventArgs e)
189 {
190     try
191     {
192         Document doc = (Document)gridDocs.SelectedItem;
193         DocumentWindow window = new DocumentWindow(EditionMode.Edit, doc);
194         window.ShowDialog();
195         Refresh();
196     }
197 }
```

Figura 303: Editar Conteúdos das Tabelas 1, Fonte Própria

A classe DocumentWindow.xaml.cs, irá receber os conteúdos da tabela referentes à linha selecionada quando foi carregado o botão “Editar”.

```

DocumentWindow.xaml.cs
BIAApp.DocumentWindow
  BtnCancel_Click(object sender, RoutedEventArgs e)
35 public DocumentWindow(EdiçãoMode mode, Document doc = null)
36 {
37     this.Mode = mode;
38     this.Doc = doc;
39
40     this.Authors = DbService.Instance.Authors;
41     this.Publishers = DbService.Instance.Publishers;
42     this.DocumentTypes = DbService.Instance.DocumentTypes;
43
44     if (this.Doc == null)
45     {
46         this.Doc = new Document();
47         this.Doc.AuthorId = this.Authors.FirstOrDefault().Id;
48         this.Doc.PublisherId = this.Publishers.FirstOrDefault().Id;
49         this.Doc.DocumentTypeId = this.DocumentTypes.FirstOrDefault().Id;
50     }
51
52     DataContext = this;
53     this.Authors = DbService.Instance.Authors;
54     this.DocumentTypes = DbService.Instance.DocumentTypes;
55     this.Publishers = DbService.Instance.Publishers;
56
57     InitializeComponent();
58
59     switch (Mode)
60     {
61     case EdiçãoMode.Add:
62         this.Title = "Adicionar Documento";
63         break;
64     case EdiçãoMode.Edit:
65         this.Title = "Editar Documento";
66         break;
67     }

```

Figura 304: Editar Conteúdos das Tabelas 2, Fonte Própria

Os conteúdos de cada campo são passados para o ficheiro de *layout* DocumentWindow.xaml através do *binding* que cada um desses controlos tem para a classe DocumentWindow.xaml.cs.

```

37 <TextBlock Text="Titulo:" Grid.Column="1" Grid.Row="0" VerticalAlignment="Center" .../>
38 <TextBox Name="txtTitle" VerticalContentAlignment="Center" Grid.Column="2" Grid.Row="0" Height="30" Margin="10"
39     Padding="5,5,5,5">
40     <TextBlock Text=
41     <Binding Path="Doc.Title" Mode="TwoWay" UpdateSourceTrigger="PropertyChanged">
42     <Binding.ValidationRules...>
43     </Binding>
44     </TextBlock.Text>
45     </TextBox>
46
47 <TextBlock Text="Tipo de Documento:" Grid.Column="1" Grid.Row="1" VerticalAlignment="Center" HorizontalAlignment="Right" .../>
48 <ComboBox Name="cmbDocumentType" Grid.Column="2" Grid.Row="1" VerticalContentAlignment="Center" Height="30" Margin="10"
49     Padding="5,5,5,5" ItemsSource="{Binding DocumentTypes, Mode=TwoWay, UpdateSourceTrigger=PropertyChanged}"
50     DisplayMemberPath="Description" SelectedValuePath="Id" SelectedValue="{Binding Path=Doc.DocumentTypeId}" />
51
52 <TextBlock Text="Autor:" Grid.Column="1" Grid.Row="2" VerticalAlignment="Center" HorizontalAlignment="Right" .../>
53 <ComboBox Name="cmbAuthor" Grid.Column="2" Grid.Row="2" VerticalContentAlignment="Center" Height="30" Margin="10"
54     Padding="5,5,5,5" ItemsSource="{Binding Authors, Mode=TwoWay, UpdateSourceTrigger=PropertyChanged}"
55     DisplayMemberPath="Name" SelectedValuePath="Id" SelectedValue="{Binding Path=Doc.AuthorId}" />
56
57 <TextBlock Text="Editores:" Grid.Column="1" Grid.Row="3" VerticalAlignment="Center" HorizontalAlignment="Right" .../>
58 <ComboBox Name="cmbPublisher" Grid.Column="2" Grid.Row="3" VerticalContentAlignment="Center" Height="30" Padding="5,5,5,5"
59     Margin="10" ItemsSource="{Binding Publishers, Mode=TwoWay, UpdateSourceTrigger=PropertyChanged}" DisplayMemberPath="Name"
60     SelectedValuePath="Id" SelectedValue="{Binding Path=Doc.PublisherId}" />
61
62 <TextBlock Text="Data:" Grid.Column="1" Grid.Row="4" VerticalAlignment="Center" HorizontalAlignment="Right" Margin="0,17,20,17" .../>
63 <TextBlock Text="Resumo:" Grid.Column="1" Grid.Row="5" VerticalAlignment="Center" HorizontalAlignment="Right" Margin="0,17,20,17" .../>
64 <TextBox x:Name="txtResume" Text="{Binding Doc.Resume}" ScrollViewer.HorizontalScrollBarVisibility="Disabled"
65     ScrollViewer.VerticalScrollBarVisibility="Auto" AcceptsReturn="True" VerticalContentAlignment="Top" Padding="5,5,5,5"
66     TextWrapping="Wrap" Grid.Column="2" Grid.Row="5" Margin="10,10,10,25" Grid.RowSpan="3">
67     <TextBlock.ToolTip...>
68     </TextBlock>
69
70 <TextBlock Text="Capa:" Grid.Row="6" VerticalAlignment="Center" HorizontalAlignment="Right" Margin="0,18,131,17" Height="16" .../>
71 <Image Name="imgCover" Source="{Binding Doc.CoverBytes, NotifyOnSourceUpdated=True, Converter={StaticResource imgConverter}}"
72     Grid.RowSpan="3" Grid.Row="7">
73     <Image.ToolTip...>
74     </Image>
75
76 <TextBlock Text="Documento:" Grid.Row="8" VerticalAlignment="Center" HorizontalAlignment="Right" Margin="0,10,99,24" Height="16" .../>
77

```

Figura 305: Editar Conteúdos das Tabelas 3, Fonte Própria

A classe Documentwindow.xaml.cs irá receber o conteúdo da interface gráfica do ficheiro de *layout* Documentwindow.xaml, através do *binding*, que esses controlos têm nas propriedades.

```
67     }
68
69     TxtTitle.GetBindingExpression(TextBox.TextProperty).UpdateSource();
70     CmbAuthor.GetBindingExpression(ComboBox.ItemsSourceProperty).UpdateSource();
71     CmbDocumentType.GetBindingExpression(ComboBox.ItemsSourceProperty).UpdateSource();
72     CmbPublisher.GetBindingExpression(ComboBox.ItemsSourceProperty).UpdateSource();
73     TxtResume.GetBindingExpression(TextBox.TextProperty).UpdateSource();
74     RefreshBtns();
75 }
```

Figura 306: Editar Conteúdos das Tabelas 4, Fonte Própria

Após os campos preenchidos na interface gráfica, quando o administrador carregar no botão “Gravar”, o mesmo tem associado um evento “Click” que chama a função “BtnSave_Click”.

```
100 <Button x:Name="BtnSave" Content="Gravar" Height="20" Width="75" Margin="5" Click="BtnSave_Click"
101         Style="{DynamicResource StyleButtonSave}"/>
```

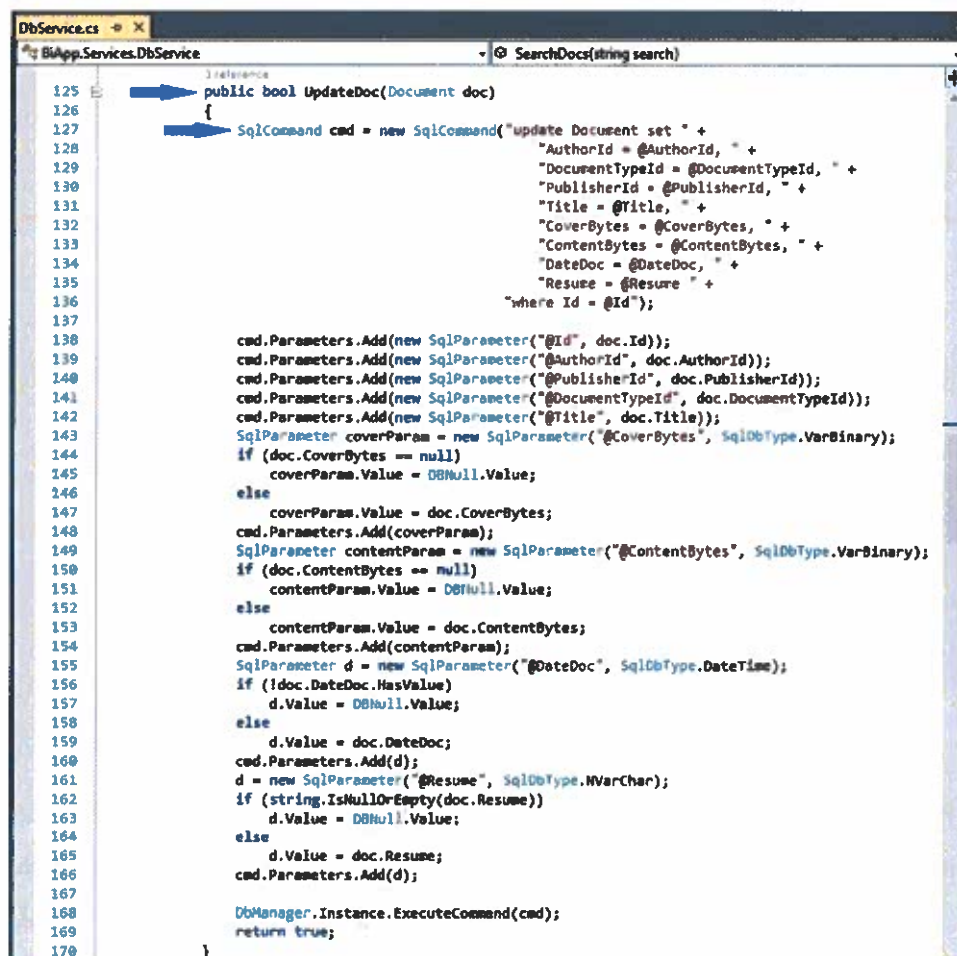
Figura 307: Editar Conteúdos das Tabelas 5, Fonte Própria

A função “BtnSave_Click” verifica qual o modo em que foi iniciada a interface gráfica. Em seguida, após verificar que foi em modo de editar irá passar o conteúdo obtido de cada controlo da interface gráfica, através do *binding*, para a função “UpdateDoc” da classe DbService.cs.

