INSTITUTO SUPERIOR DE TECNOLOGIAS AVANÇADAS

Projeto Global

Licenciatura em informática

Virtualização

Nome: João António Pinto Nascimento, Nº8136

Orientador: Pedro Brandão

Lisboa

Ano Letivo: 2017/2018

Resumo

O objetivo deste Projeto consiste no desenvolvimento de uma Rede Virtual e de uma aplicação ADO.NET e C# num ambiente de Virtualização, para isto foram utilizadas tecnologias de Virtualização de servidores. Através da utilização do *VMware* que é uma tecnologia que permite trabalhar em ambientes virtuais, esta tecnologia permite então a criação de um *Domain Controler* (DC), um Servidor *SQL* que contém a base de dados necessária para a aplicação C# e ADO.NET, um Servidor *Routing* que é utilizado para efetuar a ligação com a rede externa, e distribui internet pela rede interna e uma máquina virtual *Client* que contém a aplicação criada. Apenas os utilizadores de domínio conseguem aceder à aplicação e visualizar os registos dos dados da base de dados.

Este método de virtualização assente em *VMware,* permite a implementação de um conjunto de máquinas virtuais todas elas a comunicarem entre si, conseguindo uma otimização de recursos físicos e uma poupança de energia bastante elevada.

Palavras Chave:

Virtualização, SQL Server, Visual Studio, Windows e C#.

Abstract

The purpose of this project is to develop a Virtual Network and an ADO.NET and C # application in a Virtualization environment, for which server virtualization technologies were used. Through the use of VMware that is a technology that allows working in virtual environments, this technology allows the creation of a Domain Controller (DC), a SQL Server that contains the necessary database for the C # and ADO.NET application, a Routing server that is used to connect to the external network, and distributes the internet through the internal network and a virtual client machine that contains the application created. Only domain users are able to access the application and view the data records of the database.

This virtualization approach relies on VMware, allows the deployment of a set of virtual machines all of which communicate with each other, achieving optimum physical resources and a high energy savings.

Keywords:

Virtualização, SQL Server, Visual Studio, Windows e C#.

Lista de Acrónimos

Technological Information.....*TI*

Ip Adress management.....*IPAM*

Dynamic Host Configuration Protocol.....*DHCP*

Internet Protocol.....*IP*

Domain Name System.....*DNS*

Virtual Machine Manager.....*VMM*

Mac address.....*MAC*

Structured Query Language.....*SQL*

VBVisual Basic.....*VB*

Software as a Service.....*SaaS*

Infrastructure as a Service.....*IaaS*

Platform as a Service.....*PaaS*

C-Sharp.....*C#*

Active Directory.....*AD*

Índice

Índice de Figuras

Introdução

A virtualização devido ao desenvolvimento tecnológico, cada vez mais faz parte da realidade atual das empresas facultando enumeras vantagens.

Este projeto tem como objetivo a criação de uma rede empresarial sustentada em tecnologias de Virtualização. A virtualização permite que as empresas consigam criar ambientes virtuais, o que permite uma otimização dos recursos já existentes, flexibilidades a nível de alterações e uma poupança energética considerável. A virtualização através da otimização dos recursos e da execução de tarefas mais rapidamente aumenta ainda produtividade. Através desta tecnologia é também possível testar produtos em desenvolvimento, através da criação de ambientes de teste que permitem essa flexibilidade, colocando muito mais rapidamente um produto no mercado.

Neste Projeto será criada uma Rede Virtual e uma aplicação ADO.NET e C# num ambiente de Virtualização. Será utilizado o *VMware* que permite trabalhar em ambientes virtuais, através desta tecnologia será implementado um *Domain Controler* (DC), um Servidor *SQL* que irá conter a base de dados necessária para a aplicação C# e ADO.NET, um Servidor *Routing* que será utilizado para efetuar a ligação com a rede externa, e distribui internet pela rede interna e uma máquina virtual *Client* que conterá a aplicação criada.

Estado de Arte

Virtualização

A virtualização consiste na criação de um ambiente informático virtual, inclui ainda versões geradas por computador a nível de *hardware*, sistemas operativos, dispositivos de armazenamento, ente outros. Isto permite a criação de várias máquinas virtuais a partir de um servidor ou computador físico. Cada máquina virtual pode funcionar individualmente, pode executar diferentes sistemas operativos (por exemplo, Windows ou Linux) ou aplicações. (Microsoft, s.d.)

A criação de recursos a partir de um servidor ou computador físico, a virtualização melhora consideravelmente tanto a escalabilidade assim como as cargas de trabalho, o que resulta numa menor utilização de servidores físicos, por consequente menor consumo de energia e custos de manutenção da infraestrutura. A virtualização baseia-se em quatro categorias, ao virtualizar um ambiente de trabalho, permite que um servidor centralizado possa fornecer e gerir ambientes de trabalho individualizados. (Microsoft, s.d.)

A virtualização de rede, permite a divisão da largura de banda, de forma a ser atribuída a servidores ou dispositivos específicos. Virtualização de *Software*, o que permite a separação de aplicações do *hardware* e do sistema operativo. Virtualização de armazenamento, junta num único dispositivo de armazenamento, vários recursos de armazenamento de rede, onde vários utilizadores podem aceder. (Microsoft, s.d.)

Virtualização e servidores

Como já foi mencionado anteriormente, a virtualização de servidores consiste em dividir recursos de um servidor físico em vários servidores virtuais, ou seja, máquinas virtuais, de forma a que se possa executar vários sistemas operativos com o mesmo *hardware* físico, isolados entre si. Um servidor físico a nível de *hardware*, vem de fábrica com vários recursos físicos, como por exemplo, *CPU*, memória *RAM*, discos, placas de rede. (Blue Solution, 2014)

Assente no *hardware* do servidor físico é instalado um *Hypervisor*, de forma a permitir a divisão dos recursos do *hardware*. Quanto menores forem os recursos ocupados pelo *Hypervisor*, mais recursos sobram para serem utilizados pelas máquinas virtuais.

Cada máquina virtual pode ter capacidades diferentes dependendo do seu fim, umas podem ter mais memória *RAM*, outras mais processador e outras mais espaço em disco, mas tudo isto com base na divisão dos recursos originais (do servidor físico). O *Hypervisor* fica responsável pela divisão dos recursos entre as máquinas virtuais, alguns recursos podem ser alocados numa quantidade maior do que a existente na realidade (*over provisioning*). (Blue Solution, 2014)

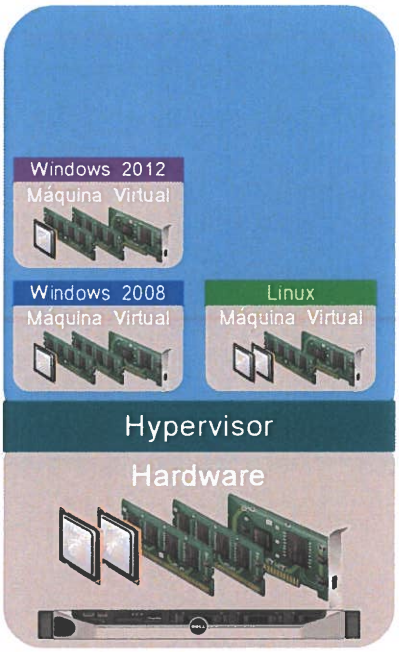

Figura 1- Exemplo de virtualização, com máquinas virtuais assentes num Hypervisor, que está assente no Hardware físico de um servidor.

Fonte: (Blue Solution, 2014)

Windows Server 2012 R2

O *Windows server* 2012 R2 foi desenvolvido com a finalidade de suportar a *Cloud Computing* e a convergência de equipamentos como computadores, *tablets* e *smartphones*. O *Windows server* 2012 R2 foi desenvolvido para garantir um servidor de rede caracterizado pelo seu desempenho, segurança, funcionalidades, facilidade de utilização e manutenção. (Rosa, 2013, p. 4)

Surgiram várias edições do *Windows Sever* 2012, com diferentes características e funcionalidades, a *Foundation, Essentials, Standard e Datacenter*.

Edição *Foundation*, destinada a implementações simples, tipicamente para pequenas empresas. O preço desta versão é proporcional às características, isto torna esta versão apelativa para um grande número de empresas de pequeno porte. A sua política de licenciamento estabelece um limite de 15 contas de utilizadores, não permite um utilizador tenha mais que duas sessões abertas, não exige a aquisição de *Client Access Licenses(CAL)*. Se for usado com *Terminal Server* tem um limite de 50 sessões. Uma licença deste tipo não permite a virtualização. (Rosa, 2013, p. 14)

Edição *Essentials*, em relação às Edição *Foundation*, esta versão suporta 25 utilizadores, 250 conexões de serviços de terminal e 250 sessões de *RRAS*. Em relação à virtualização, o esquema de licenciamento desta versão apenas permite a instalação numa máquina virtual ou física, mas não em ambas em simultâneo. (Rosa, 2013, p. 15)

Edição *Standard*, destina-se a pequenas e médias empresas, é utilizado quando se verifica que que é necessário mais características/serviços do que a edição *Foundation* e *Essentials* disponibilizam. Esta versão suporta serviços como partilha de ficheiros e impressoras, validação de utilizadores no *Active Directory, suport* para *VPN*, entre outros. Em relação à virtualização, uma licença deste tipo permite instalar numa máquina física e em duas virtuais. (Rosa, 2013, p. 15)

Edição *Datacenter*, é a edição mais escalável, pois suporta um valor muito elevado a nível de *RAM (2TB de RAM)* e máquinas com 2 e até um máximo de 64 processadores. Esta edição só pode se adquirida por fabricantes de equipamentos devidamente qualificados. Esta edição permite a instalação de máquinas virtuais ilimitadas e permite

também um número de utilizadores ilimitado (mas no caso dos utilizadores é necessário a aquisição de *Client Access Licenses(CAL)*). (Rosa, 2013, p. 16)

NIC Teaming

Também conhecido como *load-balancing failover,* o *NIC Teaming* permite juntar várias placas de rede aumentando a largura de banda e a implementação da redundância. No caso de falha de uma porta de rede do *Hyper-V* ou de um adaptador de rede virtual, automaticamente outro adaptador de rede virtual colmata essa falha.

(Rosa, 2013, p. 41)

Nic Teaming é totalmente compatível com o *Hyper-V Network Virtualization.* O sistema de gestão do *Hyper-V Network Virtualization* transmite informações ao *NIC Teaming*, o que permite que este de forma otimizada distribua a carga para o tráfego do *Hyper-V Network Virtualization.* (Microsoft, 2017)

Hyper-V

O *Hyper-V* é o *hypervisor* da *Microsoft*, ou seja, é a tecnologia usada para Virtualização da *Microsoft* (utilizada por exemplo no *Windows server 2012 R2.* Utilizando esta funcionalidade é possível emular máquinas virtuais, com sistemas operativos instalados e aplicações como se estivéssemos a trabalhar numa máquina real. Esta capacidade é bastante útil principalmente em ambientes de teste e produção, pois maximiza os equipamentos físicos, isto porque permite que uma máquina física possa correr em simultâneo mais do que um sistema operativo. (Rosa, 2013, p. 9)

Exemplos de aplicações praticas

É possível estabelecer ou expandir um ambiente de *Private Cloud*, aumentar a utilização do *hardware*, o *Hyper-V* ajuda a minimizar o impacto do tempo de inatividade, criar ou expandir uma infraestrutura de *Desktop's* Virtuais (*VDI*) e aumenta a eficiência

nas atividades quer seja de desenvolvimento assim com em teste (usar máquinas virtuais para reproduzir diferentes ambientes). (Microsoft, 2016)

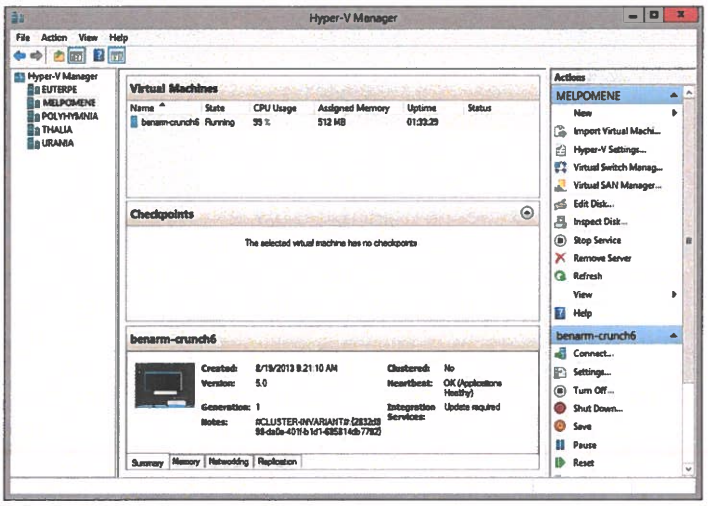

Figura 2 - Hyper-V Manager

Fonte: (Armstrong, 2014)

Algumas funcionalidades do Hyper-V:

Live Migration – Migrar máquinas virtuais de um servidor para outro, sem impacto na utilização dos utilizadores.