```
83 private void BtnSave_Click(object sender, RoutedEventArgs e)
84 {
85     try
86     {
87         switch (Mode)
88         {
89             case EditionMode.Add:
90                 DbService.Instance.AddDoc(Doc);
91                 DbService.Instance.RefreshReferenceData();
92                 this.DialogResult = true;
93                 break;
94             case EditionMode.Edit:
95                 DbService.Instance.UpdateDoc(Doc);
96                 DbService.Instance.RefreshReferenceData();
97                 this.DialogResult = true;
98                 break;
99         }
100     }
}
```

Figura 308: Editar Conteúdos das Tabelas 6, Fonte Própria

A função “UpdateDoc” irá gerar uma instrução SQL para atualizar a base de dados, passando o conteúdo recebido da função “BtnSave_Click”. Esta instrução SQL é criada no objeto ADO.NET “SqlCommand”, que recebe os valores de cada conteúdo a atualizar na base de dados, através de outro objeto ADO.NET “SqlParameter”. Esta instrução SQL contida no “SqlCommand” é passada à função “ExecuteCommand” da classe DbManager.cs, esta função “ExecuteCommand” já foi explicada anteriormente. (Microsoft Corporation, n.d.-b)



```
125 public bool UpdateDoc(Document doc)
126 {
127     SqlCommand cmd = new SqlCommand("update Document set " +
128         "AuthorId = @AuthorId, " +
129         "DocumentTypeId = @DocumentTypeId, " +
130         "PublisherId = @PublisherId, " +
131         "Title = @Title, " +
132         "CoverBytes = @CoverBytes, " +
133         "ContentBytes = @ContentBytes, " +
134         "DateDoc = @DateDoc, " +
135         "Resume = @Resume " +
136         "where Id = @Id");
137
138     cmd.Parameters.Add(new SqlParameter("@Id", doc.Id));
139     cmd.Parameters.Add(new SqlParameter("@AuthorId", doc.AuthorId));
140     cmd.Parameters.Add(new SqlParameter("@PublisherId", doc.PublisherId));
141     cmd.Parameters.Add(new SqlParameter("@DocumentTypeId", doc.DocumentTypeId));
142     cmd.Parameters.Add(new SqlParameter("@Title", doc.Title));
143     SqlParameter coverParam = new SqlParameter("@CoverBytes", SqlDbType.VarBinary);
144     if (doc.CoverBytes == null)
145         coverParam.Value = DBNull.Value;
146     else
147         coverParam.Value = doc.CoverBytes;
148     cmd.Parameters.Add(coverParam);
149     SqlParameter contentParam = new SqlParameter("@ContentBytes", SqlDbType.VarBinary);
150     if (doc.ContentBytes == null)
151         contentParam.Value = DBNull.Value;
152     else
153         contentParam.Value = doc.ContentBytes;
154     cmd.Parameters.Add(contentParam);
155     SqlParameter d = new SqlParameter("@DateDoc", SqlDbType.DateTime);
156     if (!doc.DateDoc.HasValue)
157         d.Value = DBNull.Value;
158     else
159         d.Value = doc.DateDoc;
160     cmd.Parameters.Add(d);
161     d = new SqlParameter("@Resume", SqlDbType.NVarChar);
162     if (string.IsNullOrEmpty(doc.Resume))
163         d.Value = DBNull.Value;
164     else
165         d.Value = doc.Resume;
166     cmd.Parameters.Add(d);
167
168     DbManager.Instance.ExecuteCommand(cmd);
169     return true;
170 }
```

Figura 309: Editar Conteúdos das Tabelas 7, Fonte Própria

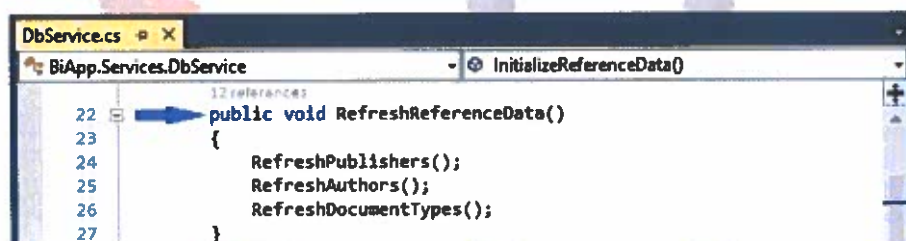
Voltando à função “BtnSave_Click” esta irá chamar agora função “RefreshReferenceData” da classe DbService.cs.



```
83 private void BtnSave_Click(object sender, RoutedEventArgs e)
84 {
85     try
86     {
87         switch (Mode)
88         {
89             case EditionMode.Add:
90                 ObService.Instance.AddDoc(Doc);
91                 DbService.Instance.RefreshReferenceData();
92                 this.DialogResult = true;
93                 break;
94             case EditionMode.Edit:
95                 ObService.Instance.UpdateDoc(Doc);
96                 DbService.Instance.RefreshReferenceData();
97                 this.DialogResult = true;
98                 break;
99         }
100     }
```

Figura 310: Editar Conteúdos das Tabelas 8, Fonte Própria

A função “RefreshReferenceData” irá chamar as funções “RefreshPublishers”, “RefreshAuthors” e “RefreshDocumentType”.



```
22 public void RefreshReferenceData()
23 {
24     RefreshPublishers();
25     RefreshAuthors();
26     RefreshDocumentTypes();
27 }
```

Figura 311: Editar Conteúdos das Tabelas 9, Fonte Própria

As funções “RefreshPublishers”, “RefreshAuthors” e “RefreshDocumentType”, irão todas efetuar as mesmas ações, cada uma na sua tabela correspondente. Vamos aqui exemplificar apenas para uma delas, neste caso a função “RefreshPublishers”. Esta irá criar um novo objeto ADO.NET “DataTable”, que irá receber da função “GetDataTable” da classe DbManager.cs o resultado da instrução “select * from Publisher”, esta instrução lista da base de dados todo o conteúdo da tabela Editoras na base de dados. (Microsoft Corporation, n.d.-b)

É verificado se o objeto “DataTable” não está vazio, em seguida é criada uma nova lista “Publishers” para atualizar o conteúdo da tabela Editoras. É feito um ciclo para correr cada linha da “DataTable” e atualizar a lista “Publishers” através da função “FillPublisher”.

```

225 public void RefreshPublishers()
226 {
227     DataTable dt = DbManager.Instance.GetDataTable("select * from Publisher");
228     if (dt != null && dt.Rows.Count > 0)
229     {
230         Publishers = new List<Publisher>();
231         foreach (DataRow r in dt.Rows)
232             Publishers.Add(FillPublisher(r));
233     }
234 }

```

Figura 312: Editar Conteúdos das Tabelas 10, Fonte Própria

A função “FillPublisher” irá passar o conteúdo coluna da linha da “DataTable”, para cada variável da classe Publisher.cs que corresponde a cada uma das colunas da tabela da aplicação e devolve o seu conteúdo à função “RefreshPublishers”.

```

250 private Publisher FillPublisher(DataRow row)
251 {
252     Publisher pub = new Publisher
253     {
254         Id = Convert.ToInt32(row["Id"]),
255         Name = row["Name"].ToString(),
256         Address = row["Address"].ToString(),
257         Email = row["Email"].ToString(),
258         PhoneNumber = row["PhoneNumber"].ToString()
259     };
260     return pub;
261 }

```

Figura 313: Editar Conteúdos das Tabelas 11, Fonte Própria

Voltando à função “RefreshPublishers”, a mesma recebe o conteúdo das variáveis da classe Publisher.cs, e atualiza o conteúdo da tabela Editoras.

```

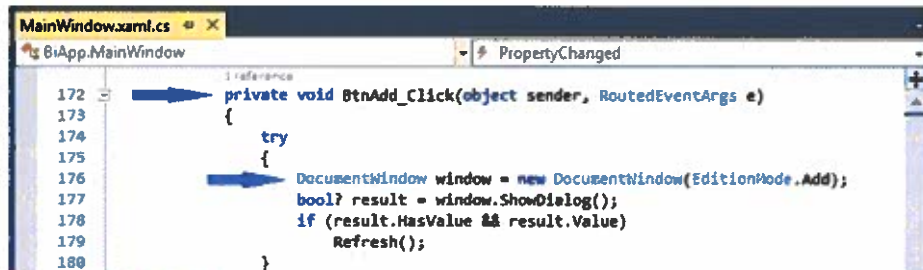
225 public void RefreshPublishers()
226 {
227     DataTable dt = DbManager.Instance.GetDataTable("select * from Publisher");
228     if (dt != null && dt.Rows.Count > 0)
229     {
230         Publishers = new List<Publisher>();
231         foreach (DataRow r in dt.Rows)
232             Publishers.Add(FillPublisher(r));
233     }
234 }

```

Figura 314: Editar Conteúdos das Tabelas 12, Fonte Própria

4.3.5.3.2 Adicionar Conteúdos nas Tabelas

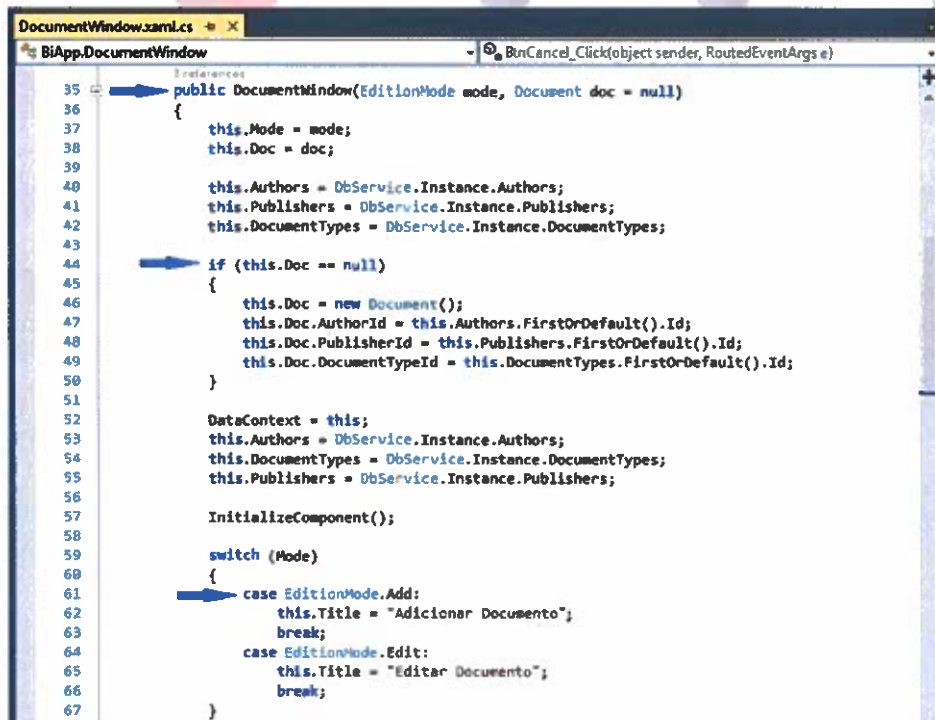
Quando gerada a interface gráfica em modo de adicionar, não é passado nenhum conteúdo para ser gerada a interface gráfica.



```
172 private void BtnAdd_Click(object sender, RoutedEventArgs e)
173 {
174     try
175     {
176         DocumentWindow window = new DocumentWindow(EditionMode.Add);
177         bool? result = window.ShowDialog();
178         if (result.HasValue && result.Value)
179             Refresh();
180     }
181 }
```

Figura 315: Adicionar Conteúdos nas Tabelas 1, Fonte Própria

Na classe DocumentWindow.xaml.cs, uma vez que não foi passado qualquer conteúdo, a mesma irá ser gerada com os controlos do tipo “TextBox” sem qualquer conteúdo para que seja preenchido pelo administrador e os controlos do tipo “ComboBox” irão ter selecionado o primeiro conteúdo da tabela a que pertencem, iremos falar dos controlos do tipo “ComboBox” mais à frente. O controlo do tipo “Image” irá estar também vazio para possibilitar ao administrador fazer o *upload* do documento em PDF e ser apresentada neste campo a capa desse documento.