Dinamic Memory – O *Hyper-V* utiliza a memória como um recurso compartilhado, que é redistribuído automaticamente pelas máquinas virtuais em execução.

Active Directory (AD)

O *Active Directory* tem um papel muito importante na implementação de uma rede, seja ela qual for. Em qualquer negócio hoje em dia é necessário ter informações rápidas, atualizadas facilmente, mas principalmente seguras. O *Active Directory* consegue garantir essas características. (Rover, 2012)

O *Active Directory* surgiu em conjunto com o *Windows server 2000*, veio resolver vários problemas, no Exemplo do *NIS (Network Information Service, o Active Directory*, surge da necessidade de os utilizadores terem uma *Password* para aceder a todos os recursos disponíveis da rede, e não uma para cada serviço. O *Active Directory* guarda estruturadamente e hierarquicamente as informações dos objetos de rede de forma a disponibilizá-la tanto a administradores e cliente como para as aplicações. Tanto a Rede como os objetos são organizados na forma de construções, por exemplo árvores, florestas e sites. Assenta no protocolo *LDAP* e na comunicação através de replicação. Esta arquitetura permite uma gestão de rede mais produtiva. (Rover, 2012)

Quando se efetua o login com um utilizador no domínio de uma empresa (por exemplo), sem que se perceba utiliza-se um *Directory Service* e por consequência o *Active Directory*. (Rover, 2012)

Quando estamos a instalar o *Active Directory* é necessário, se ainda não estiver disponível, instalar também o serviço de *DNS* (pode ser instalado em conjunto com o *Active Directory*). O *Active Directory* utiliza o *DNS* para a resolução de nomes e para nomear servidores de recursos. (Rover, 2012)

O *Active Directory* fisicamente tem uma base de dados (Ntds.dit localizado na pasta %SystemRoot%\NTDS\ntds.dit), este ficheiro só existirá nos servidores que que foram "promovidos" a *Domain Controller (DC).* Neste Diretório durante a instalação do *Active Directory* são criados 5 ficheiros, Ntds.dit (ficheiro, de base de dados), Edb.log (ficheiro que armazena as transações feitas pelo AD), Edb.Chk (ficheiro de Checkpoint este controla as transações feita no Edb.log que já foram guardadas no Ntds.dit), Res1.log e Res2.log (se não existir espaço em disco, estes dois ficheiros asseguram que as alterações efetuadas sejam gravadas no Ntds.dit). (Rover, 2012)

O *Active Directory* "apoia-se" nos *Domains Controllers (DC's)*, em que se um *DC* falhar os outros *DC's* conseguem responder às necessidades da rede. Só é possível isto acontecer porque quando um servidor recebe a função *DC*, herda toda a estrutura do diretório (todos os dados criados são replicados para esse *DC*). (Rover, 2012)

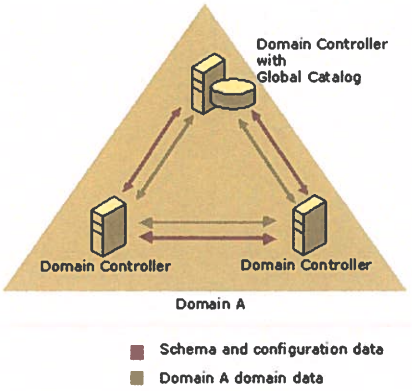

Figura 3 - Neste exemplo todos os DC's têm todos os dados de igual forma (replicação). Fonte: (Microsoft, 2000)

Estrutura lógica da Active Directory:

A estrutura lógica do *AD* está dividida em Objetos, Unidades Organizacionais (*OU*), Domínios, Árvores e Florestas. (Rover, 2012)

Objetos:

São os componentes mais básicos de uma estrutura de rede, são por exemplo, utilizadores, grupos, *passwords*, relações de confiança, impressoras, Unidades organizacionais, etc. (Rover, 2012)

Unidades Organizacionais (*OU*)

As *UO's* são objetos que têm como finalidade organizar outros objetos. Podem ser organizados de forma, geográfica, por departamentos e setores. (Rover, 2012)

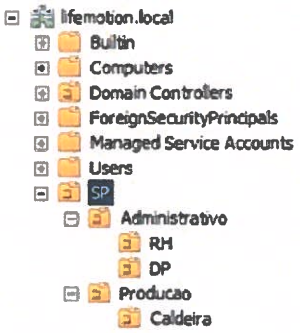

Figura 4 – Uma Unidade Organizacional.

Fonte: (Microsoft, 2012)

Domínios

O Domínio tem duas funções principais, cria um limite para os objetos (os objetos que estão dentro do domínio não podem sair dele e os que estão fora não podem entrar), esta regra pode ser contornada consoante as permissões de entrada e saída, como também através das relações de confiança. Gere dentro do *Active Directory* a Segurança das contas e recursos. (Rover, 2012)

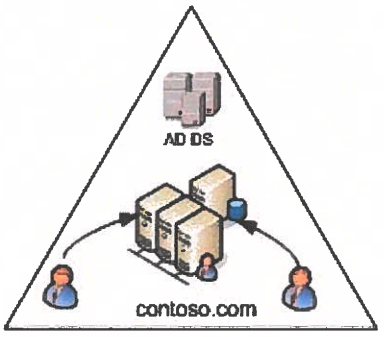

Figura 5 – Topologia de Domínio único.

Fonte: (Microsoft, 2013)

Árvores

Quando é necessário criar um segundo domínio, muitas das vezes por razões de segurança, esse mesmo domínio é chamado de Subdomínio. Quando temos um domínio com vários subdomínios dá-se o nome de árvore, os subdomínios mantêm o sufixo *DNS* do domínio principal, mas em distribuição hierárquica. (Rover, 2012)

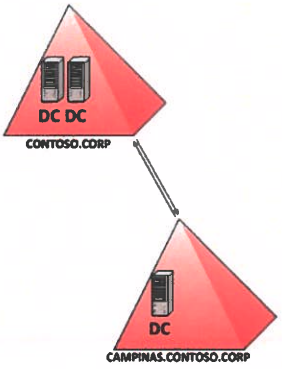

Figura 6 – Neste exemplo temos o Domínio principal (CONTOSO.CORP), e temos um subdomínio (CAMPINAS), sabendo que os subdomínios herdam o sufixo do principal e dos anteriores, o subdomínio fica CAMPINAS. CONTOSO.CORP.

Fonte: (Vidal, 2010)

Florestas

O primeiro domínio de uma floresta (Root Domain), será o nome da floresta. A floresta pode ser feita de um só domínio, mas também pode estar dividida em árvores dentro da mesma floresta. (Rover, 2012)

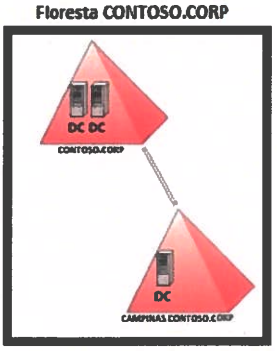

Figura 7 – Floresta com um domínio e um subdomínio.

Fonte: (Vidal, 2010

Estrutura Física de Active Directory:

A estrutura física do *Active Directory* consiste no *Domiain Controllers* e *Sites*. A estrutura Física do *AD* é totalmente independente da estrutura lógica do *AD*. A estrutura física tem como finalidade otimizar o tráfego de rede, e por garantir a segurança em locais físicos distintos. (Rover, 2012)

Domain Controllers (DC)

Um *DC* tem como função executar o *Active Directory*, armazenar e replicar toda a informação da base de dados com os outros *DC's*. De realçar que um *DC* apenas suporta um único domínio, daí que para criar uma maior disponibilidade do *AD*, podemos ter mais que um *DC*, em que a informação da base de dados (ficheiro NTDS.dit) é replicada de um *DC* para o outro. (Rover, 2012)

Tendo os *DC's* sincronizados entre si, é possível ter a *Active Directory* estável, em que se algum *DC* falhar os outros *DC's*, devido à sincronização e replicação existente entre eles, conseguem colmatar essa mesma falha de forma a não afetarem a rede. (Rover, 2012)

Sites

Os *sites* organizam a latência da replicação dos *DC's* no presente site, também permite que um Site não utilize desnecessariamente o *link* de replicação. Com esta organização por *sites* é possível restringir um certo grupo de computadores estabelecer contacto com a sua matriz ou apenas em horários de menos fluxo (agendamento de uma replicação). A tipologia de *sites* do *AD* é o mapa que descreve a conectividade de rede, diretrizes de replicação do *AD* e locais para os recursos dependendo da forma como se relacionam com a floresta do *AD*. Nesta tipologia de site os principais componentes são, as sub-redes, objetos de conexão, pontes de *link de sites e links de sites*. Estes objetos são mantidos no *container* de configuração da floresta, o que permite que localmente em todos os *DC's* estejam disponíveis as interações, para que os *DC's* consigam comunicar entre si de forma adequada. A replicação do *AD* entre *sites* pode ser usada através de *IP* ou de *SMTP* (se a rede for mais lenta). (Rover, 2012)

System Center Virtual Machine Manager

O *System Center Virtual Machine* é destinado à gestão de *DataCenters* virtualizados integrados à *Cloud*, permitindo a configuração e gestão de recursos do *host,* rede e de armazenamento para criar e implementar máquinas e serviços virtuais em *Private Cloud*. O *Virtual Machine Manager (VMM*) faz parte do *System Center*.

(Microsoft, 2017)

Capacidades do *Virtual Machine Manager (VMM*)

Datacenter – Configurar e gerir componentes de um *datacenter*, estes componentes incluem servidores de Virtualização, componentes de rede e recursos de Armazenamento. O *VMM* prevê e gere os recursos necessários de forma a criar máquinas virtuais e serviços virtuais em *Private Cloud*. (Microsoft, 2016)

Hosts de Virtualização – O *VMM*, gere, adiciona e provisiona *hosts* e *clusters* de virtualização do *VMware* e *Hyper-V*. (Microsoft, 2016)

Armazenamento – O *VMM* suporta armazenamento de bloco (*fibre channel, iSCSI, e Serial Attached SCSI(SAS) storage are networks (SANs)*). (Microsoft, 2016)

Rede – O *VMM* fornece a possibilidade de virtualizar uma rede, incluindo também suporte para criar e gerir redes virtuais e *gateways* na rede. A virtualização de rede permite ainda que existam redes isoladas e com os seus próprios intervalos de endereços *IP*, para uma maior privacidade e segurança. (Microsoft, 2016)

Recursos da biblioteca – O *VMM* mantem uma biblioteca de recursos baseados em arquivos. Os recursos baseados em arquivos incluem discos rígidos virtuais, *ISO* e *Scripts*. Os recursos que não são baseados em arquivos incluem modelos e perfis usados para padronizar a criação de máquinas virtuais. Os Recursos da Biblioteca são acedidos através da partilha de bibliotecas. (Microsoft, 2016)

SQL

A linguagem *SQL (Structure Query Language)* surgiu na década de 70, e desde então que é a linguagem mais utilizada no que toca a Bases de Dados. *SQL Server* é uma linguagem que permite a manipular, gerir e recuperar dados de uma base de dados relacional. Uma Base de dados é um conjunto de informações relacionadas.

(Rouse, 2016)

A gestão da Base de dados é efetuada através de Sistemas de Gestão de Base de Dados (*SGBD*), todos os *SGBDs* oferecem ferramentas de forma a se aceder a uma Base de Dados utilizando a linguagem *SQL*, é ainda um software que facilita os processos de construção, manipulação e partilha de Base de dados entre vários utilizadores.

(Alves, 2013)

O *SQL Server 2017*, é a versão mais recente, representa um passo importante, no que toca a tornar o *SQL server* uma plataforma mais completa, que ofereça escolhas de linguagens de desenvolvimento, tipos de dados, melhoramentos a nível de *Cloud* e sistemas operativos. O *SQL Server* 2017 conta com ferramentas e funcionalidades que proporcional uma melhor utilização, como a reconstrução do *índice online* remanescente, retoma a reconstrução *índice online* onde tinha parado anteriormente devido a uma falha (como um *failover)* ou agenda essa operação, o *IDENTITY_CACHE*, que permite evitar lacunas nas colunas de identidade caso um servidor seja reiniciado ou falhe o servidor secundário. Ajuste automático da base de dados, fornece informações sobre possíveis problemas de desempenho, recomenda soluções e também pode corrigir automaticamente os problemas identificados. *Database Tuning Advisor*, inclui opções adicionais e melhor desempenho. Funções *String* novas, *CONCAT_WS, TRANSLATE, TRIM e WITHIN GROUP* agora é suportado para a função *STRING_AGG*. (Microsoft, 2017)

Microsoft Visual studio

O *Microsoft Visual Studio* é um ambiente de desenvolvimento integrado *(IDE)*. É utilizado para desenvolver *Software* (aplicações *web, sites*, Aplicações móveis), através das plataformas de desenvolvimento que o *Microsoft Visual Studio* possui, *Windows API, Windows Presentation Foundation, Windows Forms, Windows Store e Microsoft Silverlight.* Suportar várias linguagens de programação como, C#, C++, *JavaScript* ou *Python*. (Microsoft, s.d.)