```
35 public DocumentWindow(EditionMode mode, Document doc = null)
36 {
37     this.Mode = mode;
38     this.Doc = doc;
39
40     this.Authors = DbService.Instance.Authors;
41     this.Publishers = DbService.Instance.Publishers;
42     this.DocumentTypes = DbService.Instance.DocumentTypes;
43
44     if (this.Doc == null)
45     {
46         this.Doc = new Document();
47         this.Doc.AuthorId = this.Authors.FirstOrDefault().Id;
48         this.Doc.PublisherId = this.Publishers.FirstOrDefault().Id;
49         this.Doc.DocumentTypeId = this.DocumentTypes.FirstOrDefault().Id;
50     }
51
52     DataContext = this;
53     this.Authors = DbService.Instance.Authors;
54     this.DocumentTypes = DbService.Instance.DocumentTypes;
55     this.Publishers = DbService.Instance.Publishers;
56
57     InitializeComponent();
58
59     switch (Mode)
60     {
61     case EditionMode.Add:
62         this.Title = "Adicionar Documento";
63         break;
64     case EditionMode.Edit:
65         this.Title = "Editar Documento";
66         break;
67     }
```

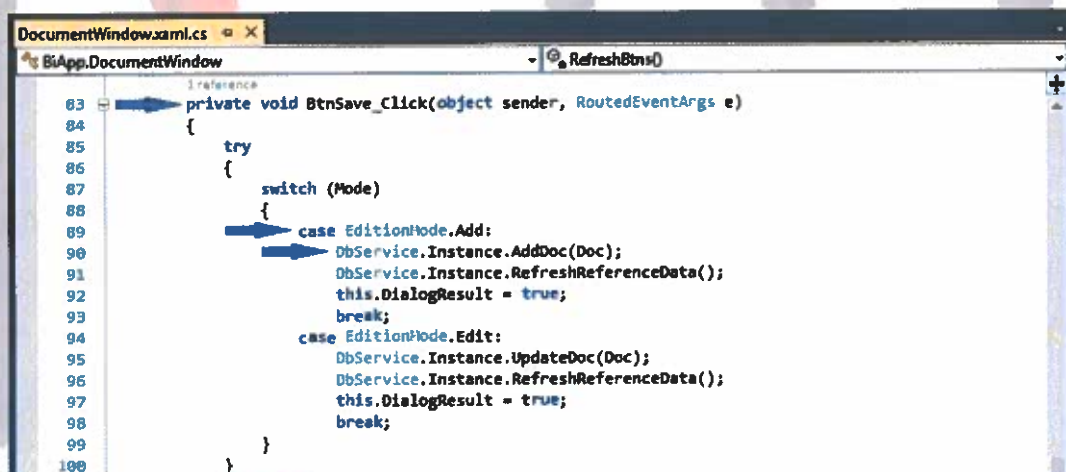
Figura 316: Adicionar Conteúdos nas Tabelas 2, Fonte Própria

Como já referido acima os controlos da interface gráfica do ficheiro de *layout* DocumentWindow.xaml, têm *bindings* em que a propriedade “Mode” está definida com o valor “TwoWay”, deste modo tem uma associação nos dois sentidos entre a classe DocumentWindow.xaml.cs e o ficheiro de *layout* DocumentWindow.xaml.

Deste modo o conteúdo que seja introduzido nos controlos da interface gráfica do ficheiro de *layout* DocumentWindow.xaml irão ser passado para a classe DocumentWindow.xaml.cs, para depois poderem ser atualizados na base de dados.

O botão “Gravar” nesta interface gráfica em modo adicionar, tal como no modo editar, tem associado o evento “Click” que chama a função “BtnSave_Click”.

N função “BtnSave_Click” tal como já referido anteriormente, vai verificar qual o modo em que foi iniciada a interface gráfica. Em seguida, após verificar que foi em modo de adicionar, irá passar o conteúdo obtido de cada controlo da interface gráfica, através do *binding*, para a função “AddDoc” da classe DbService.cs.



```
DocumentWindow.xaml.cs
BiApp.DocumentWindow
RefreshBtns()

private void BtnSave_Click(object sender, RoutedEventArgs e)
{
    try
    {
        switch (Mode)
        {
            case EditionMode.Add:
                DbService.Instance.AddDoc(Doc);
                DbService.Instance.RefreshReferenceData();
                this.DialogResult = true;
                break;
            case EditionMode.Edit:
                DbService.Instance.UpdateDoc(Doc);
                DbService.Instance.RefreshReferenceData();
                this.DialogResult = true;
                break;
        }
    }
}
```

Figura 317: Adicionar Conteúdos nas Tabelas 3, Fonte Própria

A função “AddDoc” irá gerar uma instrução SQL para adicionar o novo conteúdo à base de dados, passando o conteúdo recebido da função “BtnSave_Click”. Esta instrução SQL é criada no objeto ADO.NET “SqlCommand”, que recebe os valores de cada conteúdo a atualizar na base de dados, através de outro objeto ADO.NET “SqlParameter”. Esta instrução SQL contida no “SqlCommand” é passada à função “ExecuteCommand” da classe DbManager.cs, esta função “ExecuteCommand” já foi explicada anteriormente. (Microsoft Corporation, n.d.-b)

```
DbService.cs * x
App.Services.DbService DeleteDoc(int id)
173 public bool AddDoc(Document doc)
174 {
175     SqlCommand cmd = new SqlCommand
176         ("Insert into Document (AuthorId,PublisherId,DocumentTypeId,Title,CoverBytes,ContentBytes,DateDoc,Resume) " +
177          "values(@AuthorId,@PublisherId,@DocumentTypeId,@Title,@CoverBytes,@ContentBytes,@DateDoc,@Resume)");
178     cmd.Parameters.Add(new SqlParameter("@AuthorId", doc.AuthorId));
179     cmd.Parameters.Add(new SqlParameter("@PublisherId", doc.PublisherId));
180     cmd.Parameters.Add(new SqlParameter("@DocumentTypeId", doc.DocumentTypeId));
181     cmd.Parameters.Add(new SqlParameter("@Title", doc.Title));
182     SqlParameter coverParam = new SqlParameter("@CoverBytes", SqlDbType.VarBinary);
183     if (doc.CoverBytes == null)
184         coverParam.Value = DBNull.Value;
185     else
186         coverParam.Value = doc.CoverBytes;
187     cmd.Parameters.Add(coverParam);
188     SqlParameter contentParam = new SqlParameter("@ContentBytes", SqlDbType.VarBinary);
189     if (doc.ContentBytes == null)
190         contentParam.Value = DBNull.Value;
191     else
192         contentParam.Value = doc.ContentBytes;
193     cmd.Parameters.Add(contentParam);
194     SqlParameter d = new SqlParameter("@DateDoc", SqlDbType.DateTime);
195     if (!doc.DateDoc.HasValue)
196         d.Value = DBNull.Value;
197     else
198         d.Value = doc.DateDoc;
199     cmd.Parameters.Add(d);
200     d = new SqlParameter("@Resume", SqlDbType.NVarChar);
201     if (string.IsNullOrEmpty(doc.Resume))
202         d.Value = DBNull.Value;
203     else
204         d.Value = doc.Resume;
205     cmd.Parameters.Add(d);
206     DbManager.Instance.ExecuteCommand(cmd);
207     return true;
208 }
209 }
```

Figura 318: Adicionar Conteúdos nas Tabelas 4, Fonte Própria

Voltando à função “BtnSave_Click” esta irá chamar agora função “RefreshReferenceData” da classe DbService.cs. Todo o restante processo de atualização das tabelas da aplicação, é o mesmo processo da interface gráfica em modo de editar.

4.3.5.4 ComboBox

Foram definidos como obriatórios alguns campos da tabela Documentos. Uns destes campos são as chaves estrangeiras que fazem a relação entre a tabela Documentos e as restantes tabelas. Desta forma, estes campos terão de corresponder exatamente aos que estão inseridos nas tabelas relacionadas, de outra forma quando é atualizada a base de dados irão surgir erros. Para facilitar o administrador da aplicação a evitar estes erros, foram criados controlos do tipo “ComboBox” para estes campos, desta forma apenas poderão ser seleccionadas informações de referência já presentes nas tabelas correspondentes.

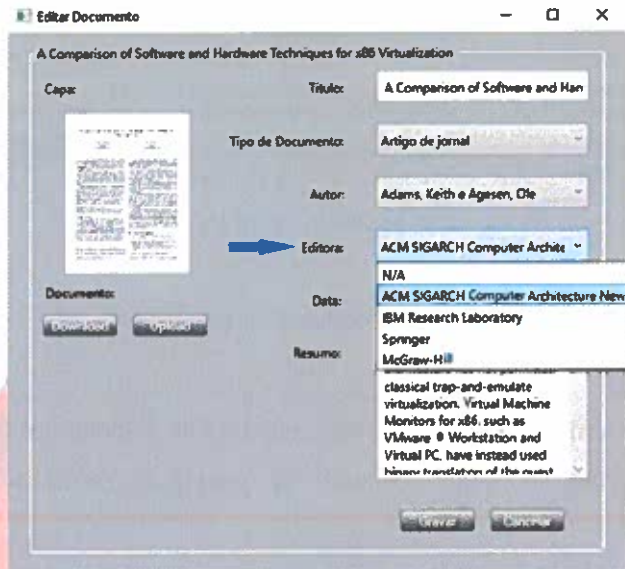


Figura 319: ComboBox 1, Fonte Própria

Nestes controlos foi-lhe definida a propriedade “Binding”. O *binding* é uma associação que permite que este campo receba as propriedades de outro objeto. Vamos exemplificar aqui a “ComboBox” para o campo da Editora. À propriedade “Binding” foi atribuído o valor “Publishers” que será a classe da qual este objeto irá receber as alterações nela realizadas. Na propriedade “Mode” colocámos o valor “TwoWay”, para que a atualização deste controlo seja feita nos dois sentidos. A propriedade “DisplayMemberPath” foi colocado o valor “Name”, assim o conteúdo que será apresentado é o do nome da Editora. Na propriedade “SelectedValuePath” foi atribuído o valor “Id”, pois quando este campo é selecionado para podermos fazer a relação das tabelas, é necessária a chave primária da tabela a que estamos a relacionar a tabela Documentos. Por fim na propriedade “SelectedValue” foi adicionado mais um *binding* neste caso a propriedade “Path” deste *binding* foi atribuído o valor “Doc.PublisherId”, pois é a chave estrangeira da tabela Documentos para a tabela Editoras.

```

62 <ComboBox Base="ComboBox" Grid.Column="2" Grid.Row="3" VerticalContentAlignment="Center" Height="30" Padding="5,5,5,5"
63 Margin="10" ItemsSource="{Binding Publishers, Mode=TwoWay, UpdateSourceTrigger=PropertyChanged}"
64 DisplayMemberPath="Name" SelectedValuePath="Id" SelectedValue="{Binding Path=Doc.PublisherId}"/>

```

Figura 320: ComboBox 2, Fonte Própria

Para que apareçam os nomes de todas as editoras na “ComboBox”, na classe DocumentWindow.xaml.cs, dentro do método construtor DocumentWindow, é feito o *update*.

```

DocumentWindow.xaml.cs
BiApp.DocumentWindow
Mode
68
69
70
71
72
73
    TxtTitle.GetBindingExpression(TextBox.TextProperty).UpdateSource();
    CmbAuthor.GetBindingExpression(ComboBox.ItemsSourceProperty).UpdateSource();
    CmbDocumentType.GetBindingExpression(ComboBox.ItemsSourceProperty).UpdateSource();
    CmbPublisher.GetBindingExpression(ComboBox.ItemsSourceProperty).UpdateSource();
    TxtResume.GetBindingExpression(TextBox.TextProperty).UpdateSource();

```

Figura 321: ComboBox 3, Fonte Própria

4.3.5.5 Upload e Download de Documentos em PDF

4.3.5.5.1 Upload

De modo a que o administrador possa fazer o *upload* de documentos em formato PDF para a base de dados, foi criado o botão “Upload” na interface gráfica do ficheiro de *layout* DocumentWindow.xaml.

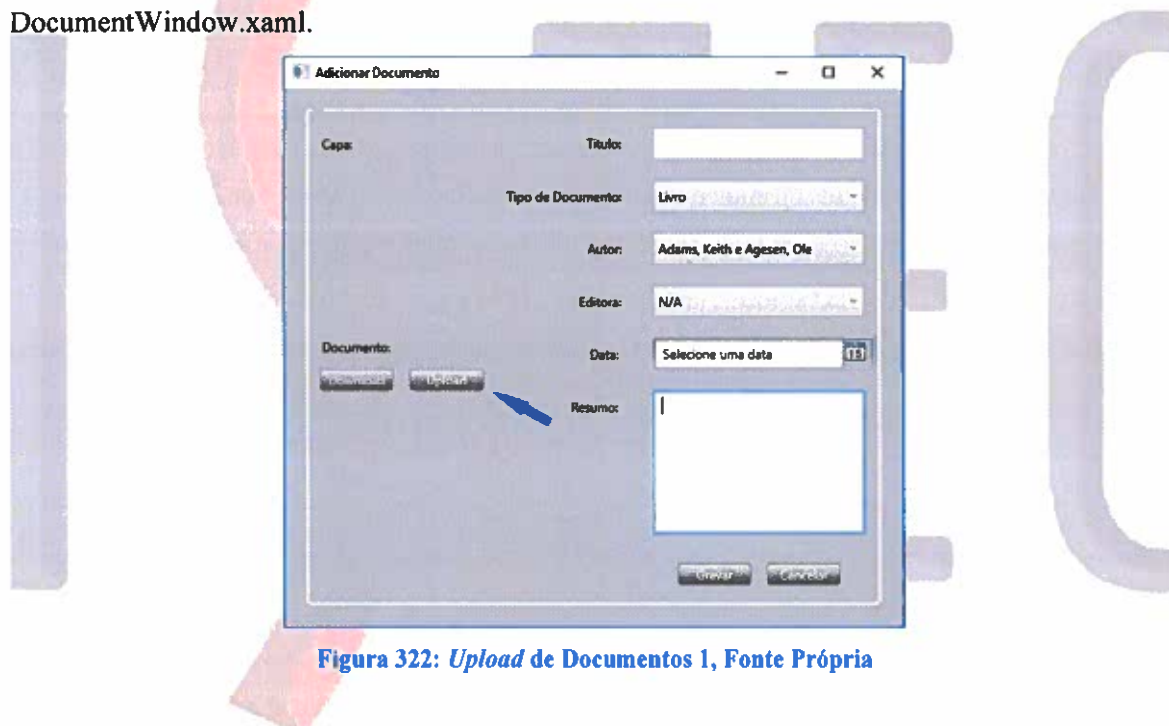


Figura 322: Upload de Documentos 1, Fonte Própria

Este botão tem um evento “Click” associado a ele e que chama a função “BtnUpload_Click”.