ADO.NET

O *ADO.NET* é um conjunto de classes que fornece um conjunto de serviços para aceder a dados, a programadores de *.NET Framework*. O *ADO.NET* fornece um conjunto de componentes para a criação de aplicações distribuídas e partilha de dados (teve um papel muito importante para programadores de C#, *Visual Basic*, entre outros). Faz Parte do *.NET Framework*, proporciona acesso relacional, *XML e Data aplication. ADO.NET* fornece ainda suporte a uma variedade de necessidades de desenvolvimento, como criação de clientes *Front-end* de Base de dados, objetos comerciais da camada intermedia usada por aplicações, ferramentas, linguagens ou *Browsers*. (Microsoft, 2017)

O *ADO.NET* separa o acesso a dados da manipulação de dados em componentes discretos que podem ser usados separadamente. Inclui ainda *.NET Framework data providers*, para a conexão à base de dados, executar comandos e recuperar resultados. Estes resultados são processados e alojados num objeto *DataSet* do *ADO.NET*, de forma a serem expostos a utilizadores *ad hoc*, combinados com dados de várias fontes ou passados entre camadas. O objeto *DataSet* também pode ser usado para gerir o *data local* para as aplicações. (Microsoft, 2017)

As Classes do *ADO.NET* encontram-se no *System.Data.dll* e são integradas às classes *XML* (estão no *System.Xml.dll*). O *ADO.NET* fornece funcionalidades a

programadores que gerem código, semelhantemente à funcionalidade fornecida aos programadores *COM (Component Object Model)* nativos pelos objetos *ActiveX Data Objects (ADO)*. É aconselhável efetivamente que se utilize o *ADO.NET*, e não o *ADO*, para aceder a aplicações *.NET*. (Microsoft, 2017)

O *ADO.NET* fornece a forma mais direta de acesso a dados *.NET Framework*, para uma abstração de alto nível permitindo que aplicações funcionem num modelo conceptual, em vez de num modelo de armazenamento subjacente. (Microsoft, 2017)

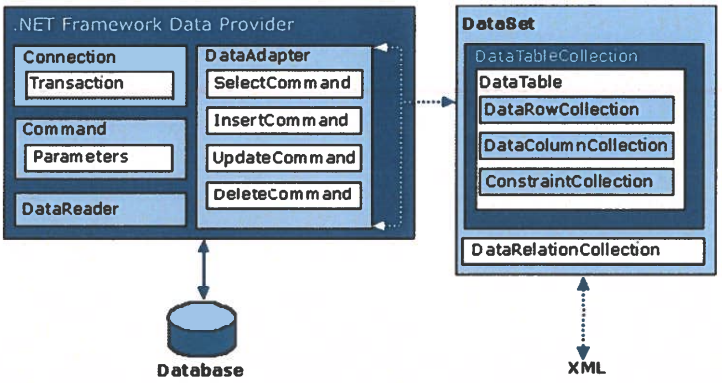

Figura 8 - .Net Framework e DataSet.

Fonte: (Microsoft, s.d.)

C#

C# é uma linguagem orientada a objetos, isto permite aos programadores a criação de uma grande variedade de aplicações bastante seguras, que são executadas na *.NET Framework*. C# é uma linguagem que pode ser usada para, criar aplicações cliente *Windows*, serviços *Web* em *XML*, aplicações Cliente Sevidor, aplicações com ligação à base de dados, entre outras. (Microsoft, 2015)

A sintaxe do C# é bastante intuitiva e fácil de aprender, principalmente para conhecedores de C, C++ ou Java. Programadores conhecedores destas linguagens rapidamente se adaptam ao C#. Esta linguagem simplifica em muito a complexidade do C++, esta também fornece uma serie de recursos como, *delegates*, acesso direto à

memória e expressões lambda. C# suporta *generic mothods and types*, que para além de um desempenho melhorado também fornecem segurança, e *iterators* que permitem que coleções de classes definam que *custom iteration* sejam simples de serem utilizados. (Microsoft, 2015)

Sendo C# uma linguagem orientada a objetos, esta suporta conceitos de herança, polimorfismo e encapsulamento. Todos os métodos e variáveis são encapsulados dentro das definições de uma classe, sendo que uma classe pode herdar diretamente de uma classe pai e pode implementar um número de *interfaces*. (Microsoft, 2015)

A linguagem C# facilita também o desenvolvimento de componente de software como, propriedades, que servem para aceder a variáveis de membros privados, atributos que fornecem meta dados declarativos sobre *types* em execução, comentários de documentação *XML* e *LINQ* (*Language-Integrated Query*), que permite recursos de consulta integrados numa variedade de *data sources.* (Microsoft, 2015)

Programas C# são executados no .Net Framework, isto é um componente integrado do Windows em que o mesmo inclui um sistema de execução virtual, que se de se denomina de CLR (*Common Language Runtime*) e um conjunte de bibliotecas de classes. (Microsoft, 2015)

Cloud Computing

Características

On-Demand Self-Service – Um Consumidor que utilize serviços de *Cloud*, se verificar que necessita de mais recursos, pode aumentar as capacidades computacionais, por exemplo armazenamento em rede, sem que seja necessário a intervenção humana.

(Pinheiro, 2012)

Ubiquitous Metwork Access – Os serviços de *Cloud* podem ser acedidos, através da rede e através de mecanismos padrão que promovem as plataformas heterogêneas (exemplo: telemóveis, computador, *tablet*). (Pinheiro, 2012)

Resource Pooling – Os recursos de Computação da *Cloud* estão reunidos geograficamente. Os seus recursos virtuais são dinamicamente atribuídos ou reatribuídos conforme as exigências do consumidor. O Consumidor não possui controlo nem a informação do local exato, de onde estão a ser fornecidos os seus recursos (exemplo: Memória, processamento, armazenamento). Apenas é possível ter uma informação mais ampla como o país em que se encontra, continente ou *Data Center*. (Pinheiro, 2012)

Rapid Elasticy – A Elasticidade define-se como disponibilizar mais ou menos recurso quando é necessário, de forma rápida e eficiente. Um Consumidor de *Cloud*, irá ficar com a perceção de que a *Cloud* parece ser infinita, pois o mesmo pode adquirir mais ou menos poder computacional conforme seja necessário para as suas aplicações. Esta é uma das características que torna tão atrativa a *Cloud Computing*. (Pinheiro, 2012)

Measured Service – Todos os serviços na *Cloud* são controlados e monitorizados de forma automática (processamento, largura de banda, contas de utilizador, entre outros), isto também ajuda a otimizar a utilização ao consumidor, assim como ajuda ao fornecedor na contabilização do que foi utilizado. De forma a que exista transparência tanto para o fornecedor de *Cloud* assim como para o consumidor na hora da cobrança do serviço.

(Pinheiro, 2012)

Modelos de Serviço:

Software as a Service (SaaS)

Permite aos utilizadores a conexão a aplicações ou serviços via internet. Os Consumidores/utilizadores compram junto dos fornecedores de serviços *Cloud*, a capacidade de aceder e utilizar uma aplicação ou um serviço presente na *Cloud*. O fornecedor de serviços *Cloud* gere o *hardware* e *Software,* também garante a disponibilidade e segurança da aplicação ou serviço fornecido. A informação necessária da interação entre o consumidor e o serviço apresenta-se como parte do serviço na *Cloud*, apenas é pago o que é utilizado, este modelo é flexível e permite controlar o que necessitamos. Por exemplo, aquisição de serviços de email. (Microsoft Azure, s.d.)

(Dialogic Corporation, 2010)

Saas tem várias vantagens entre elas, obter acesso a aplicações sofisticadas, só é pago o que é utilizado, pode ser utilizado *software* cliente gratuito, facilidade de trabalho para os utilizadores e acesso a dados de aplicações em qualquer lugar.

(Microsoft Azure, s.d.)

Platform as a Service (PaaS)

Os consumidores/utilizadores compram o acesso a plataformas, onde lhes é permitido desenvolver software e aplicações na *Cloud* para uso próprio. Os recursos são comprados junto do fornecedor de serviços *Cloud*, e os utilizadores acedem aos mesmo através de uma ligação à internet (segura).

O serviço *PaaS* permite evitar custos na compra e gestão de licenças de *software*, infraestruturas que suportam as aplicações, entre outros recursos. O utilizador gere as aplicações e serviços que desenvolve, o fornecedor de serviços *Cloud* por sua vez gere os restantes. (Microsoft Azure) (Dialogic Corporation, 2010)

O Paas tem a vantagem de que, é gasto menos tempo a programar, mantendo as mesmas equipas, acrescenta capacidades de desenvolvimento, desenvolve para várias

plataformas, utiliza ferramentas sofisticadas sem custos e gere o ciclo de vida das aplicações. (Microsoft Azure)

Infrastructure as a Service (IaaS)

Os consumidores controlam e gerem os sistemas quanto a sistemas operacionais, aplicações, armazenamento e conectividade de rede, mas não controlam a infraestrutura de nuvem. (Microsoft Azure) (Dialogic Corporation, 2010)

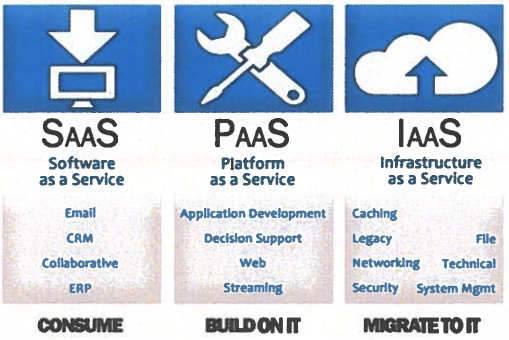

Figura 9 – Infraestruturas, SaaS, PaaS e IaaS.
Fonte: (Ivan, 2013)

Modelos de Implementação:

Cloud privada – Uma infraestrutura de *Cloud* Privada é destinada para uso exclusivo de uma organização. Pode ser gerida e pertencente a uma organização ("*in-house*) ou por uma terceira entidade ("*third party*"). Pode existir dentro ou fora das instalações da empresa. (Dialogic Corporation, 2010)

Cloud comunitária – Uma infraestrutura de *Cloud* Comunitária, é destinada a organizações que partilham de interesses idênticos. Pode ser gerida e pertencente a uma ou mais organizações, ou então por uma terceira entidade. Pode existir dentro ou fora das instalações da empresa. (Dialogic Corporation, 2010)

Cloud pública – Uma infraestrutura de *Cloud* Pública, é destinada ao uso do público em geral. Isto permite que um consumidor possa desenvolver e desdobrar um serviço, em *Cloud* sem tantos encargos financeiros. Pode ser gerida e pertencer a uma empresa, organização governamental ou uma terceira entidade. Existe na infraestrutura/instalações de um fornecedor de *Cloud*. (Dialogic Corporation, 2010)

Cloud híbrica – Uma infraestrutura de *Cloud* Híbrida, é composta por duas ou mais infraestruturas de *Cloud* (Privadas, Comunitárias e Públicas). Com esta estrutura é possível mover dados entre os diferentes tipos de *Cloud,* dentro de uma *Cloud* Híbrida. Estas permanecem como exclusivas, mas ainda assim estão conectadas por uma tecnologia padronizada, o que permitirá então a passagem de dados entre as diferentes infraestruturas de *Cloud.* (Dialogic Corporation, 2010)

Contextualização

Virtualização e *Cloud Compunting* é um tema bastante pertinente nos dias que correm, devido à ascensão tecnológica, espaço e recursos físicos começaram a ser um problema, o que fez com que surgisse e fosse desenvolvido tanto a virtualização, assim como a *Cloud Computing*. A Virtualização para além de economizar investimentos em novos equipamentos, ainda permite aproveitar ao máximo o *hardware* já existente.

Hoje em dia novas possibilidades de virtualização são exploradas, de forma a que se tire o máximo de partido da mesma.

Desenvolvimento

Especificações do Computador

Foram usadas várias ferramentas, como o *VMWARE* que suporta todo o laboratório de necessário para ser efetuado o projeto, este foi o escolhido para virtualização das máquinas virtuais e para o desenvolvimento aplicação o *software* usado foi o *Visual Studio*.

Para o desenvolvimento deste projeto que tem como base a virtualização, é necessário um computador que preencha alguns requisitos, pois caso contrário não será possível a elaboração do projeto sem constrangimentos. O computador utilizado para a realização deste projeto foi um ASUS ROG STRIX.

Características:

- Processador: *Intel*(R) *Core(*TM) i7-7700HQ *CPU* @ 2.80GHz 2.81GHz;
- Memória *RAM:* 16GB;
- Armazenamento: 256GB *SSD* e 1TB HDD;
- Sistemas Operativos: *Windows* 10 64bits, onde através do *VMWARE* foram virtualizadas máquinas virtuais com o *Windows Server Datacenter* e *o Windows* 8.1;
- *Software: SQL Server* 2014 *Management Studio e Visual Studio* 2017.

Esquematização do projeto

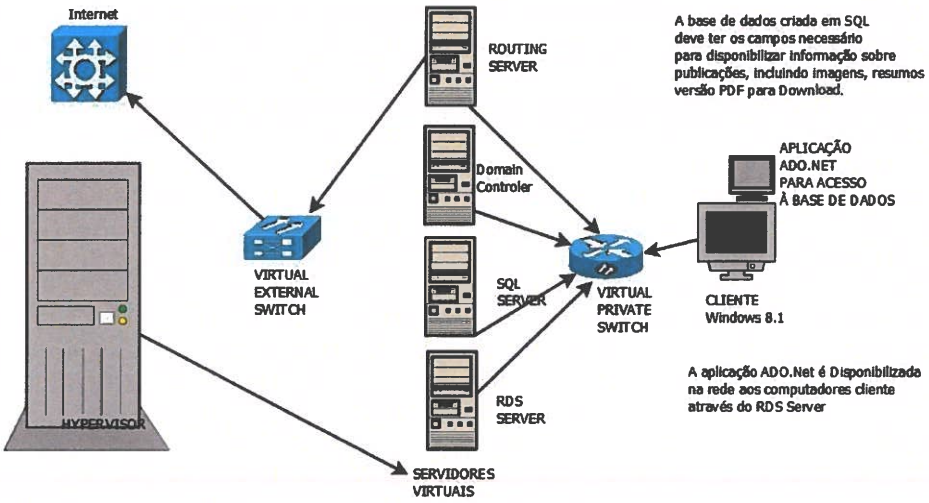

Figura 10 – Esquematização do projeto

Fonte: (Professor Pedro Brandão, 2018)

Dados das máquinas virtuais

	Sistemas Operativos	Armazenamento	RAM	IP	Adaptadores de rede
DC	Windows Server Datacenter	60GB	2GB	192.168.1.5	Host-only
Routing	Windows Server Datacenter	60GB	2GB	192.168.1.6	Host-only e NAT
SQL Server	Windows Server Datacenter	60GB	2GB	192.168.1.7	Host-only
Cliente	Windows 8.1	60GB	2GB	192.168.1.8	Host-only

Figura 11 - Dados das Máquinas Virtuais.

Fonte: do Autor

	Conta Login	Password
DC	ISTEC\Administrator	P@ssw0rd
Routing	ISTEC\Administrator	P@ssw0rd
SQL Server	ISTEC\Administrator	P@ssw0rd
Cliente	ISTEC\ Administrator	P@ssw0rd

Figura 12 - *Login* e *Password*.

Fonte: do Autor

Preparação das Máquinas Virtuais

Foi utilizado o *VMWARE* para suportar a totalidade das Máquinas Virtuais, o mesmo foi instalado na máquina local.

Após a instalação do *VMWARE*, será então possível a criação de Máquinas Virtuais. Para isto é necessário abrir o separador "*File*" e selecionar a opção "*New Virtual Machine*"

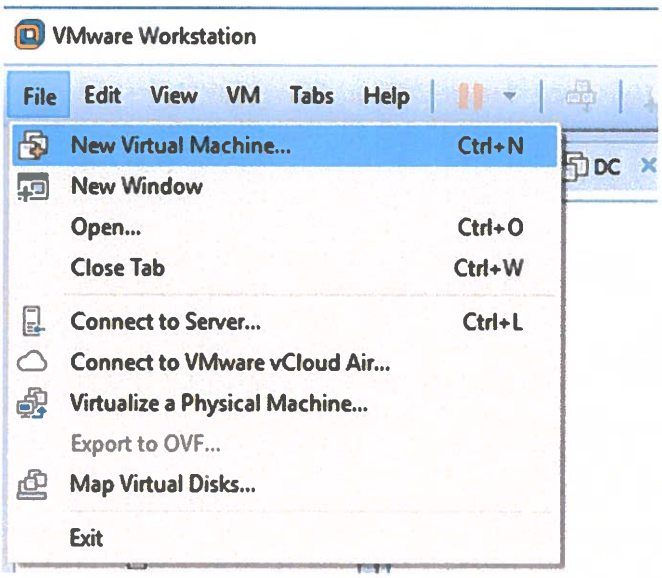

Figura 13 – Criar Máquina Virtual.

Fonte: do Autor

Na janela que abre imediatamente a seguir ao carregar em "*New Virtual Machine*", carregar em *next*, depois na janela *Guest Operating System Installation* selecionar a opção "*I will install the operating system later*" e carregar em *next.*

Na janela *Select a Guest Operating System* seleciona-se a opção "*Microsoft Windows*", na versão escolhemos "*Window Server 2012 R2*" no caso do *DC, ROUTING* ou *SQL*. No caso da máquina Cliente selecionamos a versão "*Windows 8* x64".

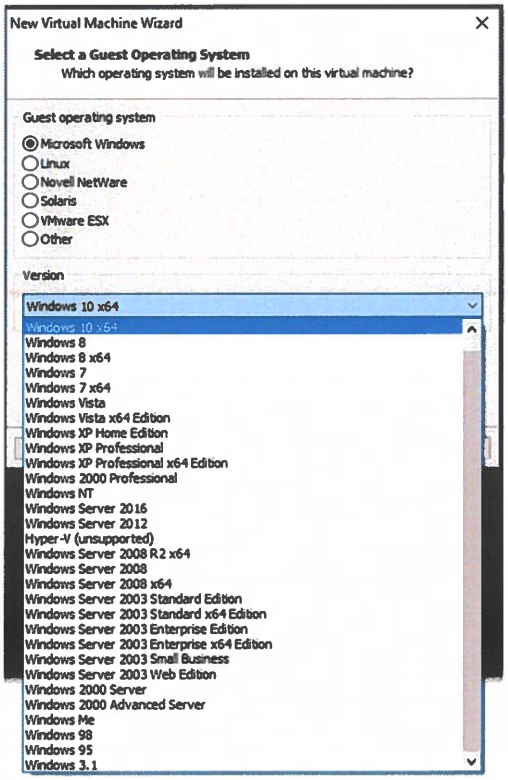

Figura 14 – Escolher o Sistema Operativo que irão ter as Máquinas Virtuais.

Fonte: do Autor

Na Janela *Name the Virtual Machine*, colocar o nome da máquina em questão que estamos a criar (*DC, ROUTING, SQL* e CLIENTE) e na *Location*, escolher o local onde se irá armazenar as máquinas virtuais.

Depois na janela *Specify Disk Capacity*, escolher a memória de armazenamento pretendida (neste caso foi escolhida a memória 60GB) e a opção "*Split virtual disk into multiple files*".

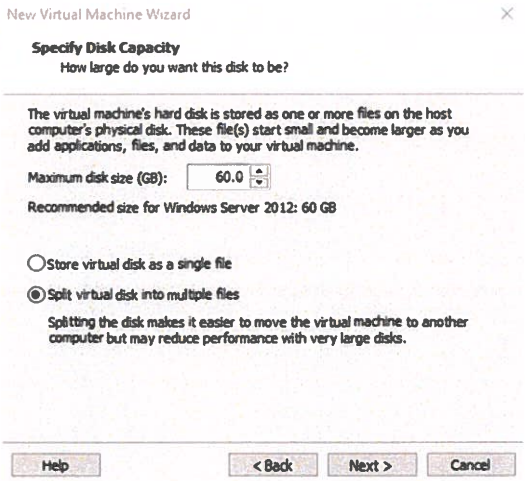

Figura 15 – Escolher o armazenamento pretendido.

Fonte: do Autor

Na janela seguinte (*Ready to Create Virtual Machine*) carregamos em "*Customize Hardware*", o que possibilita a alteração do *Hardware* da Máquina Virtual. No *DC, SQL* e CLIENTE vamos a "*Network-Adapter*" e colocamos em *Host-only* e no *ROUTING* colocamos *Host-only* e adicionamos uma *NAT* (é através da *NAT* que temos ligação à internet). Depois é necessário carregar em "*New CD/DVD*" e adicionar o *ISO* do sistema operativo. Por fim carregamos em *Close* e *Finish*.

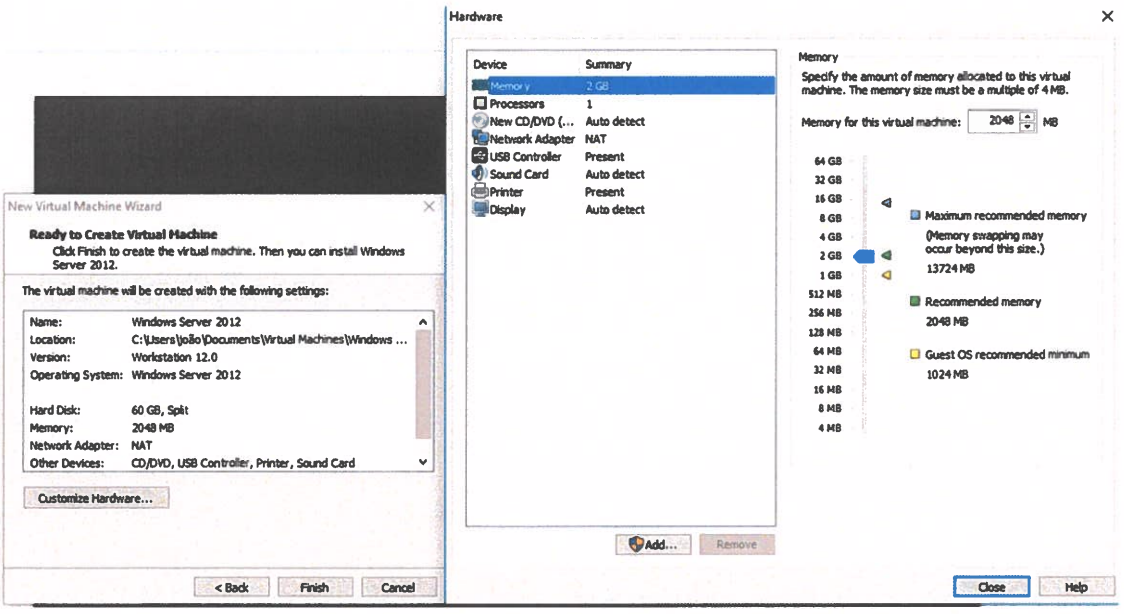

Figura 16 – Alteração do Hardware.

Fonte: do Autor

Domain Controler (DC)

Descrição do Servidor *Domanin Controler* usado:

- Nome: DC
- Endereço de IP (IPv4): 192.168.1.5
- Gateway: 192.168.1.6
- Máscara de sub-rede: 255.255.255.0
- DNS: 127.0.0.1

Após a instalação da máquina virtual e antes de se promover o Servidor a *Domain Controler*, é necessário mudar primeiro o nome do Servidor para *DC*, para isto é necessário ir ao "*Server Manager*", depois carregar em "*Local Server*" e de seguida no nome do computador. Depois carregamos em "*Change*" e no campo "*Name*" colocamos o nome *DC* (nome do Servidor Controlador de Dominio). Após a alteração do nome o mesmo necessitará de reiniciar, para aplicar esta alteração.

Após reiniciar, define-se um endereço IPv4 estático, neste caso o utilizado foi 192.168.1.5.