```

96 <Button x:Name="BtnUpload" Content="Upload" Height="20" Width="75" Margin="5" Click="BtnUpload_Click"
97       Style="{DynamicResource StyleButton}"/>

```

Figura 323: Upload de Documentos 2, Fonte Própria

A função “BtnUpload_Click” vai gerar uma nova janela para que o administrador possa selecionar a localização do ficheiro que pretende fazer o *upload*.

```
DocumentWindow.xaml.cs
BiApp.DocumentWindow
BtnDownload_Click(object sender, RoutedEventArgs e)

115 private void BtnUpload_Click(object sender, RoutedEventArgs e)
116 {
117     OpenFileDialog op = new OpenFileDialog();
118     op.Title = "Selecione um PDF";
119     op.Filter = "Ficheiros PDF (*.pdf)";
120     if (op.ShowDialog() == true)
121     {
122
123         Doc.CoverBytes = PdfHelper.Instance.GetCover(op.FileName);
124         Doc.ContentBytes = PdfHelper.Instance.GetContent(op.FileName);
125
126         MemoryStream stream = new MemoryStream(Doc.CoverBytes);
127         BitmapImage image = new BitmapImage();
128         image.BeginInit();
129         image.StreamSource = stream;
130         image.EndInit();
131         ImgCover.Source = image;
132
133         RefreshBtns();
134     }
135 }
```

Figura 324: Upload de Documentos 3, Fonte Própria

Aberta a janela o administrador pode seleccionar o ficheiro PDF pretendido e carregar no botão “Open”.

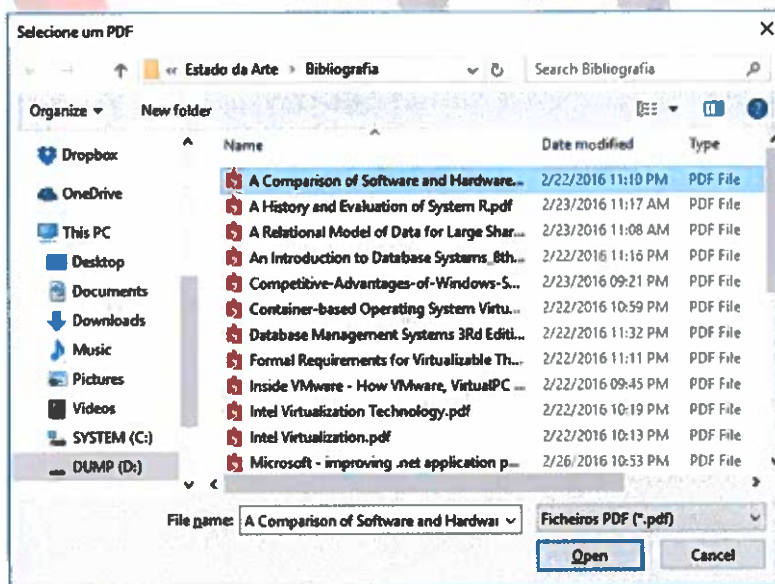


Figura 325: Upload de Documentos 4, Fonte Própria

Após ter sido seleccionado o documento pretendido para *upload* a função “BtnUpload_Click” irá chamar a função “GetCover” da classe PdfHelper.cs.

```
DocumentWindow.xaml.cs
BiApp.DocumentWindow
BtnDownload_Click(object sender, RoutedEventArgs e)

115 private void BtnUpload_Click(object sender, RoutedEventArgs e)
116 {
117     OpenFileDialog op = new OpenFileDialog();
118     op.Title = "Selecione um PDF";
119     op.Filter = "Ficheiros PDF|.pdf";
120     if (op.ShowDialog() == true)
121     {
122         Doc.CoverBytes = PdfHelper.Instance.GetCover(op.FileName);
123         Doc.ContentBytes = PdfHelper.Instance.GetContent(op.FileName);
124
125         MemoryStream stream = new MemoryStream(Doc.CoverBytes);
126         BitmapImage image = new BitmapImage();
127         image.BeginInit();
128         image.StreamSource = stream;
129         image.EndInit();
130         ImgCover.Source = image;
131
132         RefreshBtns();
133     }
134 }
135
```

Figura 326: Upload de Documentos 5, Fonte Própria

A função “GetCover” irá obter a primeira página do documento no formato (.pdf) e converte-a para uma imagem no formato (.Png). Por fim, converte a imagem para um *array* de bytes e passa esse *array* de bytes à função “BtnUpload_Click”.

```
PdfHelper.cs
BiApp.Utilities.PdfHelper
SaveBytesToFile(string filename, byte[] bytesToWrite)

16 public byte[] GetCover(string pdfFilePath)
17 {
18     byte[] result = null;
19
20     string outputPath = Path.GetTempFileName();
21     GhostscriptWrapper.GeneratePageThumb(pdfFilePath, outputPath, 1, 40, 60);
22     using (Image img = Image.FromFile(outputPath))
23     using (MemoryStream ms = new MemoryStream())
24     {
25         img.Save(ms, ImageFormat.Png);
26         result = ms.ToArray();
27     }
28     File.Delete(outputPath);
29
30     return result;
31 }
```

Figura 327: Upload de Documentos 6, Fonte Própria

Voltando à função “BtnUpload_Click” a mesma irá receber o *array* de bytes da capa. Em seguida a mesma irá chamar a função “GetContent” da classe Pdfelper.cs.

```
DocumentWindow.xaml.cs
BiApp.DocumentWindow
- BtnDownload_Click(object sender, RoutedEventArgs e)

115 private void BtnUpload_Click(object sender, RoutedEventArgs e)
116 {
117     OpenFileDialog op = new OpenFileDialog();
118     op.Title = "Selecione um PDF";
119     op.Filter = "Ficheiros PDF|.pdf";
120     if (op.ShowDialog() == true)
121     {
122         Doc.CoverBytes = PdfHelper.Instance.GetCover(op.FileName);
123         Doc.ContentBytes = PdfHelper.Instance.GetContent(op.FileName);
124     }
125     MemoryStream stream = new MemoryStream(Doc.CoverBytes);
126     BitmapImage image = new BitmapImage();
127     image.BeginInit();
128     image.StreamSource = stream;
129     image.EndInit();
130     IagCover.Source = image;
131     RefreshBtns();
132 }
133 }
134 }
135 }
```

Figura 328: Upload de Documentos 7, Fonte Própria

A função “GetContent” recebe o ficheiro em formato (.pdf) e converte o mesmo num *array* de bytes. Esse *array* de bytes é devolvido à função “BtnUpload_Click”.

```
PdfHelper.cs*
BiApp.Utilities.PdfHelper
- SaveBytesToFile(string filename, byte[] bytesToWrite)

33 public byte[] GetContent(string pdfFilePath)
34 {
35     return file.ReadAllBytes(pdfFilePath);
36 }
```

Figura 329: Upload de Documentos 8, Fonte Própria

Voltando à função “BtnUpload_Click” a mesma recebe o *array* de bytes do documento. Depois irá converter o *array* de bytes da capa do documento, numa imagem no formato bitmap e a mesma aparecerá na interface gráfica no campo com o nome “Capa”.

```

DocumentWindow.xaml.cs
private void BtnUpload_Click(object sender, RoutedEventArgs e)
{
    OpenFileDialog op = new OpenFileDialog();
    op.Title = "Selecione um PDF";
    op.Filter = "Ficheiros PDF|.pdf";
    if (op.ShowDialog() == true)
    {
        Doc.CoverBytes = PdfHelper.Instance.GetCover(op.FileName);
        Doc.ContentBytes = PdfHelper.Instance.GetContent(op.FileName);

        MemoryStream stream = new MemoryStream(Doc.CoverBytes);
        BitmapImage image = new BitmapImage();
        image.BeginInit();
        image.StreamSource = stream;
        image.EndInit();
        ImgCover.Source = image;

        RefreshBtns();
    }
}

```

Figura 330: Upload de Documentos 9, Fonte Própria

4.3.5.5.2 Download

Esta funcionalidade está disponível para ambos os perfis de acesso à aplicação. No caso de ser um utilizador terá esta funcionalidade disponível através do botão “Download” no canto superior esquerdo da interface gráfica principal.

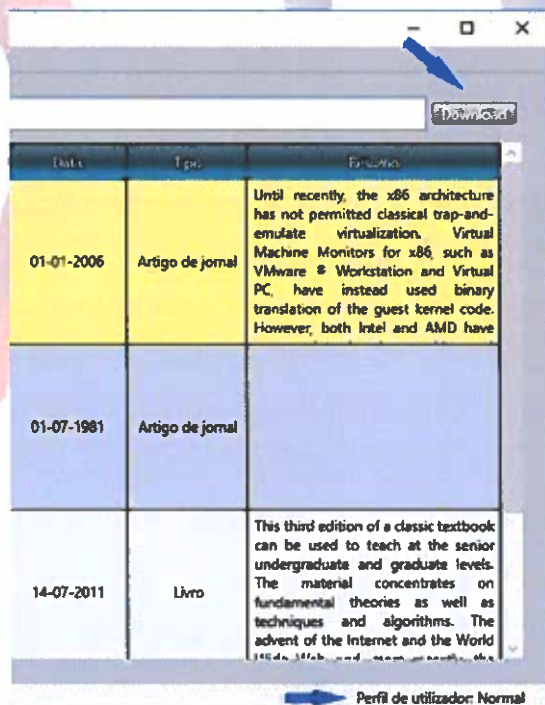


Figura 331: Dwonload de Documentos 1, Fonte Própria

Para os administradores esta funcionalidade está também disponível no mesmo local que está para os utilizadores.

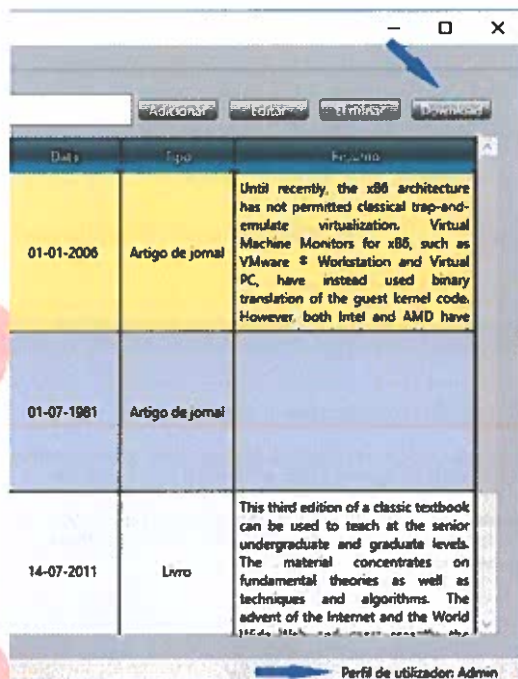


Figura 332: Dwnload de Documentos 2, Fonte Própria

Mas para os administradores esta função está também disponível na interface gráfica do ficheiro de *layout* DocumentWindow.xaml.

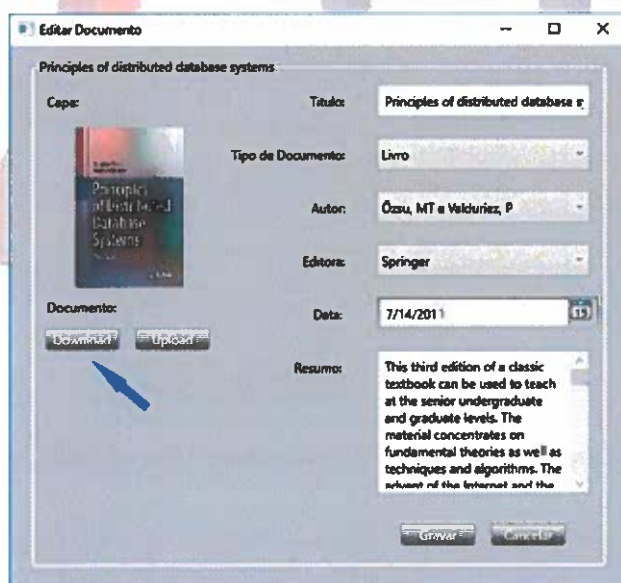


Figura 333: Dwnload de Documentos 3, Fonte Própria

Qualquer um dos botões botão “Download” tem as mesmas características. Estes botões têm associado um evento “Click” que chama a função “BtnDownload_Click”.