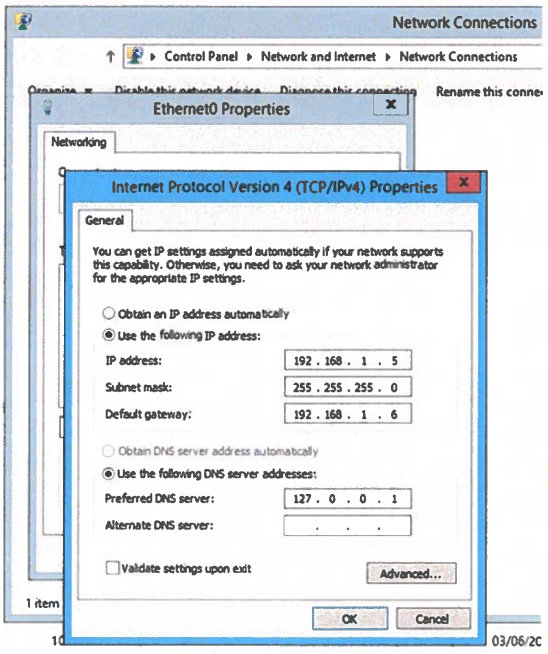

Figura 17 – Endereço *IPv4, Gateway e DNS Server.*

Fonte: do Autor

Depois destas alterações chegou o momento de se instalar a *Active Directory*, o *DNS Server*, promovendo a máquina Virtual *DC* a controlador de Domínio. O Método escolhido para ser utilizado foi a instalação através do *Windows PowerShell*, para este fim foi utilizado o seguinte comando de *PowerShell:*

Add-WindowsFeature AD-Domain-Services -IncludeManagementTools

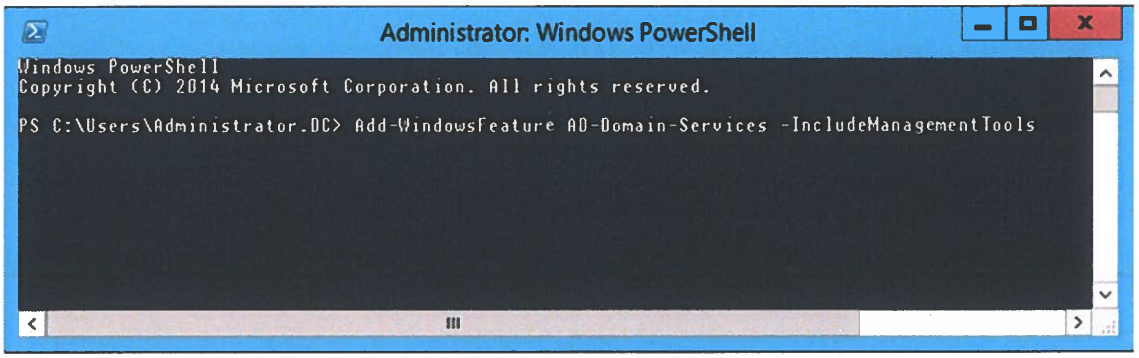

Figura 18 – Instalação da *Active Directory* Via *PowerShell.*

Fonte: do Autor

O passo seguinte consiste na criação de um domínio numa nova floresta e na promoção deste servidor a controlador de domínio. Vai ser criado o domínio "**istec.local**" numa nova floresta, o comando *PowerShell* para efetuar estas ações é:

Install-ADDSForest -DomainName "istec.local"

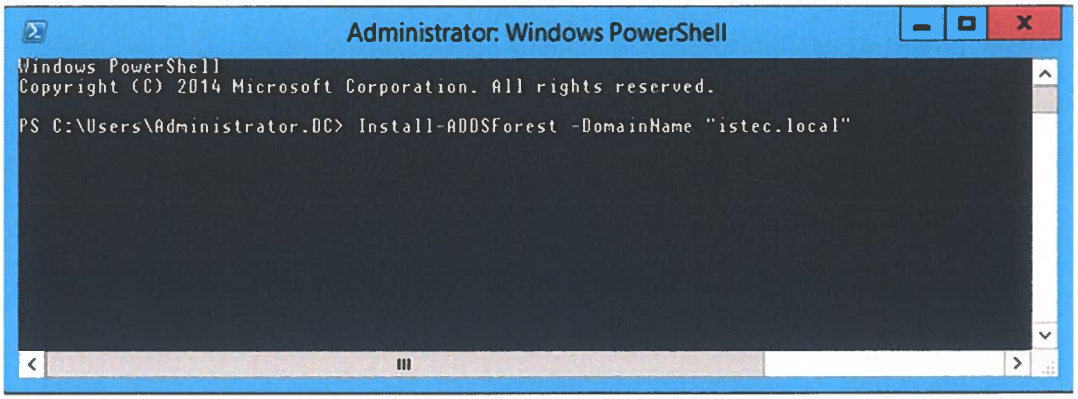

Figura 19 – Promoção a *Domain Controler*, criação de um domínio e de uma floresta.

Fonte: do Autor

É ainda necessário colocar uma *Password*, caso seja necessário efetuar um *restore* da *Active Directory* é necessário utilizar a Password para se efetuar o *restore* completo do *Domain Controler*.

Após a promoção a *Domain Controler*, criação de um domínio e de uma floresta, é necessário reiniciar a máquina virtual, depois de reiniciar começa-se a utilizar a conta de Domínio, neste caso ISTEC\A*dministrator*. A partir deste momento é possível através do *Domain Controler* criar contas de em domínio, para isto basta ir ao *Server Manager*, *Dashboar, tools, Active Directory Users and Computers* e escolhemos onde e que tipo de *user* queremos criar.

Routing

Descrição do Servidor *Domanin Controler* usado:

- Nome: Routing
- Endereço de IP (IPv4): 192.168.1.6
- Máscara de sub-rede: 255.255.255.0
- DNS: 192.168.1.5

De recordar que a Máquina Virtual *Routing* tem uma Placa de Rede *NAT* e uma *Host-only, Host-only* será uma placa privada que vai servir para a comunicação com as outras máquinas, a segunda placa de rede será uma *NAT* para fazer a conexão à rede externa das máquinas virtuais (para aceder à internet).

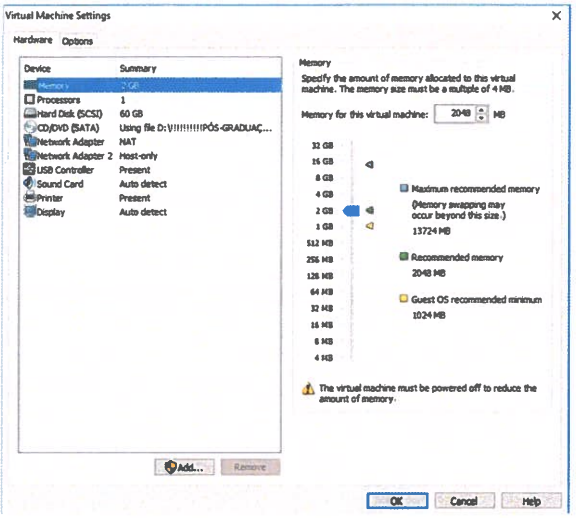

Figura 20 – Especificações da máquina *Routing*.

Fonte: do Autor

Após a Instalação da Máquina Virtual *Routing*, e da adição das placas de rede, como foi indicado anteriormente, é altura de adicionar a Máquina Virtual *Routing* ao domínio, para isto é necessário na placa de rede *Host-only* colocar um IP 192.168.1.6, Máscara de sub-rede 255.255.255.0 e *DNS* 192.168.1.5 (que é o IP da Máquina DC).

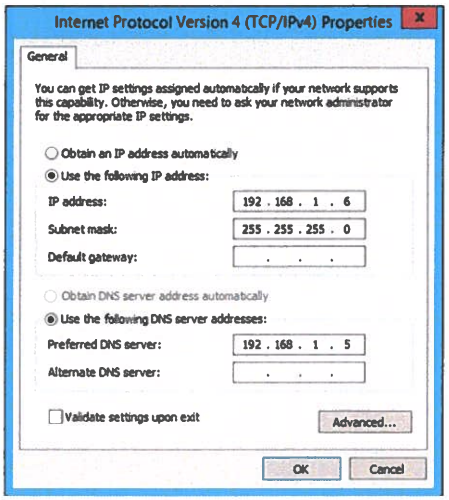

Figura 21 – Configuração da placa de Rede *Host-only*.

Fonte: do Autor

Depois já é possível adicionar a máquina ao domínio e também se pode mudar o nome da mesma para isto, basta ir a *Server Manager,* depois a *Local Server* e clicar em *Domain*. Ao Carregar em *Domain* irá surgir uma janela com o nome *System Properties,*

onde teremos um botão *Change,* ao carregar no botão *Change* irá surgir então a janela para alterar o nome da Maquina Virtual, e nessa mesma janela é então possível colocar a Máquina Virtual *Routing* em domínio (istec.local).

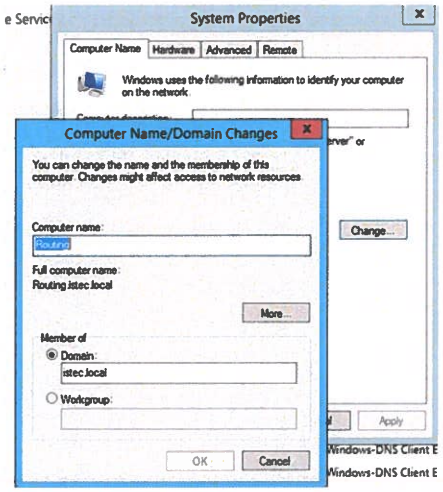

Figura 22 – Mudar o nome da Máquina Virtual e adição ao Domínio.

Fonte: do Autor

Neste momento e depois da Máquina Virtual *Routing* esta estar em domínio, é necessário instalar a *Role Routing and Remote Access*, de forma a que seja possível ter acesso do exterior à rede e de forma a que as restantes máquinas tenham acesso ao exterior.

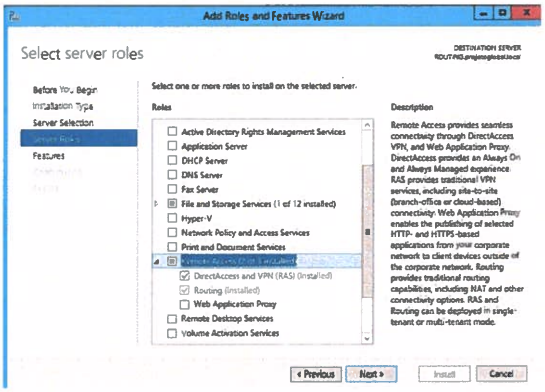

Figura 23 – Adição da Role *Routing and remote Access*.

Fonte: do Autor

Após a instalação desta *Role*, existem configurações necessárias que irão possibilitar então que toda a Rede interna tenha acesso à internet. Para realizar esta

configuração é necessário ir a *Server Manager*, depois a *Tools,* se seguida carregar em *Routing and Remote Access*. Clicar com o botão do lado direito do rato no nome do servidor carregamos em configurar, escolhemos a opção *NAT* (interface que possui ligação à internet), depois só é necessário carregar em *Next* e depois em *Complete*.

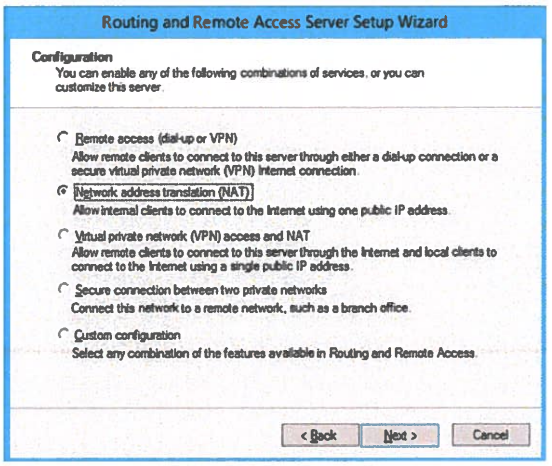

Figura 24 – Configuração do *Routing and Remote Access Setup*.

Fonte: do Autor

Neste Momento e Após estas configurações, o domínio em questão, istec.local, já se encontra com internet, por ultimo apenas é necessário colocar o IP do da Máquina Virtual *Routing* no *Default-Gateway* de todas as máquinas do domínio.

Para ser possível comunicar sem qualquer tipo de problema entre as Máquinas Virtuais do domínio, é necessário ativar algumas *Roles*. Então para serem ativas estas *Roles* basta carregar em *System and Security*, depois em *Windows Firewall*, *Advanced Settings*. Quando surgir a janela *Windows Firewall with Advanced Security*, carrega-se em *InBound Roles*, depois em *Filter by Group* e seleciona-se a opção *File and Printer Sharing* e ativam-se as que estão desativadas.

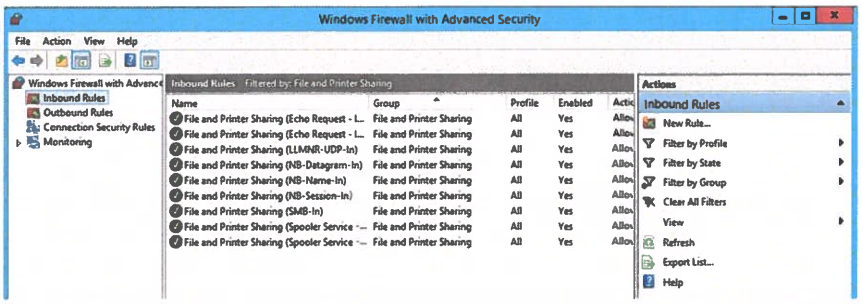

Figura 22 – Ativação nas *Roles* no *Windows Firewall*.

Fonte: do Autor

SQL

Descrição do Servidor *SQL* usado:

- Nome: SQL
- Endereço de IP (IPv4): 192.168.1.7
- Gateway: 192.168.1.6
- Máscara de sub-rede: 255.255.255.0
- DNS: 192.168.1.5

De recordar que a Máquina Virtual *SQL* tem uma Placa de Rede *Host-only*, *Host-only* será uma placa privada que vai servir para a comunicação com as outras máquinas dentro de uma rede privada, neste caso dentro do domínio istec.local.