```
94 <Button x:Name="BtnDownload" Content="Download" Height="20" Width="75" Margin="5" Click="BtnDownload_Click"
95         IsEnabled="False" Style="{DynamicResource StyleButton}"/>
```

Figura 334: Dwonload de Documentos 4, Fonte Própria

A função “BtnDownload_Click” vai gerar uma nova janela para que possa ser escolhida a localização onde se pretende gravar o documento.

```
DocumentWindow.xaml.cs
BiApp.DocumentWindow
BtnDownload_Click(object sender, RoutedEventArgs e)
137 private void BtnDownload_Click(object sender, RoutedEventArgs e)
138 {
139     System.Windows.Forms.FolderBrowserDialog dialog = new System.Windows.Forms.FolderBrowserDialog();
140     if (dialog.ShowDialog() == System.Windows.Forms.DialogResult.OK)
141     {
142         string filename = System.IO.Path.Combine(dialog.SelectedPath, Doc.Title + ".pdf");
143         PdfHelper.Instance.SaveBytesToFile(filename, Doc.ContentBytes);
144         System.Diagnostics.Process.Start(filename);
145     }
146 }
```

Figura 335: Dwonload de Documentos 5, Fonte Própria

Gerada a janela poderá ser seleccionada a localização onde se pretende gravar o documento em formato (.pdf). Após escolher a localização irá carregar no botão “Ok”.

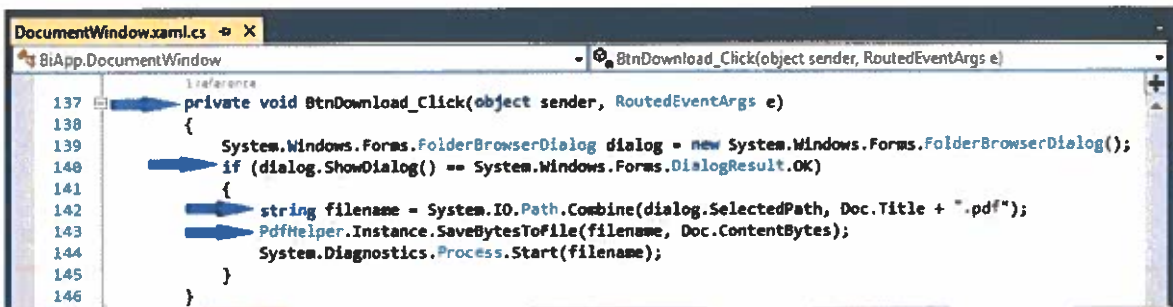


Figura 336: Dwonload de Documentos 6, Fonte Própria

Voltando à função “BtnDownload_Click” é validade que foi premido o botão “Ok” na janela anterior. Em seguida é guardada numa *string* a localização escolhida para gravar o ficheiro

e o nome gerado com que ficará gravado o documento. O nome do documento é gerado através do nome que está no campo título da tabela Documentos e dada a extensão do ficheiro (.pdf).

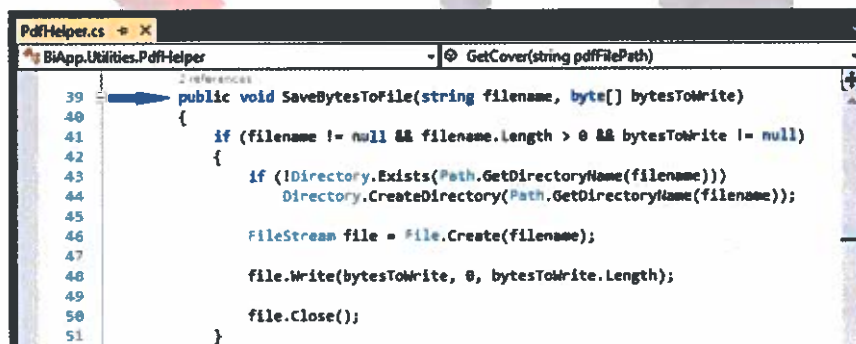
Em seguida é chamada a função “SaveBytesToFile” da classe PdfHelper.cs e é passada a *string* com a localização escolhida e o nome dado ao documento, bem como o próprio *array* de bytes com o documento.



```
DocumentWindow.xaml.cs
BiApp.DocumentWindow
- BtnDownload_Click(object sender, RoutedEventArgs e)
1 reference
137 private void BtnDownload_Click(object sender, RoutedEventArgs e)
138 {
139     System.Windows.Forms.FolderBrowserDialog dialog = new System.Windows.Forms.FolderBrowserDialog();
140     if (dialog.ShowDialog() == System.Windows.Forms.DialogResult.OK)
141     {
142         string filename = System.IO.Path.Combine(dialog.SelectedPath, Doc.Title + ".pdf");
143         PdfHelper.Instance.SaveBytesToFile(filename, Doc.ContentBytes);
144         System.Diagnostics.Process.Start(filename);
145     }
146 }
```

Figura 337: Dwonload de Documentos 7, Fonte Própria

A função “SaveBytesToFile” irá verificar se a *string* que recebeu não está vazia. Em seguida vai confirmar que a localização que lhe foi passada dentro da *string* existe. Por fim, vai converter o *array* de bytes, correspondente ao documento, para um ficheiro no formato (.pdf).



```
PdfHelper.cs
BiApp.Utilities.PdfHelper
- GetCover(string pdfFilePath)
2 references
39 public void SaveBytesToFile(string filename, byte[] bytesToWrite)
40 {
41     if (filename != null && filename.Length > 0 && bytesToWrite != null)
42     {
43         if (!Directory.Exists(Path.GetDirectoryName(filename)))
44             Directory.CreateDirectory(Path.GetDirectoryName(filename));
45
46         FileStream file = File.Create(filename);
47
48         file.Write(bytesToWrite, 0, bytesToWrite.Length);
49         file.Close();
50     }
51 }
```

Figura 338: Dwonload de Documentos 8, Fonte Própria

Por fim, de novo na função “BtnDwonload_Click” é feita a gravação do documento em formato (.pdf) na localização selecionada.

```

DocumentWindow.xaml.cs
BiApp.DocumentWindow
- BtnDownload_Click(object sender, RoutedEventArgs e)
137 private void BtnDownload_Click(object sender, RoutedEventArgs e)
138 {
139     System.Windows.Forms.FolderBrowserDialog dialog = new System.Windows.Forms.FolderBrowserDialog();
140     if (dialog.ShowDialog() == System.Windows.Forms.DialogResult.OK)
141     {
142         string filename = System.IO.Path.Combine(dialog.SelectedPath, Doc.Title + ".pdf");
143         PdfHelper.Instance.SaveBytesToFile(filename, Doc.ContentBytes);
144         System.Diagnostics.Process.Start(filename);
145     }
146 }

```

Figura 339: Dwonload de Documentos 9, Fonte Própria

4.3.5.6 Validação de Campos Obrigatórios

Uma vez que foram definidos campos obrigatórios na base de dados, é necessário na própria aplicação assegurar que esses campos são preenchidos de forma a evitar erros quando for atualizada a base de dados.

Nas interfaces gráficas que contêm esses campos obrigatórios, foi criado um *binding* nos controlos para a classe de onde irão receber as suas propriedades, neste caso recebe as propriedades da classe “RequiredField”, que está a herdar os atributos da classe “ValidationRules”. Enquanto a função “Validationresult” dentro da classe “RequiredField”, não devolver o valor “false”. Se o administrador tentar por exemplo gravar irá receber uma mensagem de erro, indicando que o referido campo é obrigatório.

```

40 <TextBox Name="TxtTitle" VerticalContentAlignment="Center" Grid.Column="2" Grid.Row="0" Height="30" Margin="10"
41     Padding="5,5,5,5">
42     <TextBox.Text>
43     <Binding Path="Doc.Title" Mode="TwoWay" UpdateSourceTrigger="PropertyChanged">
44     <Binding.ValidationRules>
45     <validators:RequiredField ErrorMessage="O campo Título é obrigatório." />
46     </Binding.ValidationRules>
47     </Binding>
48     </TextBox.Text>
49 </TextBox>

```

Figura 340: Validação de Campos Obrigatórios 1, Fonte Própria

A função “ValidationResult” da classe “RequiredField” irá validar que o controlo com o qual está estabelecido o *binding*, não está vazio. E irá despoletar a mensagem referida anteriormente.

```

RequiredField.cs
BiApp.Validators.RequiredField Validate(object value, CultureInfo cultureInfo)
0 references
12 public class RequiredField : ValidationRule
13 {
14     private String _errorMessage = String.Empty;
15     public string ErrorMessage
16     {
17         get { return _errorMessage; }
18         set { _errorMessage = value; }
19     }
20
21     public override ValidationResult Validate(object value, CultureInfo cultureInfo)
22     {
23         var str = value as string;
24
25         if (String.IsNullOrEmpty(str))
26         {
27             return new ValidationResult(false, this.ErrorMessage);
28         }
29
30         return new ValidationResult(true, null);
31     }
32 }

```

Figura 341: Validação de Campos Obrigatórios 2, Fonte Própria

No ficheiro de *layout* App.xaml foi criado um efeito gráfico ao controlo do tipo “TextBox” que tenha a propriedade “Validation.ErrorTemplate”. Este efeito estará ativo até que a função “ValidationResult” devolva o valor “true”.

```

App.xaml
249 <Style TargetType="{x:Type TextBox}">
250     <Setter Property="Validation.ErrorTemplate">
251         <Setter.Value>
252             <ControlTemplate>
253                 <DockPanel LastChildFill="True">
254                     <TextBlock DockPanel.Dock="Right"
255                         Foreground="Red"
256                         Margin="5"
257                         FontSize="12pt"
258                         Text="" />
259                     <Border BorderBrush="Red" BorderThickness="1">
260                         <AdornedElementPlaceholder Name="MyAdorner" />
261                     </Border>
262                 </DockPanel>
263             </ControlTemplate>
264         </Setter.Value>
265     </Setter>
266     <Style.Triggers>
267         <Trigger Property="Validation.HasError" Value="true">
268             <Setter Property="ToolTip"
269                 Value="{Binding RelativeSource={RelativeSource Self},
270                     Path=(Validation.Errors)[0].ErrorContent}"/>
271         </Trigger>
272     </Style.Triggers>
273 </Style>

```

Figura 342: Validação de Campos Obrigatórios 3, Fonte Própria

Aqui podemos verificar o efeito gráfico produzido no controlo, até que os critérios de validação sejam preenchidos, neste caso que o campo não esteja vazio.

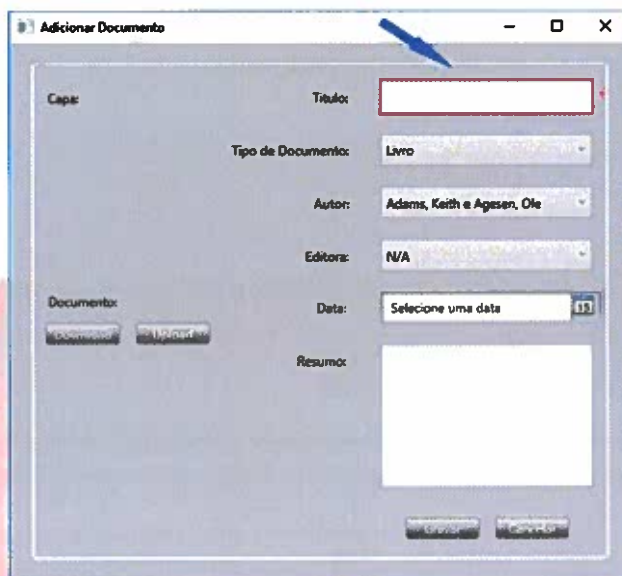


Figura 343: Validação de Campos Obrigatórios 4, Fonte Própria

4.3.5.7 Validação do Campo E-mail

Por forma a validar o campo e-mail e garantirmos que o e-mail introduzido reúne os requisitos, foi criada uma classe “EmailValidator” para esse efeito.

Tal como na validação dos campos obrigatórios foi criado um *binding* no controlo onde será introduzido o e-mail, para a classe de onde irá receber as suas propriedades, neste caso recebe as propriedades da classe “EmailValidator”, que está a herdar os atributos da classe “ValidationRules”. Enquanto a função “ValidationResult” dentro da classe “EmailValidator”, não devolver o valor “false”. Se o administrador tentar por exemplo gravar irá receber uma mensagem de erro, indicando que o e-mail introduzido não é válido.

```

51 <TextBox Name="TxtEmail" VerticalContentAlignment="Center" Grid.Column="1" Grid.Row="2" Margin="10,10,10,10"
52     Padding="5,5,5,5" Height="30">
53     <TextBox.Text>
54         <Binding Path="Pub.Email" Mode="TwoWay" UpdateSourceTrigger="PropertyChanged">
55             <Binding.ValidationRules>
56                 <validators:EmailValidator ErrorMessage="O E-mail introduzido é inválido," />
57             </Binding.ValidationRules>
58         </Binding>
59     </TextBox.Text>
60 </TextBox>

```

Figura 344: Validação de E-mail 1, Fonte Própria

Na função “ValidationResult” da classe “EmailValidator” irá ser validado que o e-mail introduzido reúne os requisitos. Para as validações de e-mail é utilizada uma expressão regular (regex), que irá validar se o formato do e-mail introduzido reúne os requisitos



```
13 public class EmailValidator : ValidationRule
14 {
15     private string _errorMessage = String.Empty;
16     public string ErrorMessage
17     {
18         get { return _errorMessage; }
19         set { _errorMessage = value; }
20     }
21
22     public override ValidationResult Validate(object value, CultureInfo cultureInfo)
23     {
24         var strmail = value as string;
25
26         if (!String.IsNullOrEmpty(strmail))
27         {
28             strmail = strmail.ToLower();
29
30             Regex rg = new Regex(@"^([\w\.\-]+\@)?((([a-z0-9]([a-z0-9]{0,31})?)|([a-z0-9]{1,3})))\.([a-z0-9]+(\.[a-z0-9]+)*)$", RegexOptions.IgnoreCase);
31
32             if (!rg.IsMatch(strmail))
33             {
34                 return new ValidationResult(false, this.ErrorMessage);
35             }
36
37             return new ValidationResult(true, null);
38         }
39     }
40 }
```

Figura 345: Validação de E-mail 2, Fonte Própria

4.3.5.8 Encriptação

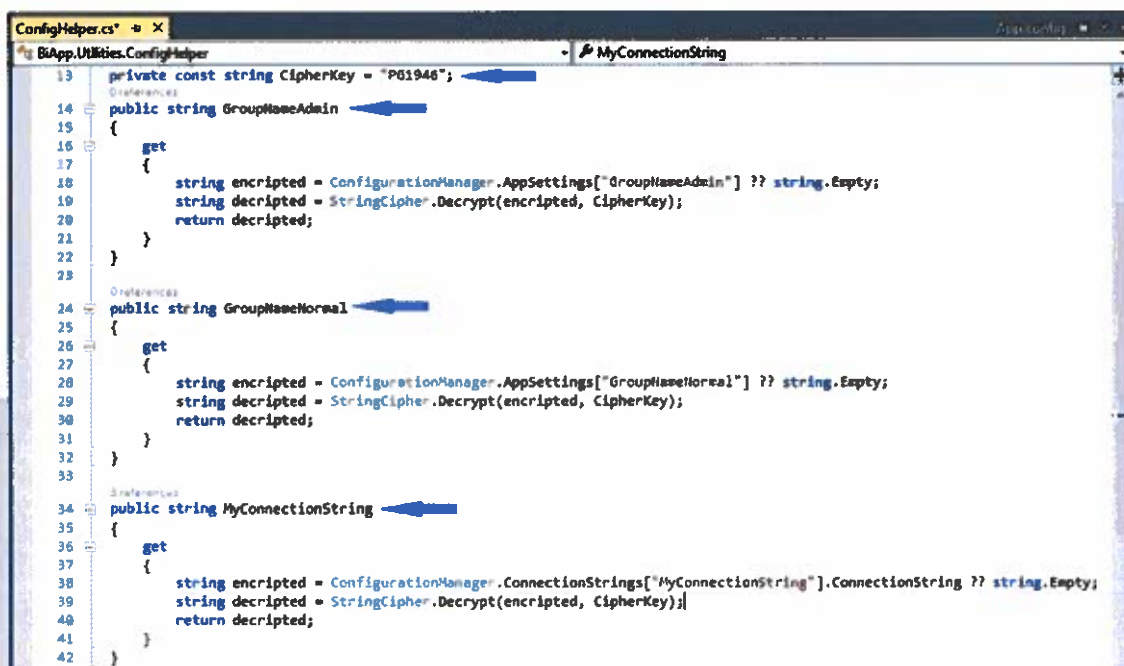
Como foi referido, nas seções de autenticação na aplicação através da *Active Directory* e Estabelecer Ligação da Aplicação à Base de Dados, foi utilizada uma função com um algoritmo de encriptação de 256 Bits, para que a “ConnectionString” e os grupos de segurança que têm acesso à aplicação fossem encriptados no ficheiro App.config.

No caso da validação do acesso à aplicação, para que sejam validados os grupos de segurança que têm acesso à aplicação, esses grupos que foram definidos e encriptados no ficheiro App.config, terão de ser desencriptados dentro já da aplicação antes dessa validação.

Para que a aplicação possa estabelecer a ligação à base de dados é necessário que através “ConnectionString” definida no ficheiro App.config, é também necessário que esta seja desencriptada primeiro antes de ser estabelecida a ligação.

A encriptação e desencriptação do conteúdo destes parâmetros é feita através de uma chave privada que foi definida dentro da aplicação, mas fora do ficheiro App.config. A chave privada foi definida na classe ConfigHelper.cs.

Na classe ConfigHelper.cs em cada função é recebida a *string* encriptada do ficheiro App.config. Em seguida chama a função “Decrypt” da classe StringCipher.cs, passando-lhe a *string* encriptada, e a função “Decrypt” devolve a *string* já desencriptada.



```
13 private const string CipherKey = "P61946";
14 public string GroupNameAdmin
15 {
16     get
17     {
18         string encrypted = ConfigurationManager.AppSettings["GroupNameAdmin"] ?? string.Empty;
19         string decrypted = StringCipher.Decrypt(encrypted, CipherKey);
20         return decrypted;
21     }
22 }
23
24 public string GroupNameNormal
25 {
26     get
27     {
28         string encrypted = ConfigurationManager.AppSettings["GroupNameNormal"] ?? string.Empty;
29         string decrypted = StringCipher.Decrypt(encrypted, CipherKey);
30         return decrypted;
31     }
32 }
33
34 public string MyConnectionString
35 {
36     get
37     {
38         string encrypted = ConfigurationManager.ConnectionStrings["MyConnectionString"].ConnectionString ?? string.Empty;
39         string decrypted = StringCipher.Decrypt(encrypted, CipherKey);
40         return decrypted;
41     }
42 }
```

Figura 346: Encriptação, Fonte Própria

4.3.5.9 Informações para o Utilizador na Interface Gráfica Principal

4.3.5.9.1 Informações sobre a Editora e Autor

O utilizador apenas terá acesso à Tab Documentos da interface principal, que lhe disponibilizará apenas os conteúdos da tabela Documentos. Na tabela Documentos a única informação que o utilizador conseguirá visualizar das restantes tabelas é o nome das Editoras, o nome dos Autores e os Tipos de Documento.

Para que tenha acesso à restante informação das outras tabelas sem ter acesso às mesmas, criámos uma funcionalidade no ficheiro de *layout* da interface principal da Tab Documentos, chamada “ToolTip”.

Foi criada uma “ToolTip” nos campos “Editora” e “Autor” da tabela Documentos, que irá disponibilizar ao utilizador a informação de todos os conteúdos das Tabelas “Editoras” e

“Autores”, respetivamente. Esta informação surgirá sempre que o utilizador passe o cursor do rato em cima de um destes campos.

```

126 <DataGridTemplateColumn Header="Editora" Width="100">
127   <DataGridTemplateColumn.CellTemplate>
128     <DataTemplate>
129       <TextBlock Text="{Binding Publisher.Name}" TextWrapping="Wrap" TextAlignment="Justify" Padding="5,5,5,5">
130         <TextBlock.ToolTip>
131           <StackPanel>
132             <TextBlock>
133               <Run Text="Nome: " FontWeight="Bold" />
134               <Run Text="{Binding Publisher.Name}" />
135             </TextBlock>
136             <TextBlock TextWrapping="Wrap">
137               <Run Text="Morada: " FontWeight="Bold" />
138               <Run Text="{Binding Publisher.Address}" />
139             </TextBlock>
140             <TextBlock>
141               <Run Text="E-mail: " FontWeight="Bold" />
142               <Run Text="{Binding Publisher.Email}" />
143             </TextBlock>
144             <TextBlock>
145               <Run Text="Contacto: " FontWeight="Bold" />
146               <Run Text="{Binding Publisher.PhoneNumber}" />
147             </TextBlock>
148           </StackPanel>
149         </TextBlock.ToolTip>
150       </TextBlock>
151     </DataTemplate>
152   </DataGridTemplateColumn.CellTemplate>
153 </DataGridTemplateColumn>

```

Figura 347: ToolTip Editora/Autor na Interface Principal 1, Fonte Própria

Na interface gráfica o utilizador terá a seguinte visualização desta informação.

	Editora	Data	
n.	ACM SIGARCH Computer Architecture News	01-01-2006	Artigo
W S - h)	IBM Research Laboratory	01-07-1981	Artigo
	Nome: IBM Research Laboratory Morada: Austin, Texas, U.S.A. E-mail: ews@us.ibm.com Contacto: 914 499 1900		
P	Springer	14-07-2011	Li

Figura 348: ToolTip Editora/Autor na Interface Principal 2, Fonte Própria

4.3.5.9.2 Ampliação da Capa e Resumo

A imagem da capa dos documentos na tabela é demasiado pequena para o utilizador a visualizar sem efetuar o *download* do documento. O campo resumo também não permite a visualização completa do resumo do documento. Utilizando a mesma funcionalidade do “ToolTip” no campo da capa e do resumo da tabela Documentos, iremos ampliar a imagem da capa do documento e ampliar o campo resumo para que possa ser visualizado todo o seu conteúdo.