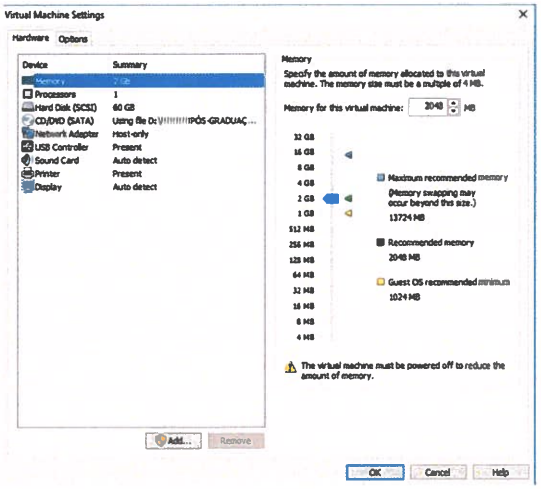

Figura 25 – Especificações da máquina *SQL*.

Fonte: do Autor

Após a Instalação da Máquina Virtual *SQL*, e da adição da placa de rede, como foi indicado anteriormente, é altura de adicionar a Máquina Virtual *SQL* ao domínio, para isto é necessário na placa de rede *Host-only* colocar um IP 192.168.1.7, Máscara de sub-rede 255.255.255.0, *Default-Gateway* 192.168.1.6 (IP do *Routing)* e *DNS* 192.168.1.5 (que é o IP da Máquina DC).

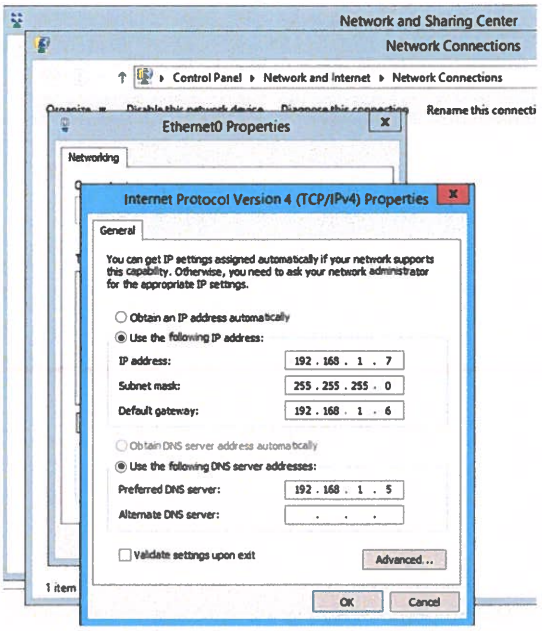

Figura 26 – Configuração da placa de Rede *Host-only*.

Fonte: do Autor

Depois já é possível adicionar a máquina ao domínio e também se pode mudar o nome da mesma, para isto, basta ir a *Server Manager,* depois a *Local Server* e clicar em *Domain*. Ao Carregar em *Domain* irá surgir uma janela com o nome *System Properties,* onde teremos um botão *Change,* ao carregar no botão *Change* irá surgir então a janela para alterar o nome da Maquina Virtual, e nessa mesma janela é então possível colocar a Máquina Virtual *SQL* em domínio (istec.local).

Para ser possível comunicar sem qualquer tipo de problema entre as Máquinas Virtuais do domínio, é necessário ativar algumas *Roles*. Então para serem ativas estas *Roles* basta carregar em *System and Security*, depois em *Windows Firewall*, *Advanced Settings*. Quando surgir a janela *Windows Firewall with Advanced Security*, carrega-se em *InBound Roles*, depois em *Filter by Group* e seleciona-se a opção *File and Printer Sharing* e ativam-se as que estão desativadas.

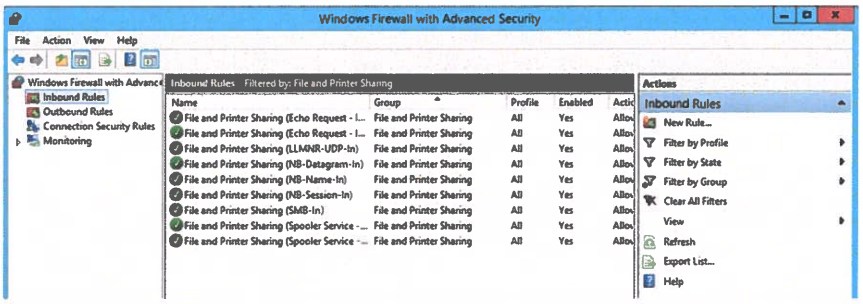

Figura 27 – Ativação nas Roles no Windows Firewall.

Fonte: do Autor

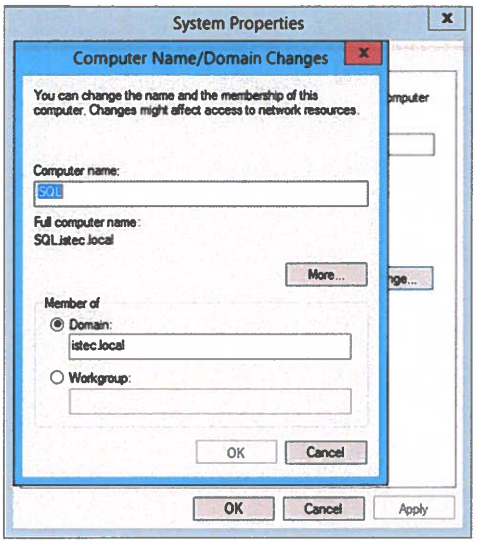

Figura 28 - Mudar o nome da Máquina Virtual e adição ao Domínio da Máquina SQL

Fonte: do Autor

Na Máquina Virtual *SQL* também vai ser necessário instalar o *Microsoft SQL Server Management Studio 17 e o SQL Server*. Após a instalação *Microsoft SQL Server Management Studio 17, SQL Server,* basta abrir o *Microsoft SQL Server Management Studio 17,* depois verificar se o *user* está em domínio.

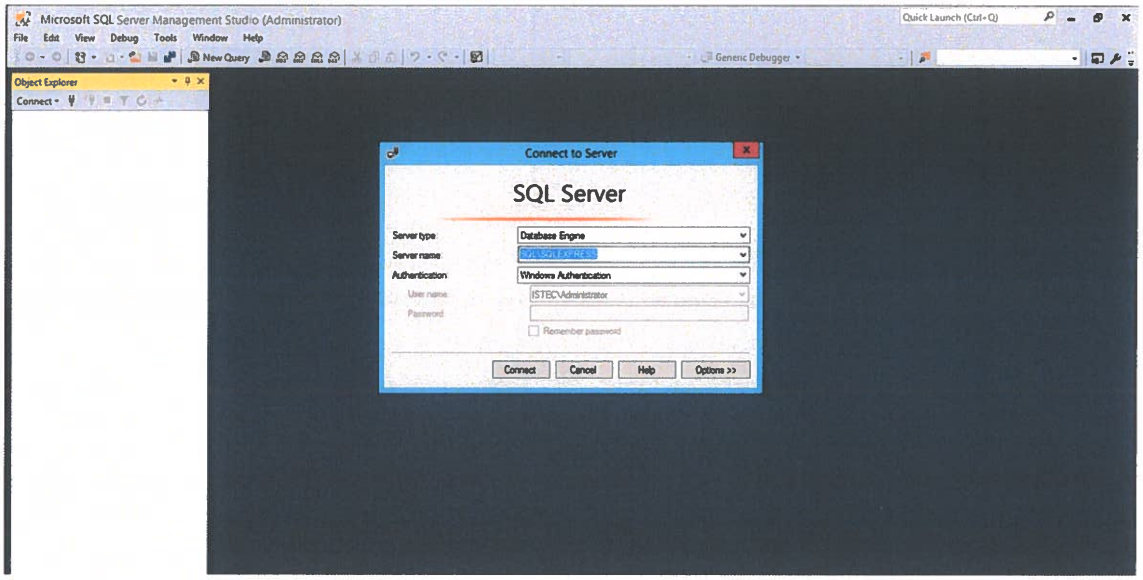

Figura 29 – Microsoft SQL Server Management Studio 17.

Fonte: do Autor

Máquina Client

Descrição do Servidor *SQL* usado:

- Nome: Cliente
- Endereço de IP (IPv4): 192.168.1.8
- Gateway: 192.168.1.6
- Máscara de sub-rede: 255.255.255.0
- DNS: 192.168.1.5

De recordar que a Máquina Virtual *Client* tem uma Placa de Rede *Host-only*, *Host-only* será uma placa privada que vai servir para a comunicação com as outras máquinas dentro de uma rede privada, neste caso dentro do domínio istec.local.

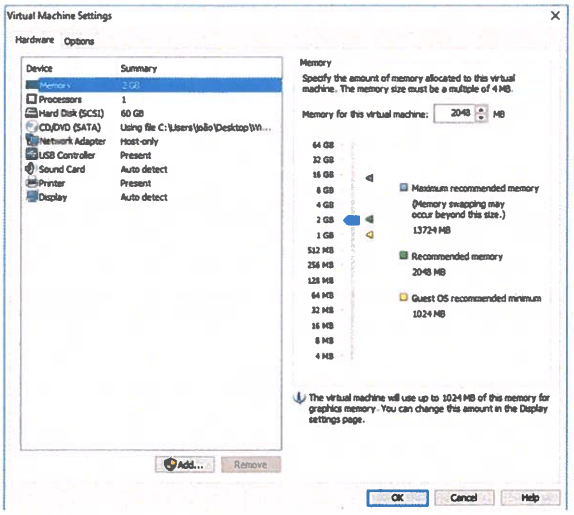

Figura 30 – Especificações da máquina *Client*.

Fonte do Autor

Após a Instalação da Máquina Virtual *Client*, e da adição da placa de rede, como foi indicado anteriormente, é altura de adicionar a Máquina Virtual *SQL* ao domínio, para isto é necessário na placa de rede *Host-only* colocar um IP 192.168.1.8, Máscara de sub-rede 255.255.255.0, *Default-Gateway* 192.168.1.6 (IP do *Routing)* e *DNS* 192.168.1.5 (que é o IP da Máquina DC).

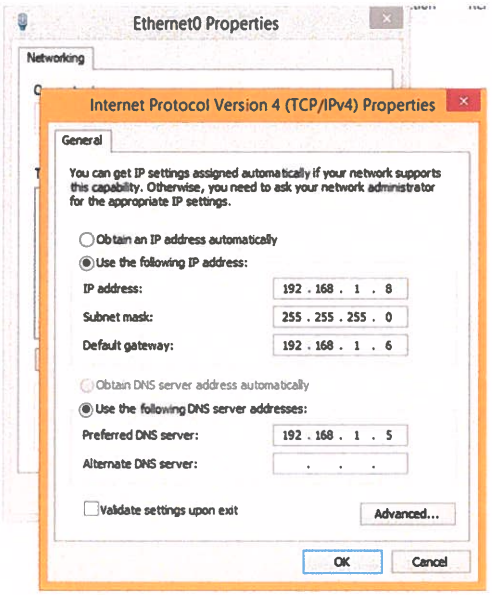

Figura 31 – Configuração da placa de Rede *Host-only*.

Fonte: do Autor

Depois já é possível adicionar a máquina ao domínio e também se pode mudar o nome da mesma, para isto, basta ir a *Control Panel*, depois a *System and Security,* de seguida a *System,* depois basta carregar em *Change* settings e na janela *System Properties,* onde teremos um botão *Change,* ao carregar no botão *Change* irá surgir então a janela para alterar o nome da Máquina Virtual, e nessa mesma janela é então possível colocar a Máquina Virtual Cliente em domínio (istec.local).

Para ser possível comunicar sem qualquer tipo de problema entre as Máquinas Virtuais do domínio, é necessário ativar algumas *Roles*. Então para serem ativas estas *Roles* basta carregar em *System and Security*, depois em *Windows Firewall*, *Advanced Settings*. Quando surgir a janela *Windows Firewall with Advanced Security*, carrega-se em *InBound Roles*, depois em *Filter by Group* e seleciona-se a opção *File and Printer Sharing* e ativam-se as que estão desativadas.