```
81 <DataGridTemplateColumn Header="Capa" Width="100">
82   <DataGridTemplateColumn.CellTemplate>
83     <DataTemplate>
84       <Image Width="100" Source="{Binding CoverBytes, Converter={StaticResource imgConverter}}">
85         <Image.ToolTip>
86           <StackPanel>
87             <Image Source="{Binding CoverBytes, Converter={StaticResource imgConverter}}" Width="300"/>
88           </StackPanel>
89         </Image.ToolTip>
90       </Image>
91     </DataTemplate>
92   </DataGridTemplateColumn.CellTemplate>
93 </DataGridTemplateColumn>
```

Figura 349: Ampliação da Capa na Interface Gráfica Principal 1, Fonte Própria

O utilizador terá a seguinte visualização na interface gráfica.

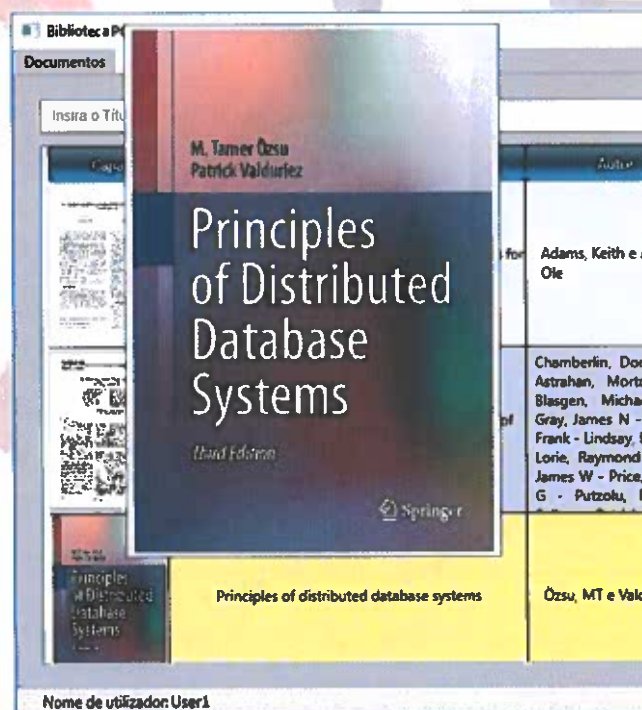


Figura 350: Ampliação da Capa na Interface Gráfica Principal 2, Fonte Própria

4.3.5.10 Criar Ficheiro de Instalação da Aplicação

Para disponibilizarmos a aplicação em todas as máquinas cliente que pretendemos criar no SCVMM, iremos despoletar a instalação da aplicação através de uma Group Policy Object.

As Group Policy Objects apenas permitem despoletar a instalação de aplicações através de ficheiro (.msi), para tal iremos criar um ficheiro de instalação da aplicação da Biblioteca Virtual.

Adicionámos o InstallShield Limited Edition ao Visual Studio onde foi desenvolvida a aplicação e através do “Project Assistant” no Visual Studio criámos o ficheiro de instalação (.msi).



Figura 351: Criar Ficheiro de Instalação da Aplicação 1, Fonte Própria

Na secção “Application Information”, temos a possibilidade de indicar o nome da aplicação, à qual chamámos BiApp.

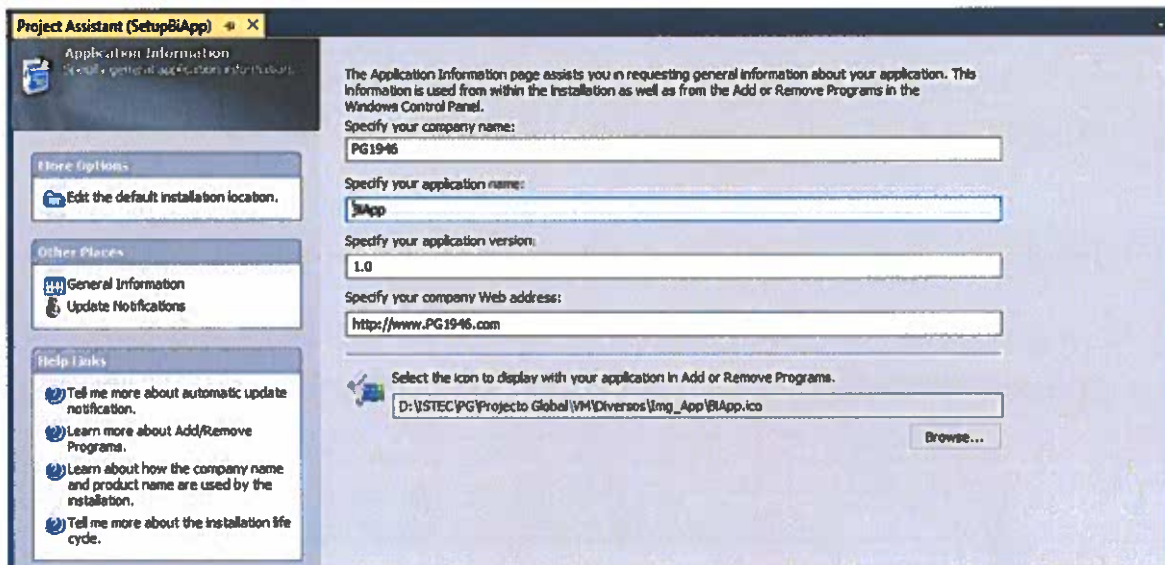


Figura 352: Criar Ficheiro de Instalação da Aplicação 2, Fonte Própria

Passando à secção “Application Files”, indicamos quais os ficheiros necessários para a instalação da aplicação e a localização onde a mesma irá instalada.

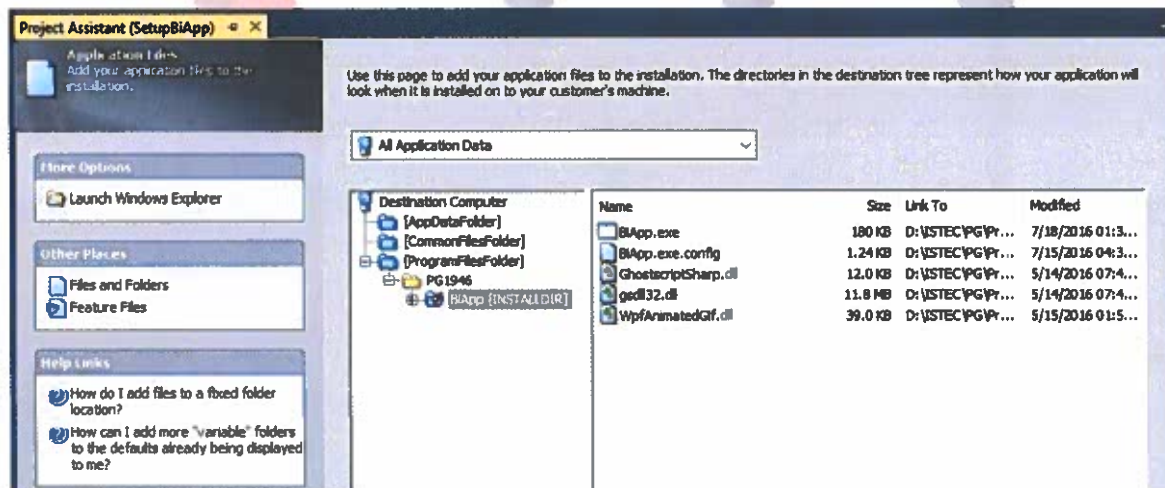


Figura 353: Criar Ficheiro de Instalação da Aplicação 3, Fonte Própria

Chegando à secção “Application Shortcuts”, iremos definir o *icon* da nossa aplicação e quais os atalhos que pretendemos criar.

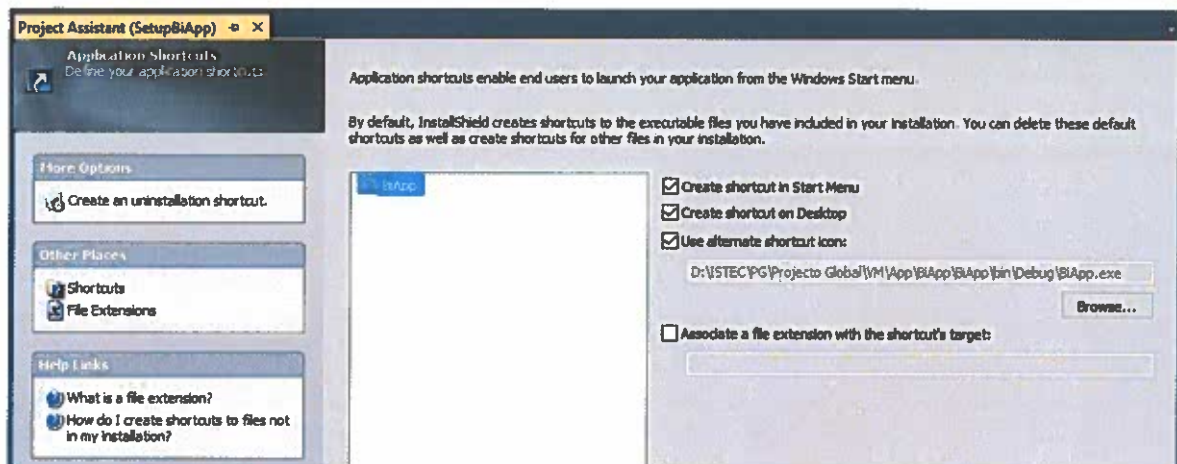
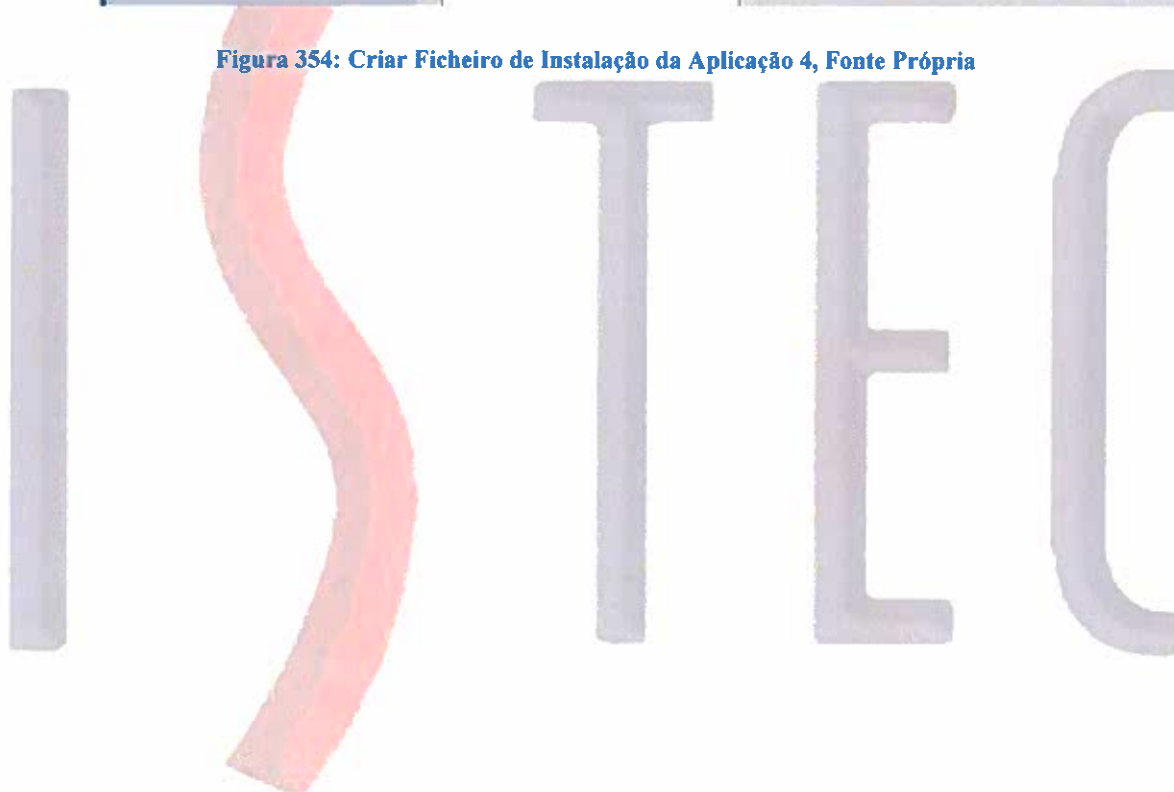


Figura 354: Criar Ficheiro de Instalação da Aplicação 4, Fonte Própria



I S T E C

(Página intencionalmente deixada em branco)

5 Conclusão

Podemos afirmar pela realização deste projeto, que acabámos por demonstrar uma proposta de uma solução concreta e implementável de uma infraestrutura baseada em virtualização, que garante e facilita a manutenção, bem como a redução de custos, tendo em conta o crescimento verificado normalmente nas infraestruturas, face ao surgimento de novos sistemas para suporte de negócio e que quase sempre exigem a aquisição de novo *hardware*. Por isso, poderemos dizer que esta infraestrutura virtualizada identifica-se como um elemento de competitividade no mundo de negócio de uma Biblioteca, pois permite o alcance de forma muito facilitada e sofisticada dos seus objetivos relativamente ao conjunto de tecnologias para o suporte do negócio.

Este trabalho traz também, a possibilidade de mostrarmos as tecnologias existentes hoje e como funcionam, quanto ao *Cloud Computing* e que têm como suporte a virtualização.

Contudo a experiência obtida durante a realização deste projeto mostrou que a implementação de uma infraestrutura virtualizada é um grande desafio, pois por vezes pode culminar em fracassos, não foi o caso, mas pode-se afirmar que tudo foi feito de modo a que no final fossem obtidos os resultados inicialmente esperados.

Uma vez que este projeto se enquadrou no contexto de um cenário concreto, em que as tecnologias utilizadas foram tidas como novas, exigiu assim um nível de aprendizagem mais elevado, que não deixou de influenciar no tempo total da realização deste projeto.

ISTEC

(Página intencionalmente deixada em branco)

Bibliografia

- Advantages, C., & Server, W. (2013). Why Hyper-V? Competitive Advantages of Windows Server 2012 R2 Hyper-V over VMware vSphere 5.5, (October), 1–44.
- Burleson, D. K. (n.d.). Vertical vs. Horizontal scalability for Oracle. Retrieved February 23, 2016, from http://www.dba-oracle.com/t_Vertical_vs_Horizontal_scalability_Oracle.htm
- Chamberlin, D. D., Astrahan, M. M., Blasgen, M. W., Gray, J. N., King, W. F., Lindsay, B. G., ... Yost, R. A. (1981). COMPUTING PRACTICES A History and Evaluation of System R. *IBM Research Laboratory*, 24(10), 632–646.
- Chamberlin, D. D., & Boyce, R. F. (1974). SEQUEL: A STRUCTURED ENGLISH QUERY LANGUAGE. *IBM Research Laboratory*, 249–264.
- Codd, E. F. (1970). A Relational Model of Data for Large Shared Data Banks. *IBM Research Laboratory*, 13(6), 377–387.
- Corporation, M. (n.d.). Provedores de dados .NET Framework. Retrieved February 23, 2016, from [https://msdn.microsoft.com/pt-br/library/a6cd7c08\(v=vs.110\).aspx](https://msdn.microsoft.com/pt-br/library/a6cd7c08(v=vs.110).aspx)
- Date, C. J. (2004). An Introduction to Database Systems_ 8th Edition_C J Date_4.pdf. Pearson.
- Davis, D. (2013). How to Download for Hyper-V Server 2012 Free Edition. Retrieved February 23, 2016, from <http://www.virtualizationsoftware.com/download-for-hyper-v-server-2012-free/>
- Goldberg, R. P. (1974). Survey of virtual machine research. *Computer*. <http://doi.org/10.1109/MC.1974.6323581>
- Gray, J. (1981). The Transaction Concept: Virtues and Limitations. *Proceedings of the 7th International Conference on Very Large Data Bases*, (1), 144–154. <http://doi.org/10.1.1.59.5051>
- Jason, & Basrai, J. (2015). Virtzone - The Difference Between a “Type 2” Hypervisor and a “Type 1” Hypervisor. Retrieved February 23, 2016, from <http://www.virtzone.net/the-difference-between-a-type-2-hypervisor-and-a-type-1-hypervisor/>
- Leavitt, N. (2010). Will NoSql live to Their Promise?, 12–14.
- Mariani, R., Meier, J. D., Vasireddy, S., Babbar, A., & Mackman, A. (2004). *Improving .NET Application Performance and Scalability*. Microsoft Corporation, *Patterns and Practices* (Vol. 649527). Retrieved from [http://msdn.microsoft.com/en-us/library/ff649527\(d=printer\).aspx](http://msdn.microsoft.com/en-us/library/ff649527(d=printer).aspx)

Meier, J. D., Vasireddy, S., Babbar, A., & Mackman, A. (2004). Chapter 12 - Improving ADO.NET Performance. Retrieved February 23, 2016, from <https://msdn.microsoft.com/en-us/library/ff647768.aspx>