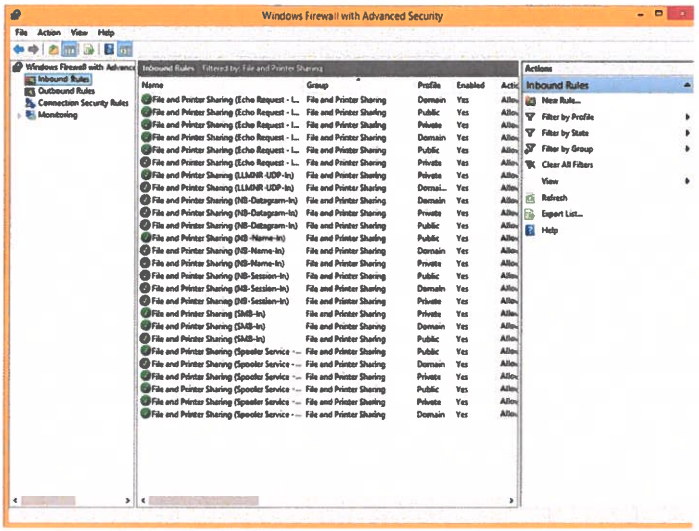

Figura 32 – Ativação nas *Roles* no *Windows Firewall.*

Fonte: do Autor

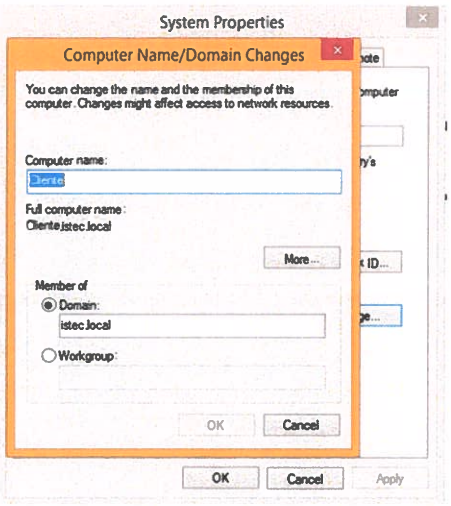

Figura 33 – Mudar o nome da Máquina Virtual e adição ao Domínio da Máquina Cliente.

Fonte: do Autor

Agora será então necessário a instalação do *Visual Studio* de forma a ser desenvolvida uma aplicação C# com ligação à base de dados. É possível aceder à máquina virtual *client* com a conta de administração de domínio, e com uma conta que foi criada no *Domain Controler* na *Active Directory Users and Computers*. Neste momento está na altura de ser efetuada uma ligação à base de dados da máquina virtual *Client à SQL*, onde está alojada a base de dados. Após esta ligação irá então ser iniciada a aplicação.

Figura 34 - Código do *Login*, do botão Autenticação.

Fonte: do Autor

O Código representado na figura 34 é referente à parte do *Login*, como é feito e como se processa a nível de validação dos campos. Na primeira parte irá ser verificado se a caixa de texto *txtusername*, se se encontra preenchida, caso a mesma não esteja preenchida é exibida uma mensagem a alertar essa mesma situação, com a seguinte frase: "Insira o seu *Username*". Após esta verificação e se for preenchido tanto o campo de Utilizador assim com o de *Password*, existe uma validação de forma a comparar com os utilizadores que estão na base de dados, caso a *password* ou o *username* estejam errados é exibida a seguinte um alerta: "A sua *Password* ou o seu *Username* estão incorretos!", se as credenciais estiverem corretas, o utilizador consegue então entrar na aplicação.

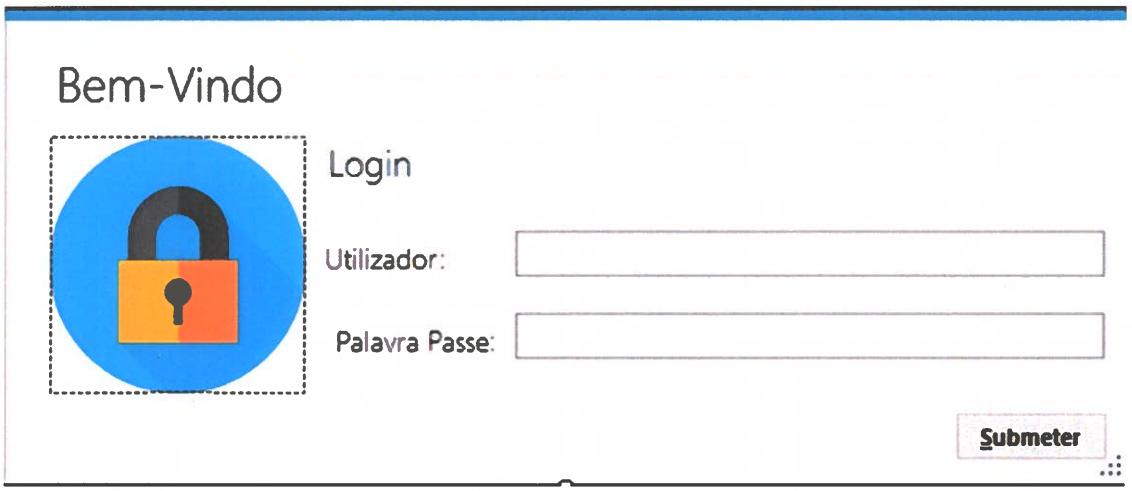

Figura 35 - *Layout do Login.*

Fonte: do Autor

Figura 36 - Código referente aos botões de Adicionar, Editar e Apagar.

Fonte: do Autor

Na Figura 36 é demonstrado o código referente aos botões Adicionar, Editar e Apagar. A primeira parte do código é referente ao botão Adicionar (btnAdicionar), ao carregar no botão adicionar é "disparado" um evento, surgindo uma nova janela sendo que nessa janela, será então possível adicionar os dados do filme (nome, género, descrição e imagem), que após se gravar são apresentados num *Gridview*. Na segunda parte do código está representado o botão Editar (btnEditar), ao carregar no botão editar é "disparado" um evento surgindo uma nova janela, em que a mesma terá todos os campos preenchidos referentes ao filme selecionado, onde será possível alterar qualquer campo (nome, género, descrição e imagem), após esta alteração basta carregar no botão guardar e é automaticamente atualizado na *Gridview* o registo do filme. A terceira parte do código é referente ao botão apagar (btnApagar), ao selecionar o registo que se pretende apagar e após carregar no botão apagar é exibido um alerta: "De certeza que pretende eliminar este Filme?", se se carregar que sim é eliminado o registo da base de dados e o mesmo deixa de ser exibido na *Gridview*.

```
//botão download para ir para uma página onde se efectuará o download do PDF
private void btnDownload_Click(object sender, EventArgs e)
{
    if (dadoBindingSource.Current == null)
        return;
    using (PDF frm = new PDF(dadoBindingSource.Current as dado))
    {
        if (frm.ShowDialog() == DialogResult.OK)
            dadoBindingSource.DataSource = db.dados.ToList();
    }
}
```

Figura 37 - Código referente ao botão *Download*.

Fonte: do Autor

Na Figura 37 é exibido o código do botão *download* (btnDownload), em primeira instância é necessário selecionar o registo para ser efetuado o *download* desse mesmo registo em formato PDF, depois ao carregar no botão irá ser "disparado" um evento surgindo uma nova janela onde será possível então efetuar o *Download* do registo em formato PDF.

Figura 38 - *Layout* da janela que contem a *Gridview*, o botão Adicionar, Editar, Apagar e *Download*.

Fonte: do Autor

```
private void metroButton1_Click(object sender, EventArgs e)
{
    Document document = new Document();
    Section section = document.AddSection();

    section.AddParagraph("Marca do automóvel é: " + txtNome.Text);

    section.AddParagraph("O Design do automóvel é: " + txtGenero.Text);

    section.AddParagraph("A potência do automóvel são: " + txtDescricao.Text);

    PdfDocumentRenderer renderer = new PdfDocumentRenderer(false, PdfFontEmbedding.Always);
    renderer.Document = document;
    renderer.RenderDocument();
    string pdfFilename = string.Format("Rekla-{0:dd.MM.yyyy_hh-mm-ss}.pdf", DateTime.Now);
    renderer.PdfDocument.Save(pdfFilename);
    Process.Start(pdfFilename);
}
```

Figura 39 - Código para gerar e organizar o PDF.

Fonte: do Autor

A primeira parte do código da Figura 39 organiza e faz a estrutura do PDF, com todos os campos do registo selecionado, a segunda parte e converte e cria um documento em formato PDF.

```
public AdicionarEditar(dado obj)
{
    InitializeComponent();
    db = new filmesEntities();
    //verificar a existencia de campos preenchidos, se já existe um registo criado
    if (obj == null)
    {
        dadoBindingSource.DataSource = new dado();
        db.dados.Add(dadoBindingSource.Current as dado);

    }

    else
    {
        dadoBindingSource.DataSource = obj;
        db.dados.Attach(dadoBindingSource.Current as dado);

    }

}
```

Figura 40 - Verificar se foi criado um registo anteriormente.

Fonte: do Autor

O código presente na Figura 40 vai verificar se já foi criado um registo do filme, caso não tenha sido criado é então criado um registo novo, se já tiver criado são mostrados os dados anteriormente inseridos na base de dados e na *Gridview*.

```
private void AdicionarEditar_FormClosing(object sender, FormClosingEventArgs e)
{
    if (DialogResult == DialogResult.OK)
    { //inserir e validar se os campos estão preenchidos
        if (string.IsNullOrEmpty(txtNome.Text))
        {
            MessageBox.Show("Porfavor introduza o nome do filme.", "Mensagem", MessageBoxButtons.OK, MessageBoxIcon.Information);
            txtNome.Focus();
            e.Cancel = true;
            return;
        }

        if (string.IsNullOrEmpty(txtGenero.Text))
        {
            MessageBox.Show("Porfavor indique o Genero do filme.", "Mensagem", MessageBoxButtons.OK, MessageBoxIcon.Information);
            txtGenero.Focus();
            e.Cancel = true;
            return;
        }

        if (string.IsNullOrEmpty(txtDescricao.Text))
        {
            MessageBox.Show("Porfavor digite a descrição do filme.", "Mensagem", MessageBoxButtons.OK, MessageBoxIcon.Information);
            txtDescricao.Focus();
            e.Cancel = true;
            return;
        }


        db.SaveChanges(); //guardar na base de dados
        e.Cancel = false;
    }
    e.Cancel = false;
}
```

Figura 41 - Inserir e validar se os campos estão preenchidos.

Fonte: do Autor

A Figura 41 representa a criação de um registo, em que caso não sejam preenchidas todas as caixas de texto (txtNome, txtGenero e txtDescricao), é automaticamente gerado um alerta, de forma a alertar que o campo que está em branco e que necessita de ser preenchido.

```
//selecção da imagem pretendida
private void pictureBox_Click(object sender, EventArgs e)
{
    dado obj = dadoBindingSource.Current as dado;
    if (obj != null)
        pictureBox.Image = Image.FromFile(obj.foto);
}

private void btnImagem_Click_1(object sender, EventArgs e)
{
    using (OpenFileDialog open = new OpenFileDialog() { Filter = "JPEG|*.jpg" })
    {

        if (open.ShowDialog() == DialogResult.OK)
        {
            pictureBox.Image = Image.FromFile(open.FileName);
            dado obj = dadoBindingSource.Current as dado;
            if (obj != null)
                obj.foto = open.FileName;
        }
    }

}
```

Figura 42 - Selecionar e mostrar a imagem.

Fonte: do Autor

Conclusão

Em função do objetivo de estudo chegou-se à conclusão de que, a virtualização é uma mais valia para o mundo empresarial atual. Hoje em dia está disponível uma vasta gama de tecnologias, que poderão não estar a ser utilizadas da forma mais eficiente, um desses caso é a virtualização. A partir de uma máquina física foram criadas quatro máquinas virtuais (*Domain Controler, Routing, SQL Server* e um *Client*), efetuando uma poupança de recursos e a nível energético.

O Objetivo proposto foi alcançado, consistindo na criação a partir de uma máquina física quatro máquinas virtuais, todas em domínio e todas a comunicar entre si. Estas máquinas virtuais são *Domain Controler, Routing, SQL Server* e um *Client*, acedendo a máquina virtual *Client* à base de dados presente na máquina virtual *SQL Server.*

Ao ser realizado este projeto, foi notória a viabilidade/utilidade que a virtualização tem nos dias que correm. Esta tecnologia oferece inúmeras vantagens, entre elas a segurança, a viabilidade, a disponibilidade e a otimização do *hardware* existente. O que permite comprovar que atualmente é uma excelente aposta tanto a nível económico, assim como a nível de eficácia e fiabilidade.

Bibliografia

Alves, G. (26 de 4 de 2013). *Você precisa saber o que é SQL!* Obtido de Dicas de Programação: https://dicasdeprogramacao.com.br/o-que-e-sql/

Armstrong, B. (24 de 1 de 2014). *Ben Armstrong's Virtualization Blog*. Obtido de Microsoft: https://blogs.msdn.microsoft.com/virtual_pc_guy/2014/01/24/managing-windows-server-2012-hyper-v-from-windows-8-1/

Blue Solution. (Setembro de 2014). *Como funciona a Virtualização de Servidores?* Obtido de Blue Solution: https://www.bluesolutions.com.br/2014/09/como-funciona-a-virtualizacao-de-servidores/

Dialogic Corporation. (2010). *Introduction to Cloud Computing.* Obtido de Dialogic Corporation: http://www.dialogic.com/~/media/products/docs/whitepapers/12023-cloud-computing-wp.pdf

Ivan. (2 de 3 de 2013). *SaaS, PaaS and IaaS: What are They?* Obtido de The Cloud Infographic: http://www.thecloudinfographic.com/2012/03/02/saas-paas-and-iaas-what-are-they.html

Microsoft. (2 de 3 de 2000). *Architecture Active Directory*. Obtido de Microsoft: https://technet.microsoft.com/fr-fr/library/bb967320.aspx

Microsoft. (26 de 6 de 2013). *Active Directory Domain Services Requirements*. Obtido de Microsoft: https://technet.microsoft.com/en-us/library/dd441359(v=office.13).aspx

Microsoft. (20 de 07 de 2015). *Introduction to the C# Language and the .NET Framework.* Obtido de Microsoft: https://docs.microsoft.com/en-us/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework

Microsoft. (31 de 8 de 2016). *Hyper-V overview*. Obtido de Microsoft: https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-R2-and-2012/hh831531(v=ws.11)

Microsoft. (13 de 5 de 2016). *Virtual Machine Manager*. Obtido de Microsoft: https://technet.microsoft.com/en-us/library/gg610610(v=sc.12).aspx

Microsoft. (30 de 3 de 2017). *ADO.NET*. Obtido de Microsoft: https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/index

Microsoft. (30 de 3 de 2017). *ADO.NET Overview*. Obtido de Microsoft: https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/ado-net-overview

Microsoft. (17 de 10 de 2017). *NIC Teaming*. Obtido de Mocrosoft: https://docs.microsoft.com/en-us/windows-server/networking/technologies/nic-teaming/nic-teaming#span-data-ttu-id56d3d-114a-namebkmkoveradescri%C3%A7%C3%A3o-geral-de-forma%C3%A7%C3%A3o-de-placa-de-redespanspan-classsxs-lookupspan-data-stu-id56d3d-114a-name

Microsoft. (11 de 7 de 2017). *What is Virtual Machine Manager?* Obtido de Microsoft: https://docs.microsoft.com/en-us/system-center/vmm/overview?view=sc-vmm-1801

Microsoft. (19 de 10 de 2017). *What's new in SQL Server 2017*. Obtido de Microsoft: https://docs.microsoft.com/en-us/sql/sql-server/what-s-new-in-sql-server-2017

Microsoft. (s.d.). *ADO.NET Architecture*. Obtido de Microsoft: https://msdn.microsoft.com/en-us/library/27y4ybxw(v=vs.71).aspx

Microsoft Azure. (s.d.). *What is IaaS?* Obtido de Microsoft Azure: https://azure.microsoft.com/en-us/overview/what-is-iaas/

Microsoft Azure. (s.d.). *What is PaaS?* Obtido de Microsoft Azure: https://azure.microsoft.com/en-us/overview/what-is-paas/

Microsoft Azure. (s.d.). *What is SaaS?* Obtido de Microsoft Azure: https://azure.microsoft.com/en-us/overview/what-is-saas/

Microsoft Azure. (s.d.). *What is SaaS?* Obtido de Microsoft Azure: https://azure.microsoft.com/en-us/overview/what-is-saas/

Microsoft. (s.d.). *O que é a Virtualização*. Obtido de Microsoft: https://azure.microsoft.com/pt-pt/overview/what-is-virtualization/

Microsoft. (s.d.). *Visual Studio*. Obtido de Microsoft: https://www.visualstudio.com/pt-br/vs/ide/

Pinheiro, F. (2012). *CLOUD COMPUTING*. Obtido de gta.ufrj: https://www.gta.ufrj.br/ensino/eel879/trabalhos_vf_2010_2/fernando/caracteristicas.html

Rosa, A. (2013). *Windows Server 2012.* Lisboa: FCA - Editora de Informática, Lda.

Rouse, M. (9 de 2016). *SQL (Structured Query Language)*. Obtido de techtarget: http://searchsqlserver.techtarget.com/definition/SQL

Rover, M. (6 de 2012). *O que é Active Directory, topologia física e lógica?* . Obtido de Microsoft: https://technet.microsoft.com/pt-br/library/jj206711.aspx

Vidal, J. (26 de 9 de 2010). *MVP in System Center Cloud and Datacenter Management*. Obtido de josuevidal: https://josuevidal.wordpress.com/2010/11/26/voc-conhece-o-dcpromo/

Anexos

Anexo 1 – Login

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Filmes
{
  public partial class Pag1 : MetroFramework.Forms.MetroForm
  {
    static Pag1 _instance;

    public static Pag1 Instance
    {    // neste ponto irá ser instanciado a pagina do Login
      get
      {
        if (_instance == null)
          _instance = new Pag1 ();
        return _instance;
      }
    }
    // neste ponto itam ser inicializado todos os componentes da pág, neste capo pag1
    public Pag1()
    {
      InitializeComponent();
    }
```

```
        private void Pag1_Load(object sender, EventArgs e)
        {

        }
        //validar preenchimento dos campo no Login
        private void btLogin_Click(object sender, EventArgs e)
        {  //validar se foi escrito o Username
           if (string.IsNullOrEmpty(txtusername.Text))
           {
              MetroFramework.MetroMessageBox.Show(this, "Insira o seu Username",
"Mensagem", MessageBoxButtons.OK, MessageBoxIcon.Error);
              txtusername.Focus();
              return;
           }
           try
           {  //validar se o User e a Password estão correctos
              using (filmesEntities db = new filmesEntities())
              {
                 var query = from u in db.autenticacaos
                             where u.username == txtusername.Text && u.password ==
txtpassword.Text
                             select u;
                if (query.SingleOrDefault() != null)
 //se as credenciais estiverem correctas o user loga-se e transita para a proxima página
                {
                   this.Hide();
                   Pag2 segunda = new Pag2();
                   segunda.ShowDialog();
                }
                else // se as credenciais não estiverem correctas é acionado o alerta
                   MetroFramework.MetroMessageBox.Show(this, "A sua Password ou o
seu username estão incorretos!", "Mensagem", MessageBoxButtons.OK,
MessageBoxIcon.Information);
              }
```

```
            }

            catch (Exception ex)
            {
                MetroFramework.MetroMessageBox.Show(this, ex.Message, "Mensagem",
MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
    }
}
```

Anexo 2 – *GridView*, botões Adicionar, Editar, Apagar e Downlod do PDF

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Diagnostics;
using PdfSharp;
using PdfSharp.Drawing;
using PdfSharp.Pdf;
using PdfSharp.Pdf.IO;
using MigraDoc.DocumentObjectModel;
using MigraDoc.Rendering;
```

```
namespace Filmes
{
    public partial class Pag2 : MetroFramework.Forms.MetroForm
    { //ligação à base de dados
        filmesEntities db;

        bool _sair;
        //inicialização de todos os componentes da página
        public Pag2()
        {
            InitializeComponent();
        }

        //botão Logout
        private void lnkLogout_Click(object sender, EventArgs e)
        {
            _sair = true;
            this.Close();
            Pag1.Instance.Show();
    }

        private void Pag2_FormClosed(object sender, FormClosedEventArgs e)
        {
            //main form closed
            if (!_sair)
                Application.Exit();
        }
        //popular dados da grid view
        private void Pag2_Load(object sender, EventArgs e)
        {
            db = new filmesEntities();
            dadoBindingSource.DataSource = db.dados.ToList();
```

```
        }
        //botão para adicionar filmes à base de dados
        private void btnAdicionar_Click(object sender, EventArgs e)
        {
            using(AdicionarEditar addEdit = new AdicionarEditar(null))
            {
                if (addEdit.ShowDialog() == DialogResult.OK)
                    dadoBindingSource.DataSource = db.dados.ToList();

            }
        }
        //botão para editar o dados preenchidos
        private void btnEditar_Click(object sender, EventArgs e)
        {
            if (dadoBindingSource.Current == null)
                return;
            using (AdicionarEditar addEdit = new
AdicionarEditar(dadoBindingSource.Current as dado))
            {
                if (addEdit.ShowDialog() == DialogResult.OK)
                    dadoBindingSource.DataSource = db.dados.ToList();

            }
        }

        //botão para apagar filmes da base de dados
        private void btnApagar_Click(object sender, EventArgs e)
        {
            if (dadoBindingSource.Current != null)
            { //confirmação se se quer apagar
                if(MessageBox.Show("De certeza que pretende eliminar este Filme?",
"Mensagem", MessageBoxButtons.YesNo, MessageBoxIcon.Question) ==
DialogResult.Yes)
                {
```

```
                db.dados.Remove(dadoBindingSource.Current as dado);
                dadoBindingSource.RemoveCurrent();
                db.SaveChanges();
            }
        }
    }
    //botão PDF
    private void btnDownload_Click(object sender, EventArgs e)
    {
        if (dadoBindingSource.Current == null)
            return;
        using (PDF frm = new PDF(dadoBindingSource.Current as dado))
        {
            if (frm.ShowDialog() == DialogResult.OK)
                dadoBindingSource.DataSource = db.dados.ToList();
        }
    }
}
```

Anexo 3 – Inserir, validação dos dados e adição da imagem.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Filmes
{
   public partial class AdicionarEditar : MetroFramework.Forms.MetroForm
   {
//ligação aos campos da base de dados
      filmesEntities db;

      public AdicionarEditar(dado obj)
      {
         InitializeComponent();
         db = new filmesEntities();
         //verificar a existencia de campos preenchidos, se já existe um registo criado
         if (obj == null)
         {
            dadoBindingSource.DataSource = new dado();
            db.dados.Add(dadoBindingSource.Current as dado);
         }
         else
         {
            dadoBindingSource.DataSource = obj;
            db.dados.Attach(dadoBindingSource.Current as dado);
```

```
        }
    }

    private void AdicionarEditar_FormClosing(object sender, FormClosingEventArgs
e)
    {
        if (DialogResult == DialogResult.OK)
        { //inserir e vallidaar se os campo estão preenchidos
            if (string.IsNullOrEmpty(txtNome.Text))
            {
                MessageBox.Show("Porfavor introduza o nome do filme.", "Mensagem",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                txtNome.Focus();
                e.Cancel = true;
                return;
            }

            if (string.IsNullOrEmpty(txtGenero.Text))
            {
                MessageBox.Show("Porfavor indique o Genero do filme.", "Mensagem",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                txtGenero.Focus();
                e.Cancel = true;
                return;
            }

            if (string.IsNullOrEmpty(txtDescricao.Text))
            {
                MessageBox.Show("Porfavor digite a descrição do filme.", "Mensagem",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                txtDescricao.Focus();
                e.Cancel = true;
                return;
            }
```

```
            db.SaveChanges(); //guaradr na base de dados
            e.Cancel = false;
        }
        e.Cancel = false;
    }
    //botão imagem

    //selecção da imagem pretendida
    private void pictureBox_Click(object sender, EventArgs e)
    {
        dado obj = dadoBindingSource.Current as dado;
        if (obj != null)
            pictureBox.Image = Image.FromFile(obj.foto);
    }

    private void btnImagem_Click_1(object sender, EventArgs e)
    {
        using (OpenFileDialog open = new OpenFileDialog() { Filter = "JPEG|*.jpg" })
        {

            if (open.ShowDialog() == DialogResult.OK)
            {
                pictureBox.Image = Image.FromFile(open.FileName);
                dado obj = dadoBindingSource.Current as dado;
                if (obj != null)
                    obj.foto = open.FileName;
            }
        }
    }
}
```

Anexo 4 – Criação do PDF

```
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Diagnostics;
using PdfSharp;
using PdfSharp.Drawing;
using PdfSharp.Pdf;
using PdfSharp.Pdf.IO;
using MigraDoc.DocumentObjectModel;
using MigraDoc.Rendering;


namespace Filmes
{

  public partial class PDF : Form
  {
    filmesEntities db;
    public PDF(dado obj)
    {
      InitializeComponent();
      db = new filmesEntities();
      if (obj == null)
      {
        dadoBindingSource.DataSource = new dado();
        db.dados.Add(dadoBindingSource.Current as dado);
      }
```

```
            else
            {
                dadoBindingSource.DataSource = obj;
                db.dados.Attach(dadoBindingSource.Current as dado);
            }

        }

        private void PDF_Load(object sender, EventArgs e)
        {

        }
        private void metroButton1_Click(object sender, EventArgs e)
        {
            Document document = new Document();
            Section section = document.AddSection();

            section.AddParagraph("Marca do automóvel é: " + txtNome.Text);

            section.AddParagraph("O Design do automóvel é: " + txtGenero.Text);

            section.AddParagraph("A potência do automóvel são: " + txtDescricao.Text);

            PdfDocumentRenderer renderer = new PdfDocumentRenderer(false,
PdfFontEmbedding.Always);
            renderer.Document = document;
            renderer.RenderDocument();
            string pdfFilename = string.Format("Rekla-{0:dd.MM.yyyy_hh-mm-ss}.pdf",
DateTime.Now);
            renderer.PdfDocument.Save(pdfFilename);
            Process.Start(pdfFilename);
        }
    }
}
```