Microsoft Corporation. (n.d.-a). ADO.NET Overview. Retrieved January 3, 2016, from [https://msdn.microsoft.com/en-us/library/h43ks021\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/h43ks021(v=vs.110).aspx)

Microsoft Corporation. (n.d.-b). ADO.NET Overview. Retrieved from [https://msdn.microsoft.com/en-us/library/h43ks021\(v=vs.100\).aspx](https://msdn.microsoft.com/en-us/library/h43ks021(v=vs.100).aspx)

Microsoft Corporation. (n.d.-c). Descrição geral do Hyper-V. Retrieved February 23, 2016, from <https://technet.microsoft.com/pt-pt/library/hh831531.aspx>

Microsoft Corporation. (n.d.-d). Hyper-V Architecture. Retrieved February 23, 2016, from [https://msdn.microsoft.com/en-us/library/cc768520\(v=bts.10\).aspx](https://msdn.microsoft.com/en-us/library/cc768520(v=bts.10).aspx)

Microsoft Corporation. (n.d.-e). Overview of the .NET Framework. Retrieved January 16, 2016, from [https://msdn.microsoft.com/pt-pt/en-en/library/zw4w595w\(v=vs.110\).aspx](https://msdn.microsoft.com/pt-pt/en-en/library/zw4w595w(v=vs.110).aspx)

Microsoft Corporation. (n.d.-f). SqlConnection.ConnectionString Property. Retrieved from [https://msdn.microsoft.com/en-us/library/system.data.sqlclient.sqlconnection.connectionstring\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.data.sqlclient.sqlconnection.connectionstring(v=vs.110).aspx)

Microsoft Corporation. (n.d.-g). System Requirements: VMM Management Server in System Center 2012 and in System Center 2012 SP1. Retrieved from [https://technet.microsoft.com/en-us/library/gg610562\(v=sc.12\).aspx](https://technet.microsoft.com/en-us/library/gg610562(v=sc.12).aspx)

Microsoft Corporation. (n.d.-h). System.DirectoryServices.AccountManagement Namespace. Retrieved from [https://msdn.microsoft.com/en-us/library/system.directoryservices.accountmanagement\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.directoryservices.accountmanagement(v=vs.110).aspx)

Microsoft Corporation. (n.d.-i). Virtualization - Get enterprise-class virtualization for your datacenter and hybrid cloud. Retrieved February 26, 2016, from <https://www.microsoft.com/en-us/server-cloud/solutions/virtualization.aspx>

Microsoft Corporation. (2005). Moving Java Applications to .NET. Retrieved February 23, 2016, from <https://msdn.microsoft.com/en-us/library/ms973842.aspx>

Microsoft Corporation. (2011). TabControl Overview. Retrieved from [https://msdn.microsoft.com/en-us/library/ms749171\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/ms749171(v=vs.85).aspx)

Microsoft Corporation. (2012). Redircmp. Retrieved from [https://technet.microsoft.com/en-us/library/cc770619\(v=ws.11\).aspx](https://technet.microsoft.com/en-us/library/cc770619(v=ws.11).aspx)

- Microsoft Corporation. (2015). What's New in Hyper-V for Windows Server 2012 R2. Retrieved February 26, 2016, from <https://technet.microsoft.com/en-us/library/dn282278.aspx>
- Mombrea, M. (2013). Choosing a virtualization strategy for general purpose web hosting. Retrieved February 23, 2016, from <http://www.itworld.com/article/2710775/virtualization/choosing-a-virtualization-strategy-for-general-purpose-web-hosting.html>
- Natário, R. (2011). Redes e Servidores: Virtualização (II). Retrieved February 23, 2016, from <http://redes-e-servidores.blogspot.pt/2011/10/virtualizacao-ii.html>
- Oswalt, M. (2013). [Overlay Networking] Part 1 - The Basics. Retrieved February 23, 2016, from <https://keepingitclassless.net/2013/09/overlay-networking-part-1-the-basics/>
- Özsu, M., & Valduriez, P. (2011). *Principles of distributed database systems*. Springer. <http://doi.org/10.1007/978-1-4419-8834-8>
- Paper, a W., & Analyst, E. M. a S. (2006). Virtualization 101 : Technologies , Benefits , and Challenges, (August 2006).
- Petković, D. (2008). *Microsoft ® SQL Server™ 2008 - A Beginner's Guide* (4th ed.). McGraw-Hill Companies.
- Popek, G. J., & Goldberg, R. P. (1973). Formal requirements for virtualizable third generation architectures. *ACM SIGOPS Operating Systems Review*, 7(4), 121. <http://doi.org/10.1145/957195.808061>
- Qureshi, O. (2007). Application Virtualization: The Next Frontier. Retrieved February 23, 2016, from <https://technet.microsoft.com/en-us/library/bb608288.aspx>
- Ramakrishnan, R., & Gehrke, J. (2003). *Database Management Systems. Computing* (3rd ed., Vol. 8). McGraw-Hill. http://doi.org/10.1300/J115v06n04_07
- Ribeiro, M. S. (2009). Virtualization Basics. Retrieved February 23, 2016, from <https://itechthoughts.wordpress.com/2009/11/10/virtualization-basics/>
- Rouse, M. (2013). What is scalability? - Definition from WhatIs.com. Retrieved February 23, 2016, from <http://searchdatacenter.techtarget.com/definition/scalability>
- Ruest, N., & Ruest, D. (2009). *Virtualization: A Beginner's Guide*. McGraw-Hill Companies.
- Siemens, C. (2014). How to Install Hyper-V on vmware Workstation 10. Retrieved from <http://social.technet.microsoft.com/wiki/contents/articles/22283.how-to-install-hyper-v-on-vmware-workstation-10.aspx>

- Steil, M. (2006). Inside VMware. *23rd Chaos Communication Congress*, (Vmm). Retrieved from <http://events.ccc.de/congress/2006/Fahrplan/attachments/1132-InsideVMware.pdf>
papers2://publication/uuid/01624764-5463-40F6-B430-5D9AA44C7388
- Tracy, M. (2003). Walkthrough: Editing an Access Database with ADO.NET. Retrieved February 26, 2016, from <https://msdn.microsoft.com/en-us/library/ms971485.aspx>
- Virtualization. (2012). Retrieved February 23, 2016, from <http://priyanet.org:8090/managedservices.istonish/virtualization/>
- Virtue IT. (2015a). Virtualização de Aplicações. Retrieved February 23, 2016, from http://www.virtueit.com.br/site/?page_id=463
- Virtue IT. (2015b). Virtualização de Desktops. Retrieved February 23, 2016, from http://www.virtueit.com.br/site/?page_id=459
- VMware. (n.d.). What is virtualization? Virtualization 101. Retrieved February 23, 2016, from <http://www.vmware.com/virtualization/how-it-works.html>
- VMware. (2007). Understanding Full Virtualization, Paravirtualization, and Hardware Assist. *Memory*, 17. Retrieved from www.vmware.com
- Zyp, K. (2010). NoSQL Architecture. Retrieved February 23, 2016, from <https://www.sitepen.com/blog/2010/05/11/nosql-architecture/>