

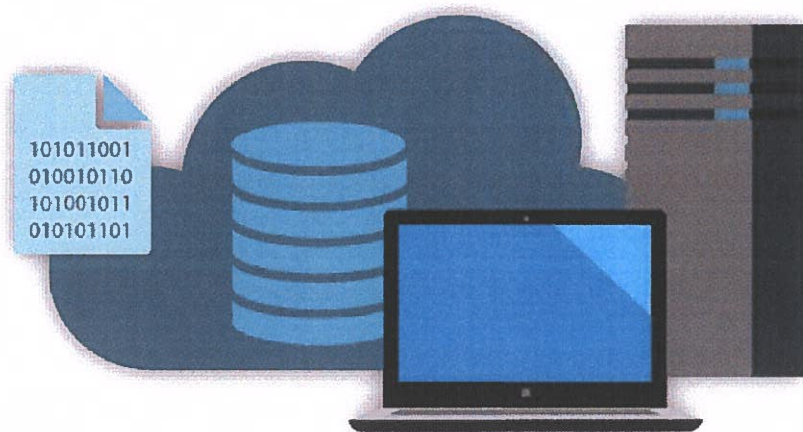
LICENCIATURA EM **INFORMÁTICA**

TURMA: **B**

**ACESSO A UMA BASE DE DADOS SQL ATRAVÉS DE UMA
APLICAÇÃO ADO.NET**

(IMPLEMENTAÇÃO EM AMBIENTE VIRTUAL)

ISAQUIEL LUZOLO BARROS JOÃO - 8130



PROJETO GLOBAL

PROFESSOR: **PEDRO BRANDÃO**

Lisboa, Setembro de 2018

DEDICATÓRIA

Dedico este trabalho primeiramente a mim mesmo porque auto motivei-me enfrentando todos os momentos possíveis enquanto estudante, dedico este trabalho à minha família que todo apoio prestou para o meu sustento aqui no território português. Em suma dedico este projeto a todos quanto depositaram um voto de confiança na minha formação/capacidade/qualidade.

AGRADECIMENTO

Neste ponto quero agradecer ao criador dos céus e da terra que pelo seu poder, abriu-me as portas para que hoje se concretize este grande momento na minha formação académica. Agradeço à minha família por todos os conselhos/apoios prestados em especial à minha irmã Zinha Caunde que tudo fez para apoiar-me sem cessar. Agradeço ao O Producer Family pelo apoio incalculável desde 2014. À todos quanto desejaram o melhor para mim direta ou indiretamente peço-vos que aceitem os meus agradecimentos.

Muito Obrigado!

RESUMO E PALAVRA-CHAVE

O objetivo deste projeto é criar uma aplicação em ADO.NET que tenha acesso a uma base de dados em SQL. A implementação e concretização do objetivo foi feita em um laboratório virtual.

Com este projeto conclui-se que a virtualização é uma componente muito importante, tal que, foi a base para a realização do laboratório, permitindo numa única máquina alojar outras máquinas de características diferentes, o SQL com a sua padronização universal permitiu criar e alojar a base de dados que é acedida pela aplicação e o aparecimento do ADO.NET o programador concebeu aplicação para que seja utilizada em rede (para ambiente virtual) com a funcionalidade de ser utilizada conectada ou desconectada da fonte de dados, *Cloud Computing* embora não concretizado na prática, esta tecnologia tem sido uma ferramenta útil para acesso a aplicações, máquinas virtuais e outros objetos de trabalho em qualquer sítio, sendo necessário somente na maioria dos casos o acesso à Internet.

Palavras-chaves: Virtualização, *Hypervisor*, *Cloud Computing*, SQL, ADO.NET, Máquina Virtual.

ABSTRACTS AND KEYWORDS

The project's goal is to create an ADO.NET application which it has access to an SQL database. The goal's concretization was made by a virtual lab.

This project can conclude that virtualization is an component very important, such as, it was the base to implement the lab, allowing one physical pc host others different virtual machines, SQL with its worldwide standardization allows to create and host the database accessed by the application and with ADO.NET the programmer developed an application that could be used in virtual network environment with capability to be used connect and disconnected from data source, Cloud Computing even not seen used in the project, it is being used as a useful tool to access applications, virtual machine and others objects anywhere, being necessary in most of the cases Internet access.

Keywords: Virtualization, *Hypervisor*, *Cloud Computing*, SQL, ADO.NET, Virtual Machine.

ÍNDICE

DEDICATÓRIA	i
AGRADECIMENTO.....	ii
RESUMO E PALAVRA-CHAVE	iii
ABSTRACTS AND KEYWORDS	iv
ÍNDICE.....	v
ÍNDICE DE TABELAS	vi
ÍNDICE DE IMAGENS.....	vii
INTRODUÇÃO	3
ESTADO DA ARTE.....	4
Virtualização	4
<i>Cloud Computing</i>	5
<i>SQL</i>	7
<i>ADO.NET</i>	9
CONTEXTUALIZAÇÃO.....	12
DESENVOLVIMENTO	13
Implementação do Ambiente Virtual (Laboratório).....	13
Criação das Máquinas Virtuais	15
Instalação do Sistema Operativo Windows Server 2012 R2 Standard	23
Configuração da Máquina Virtual para permitir Virtualização sobre ela	24
Promoção da Máquina Virtual DC a controlador de domínio	32
Criação de Conta de Utilizador no Domínio (SQL User)	35
Instalação do Microsoft SQL Server 2014 Standard Edition na máquina SQLSERVER ...	37
Criação da base de dados	43
Criação e desenvolvimento da aplicação ADO.NET	46
CONCLUSÃO	63
REFERÊNCIA BIBLIOGRÁFICAS.....	64

ÍNDICE DE TABELAS

Tabela 1 - Características físicas/virtuais das máquinas no laboratório	15
Tabela 2 - Configurações das máquinas virtuais	32

ÍNDICE DE IMAGENS

Imagem 1 – Hypervisor do Tipo 1	5
Imagem 2 – Hypervisor do Tipo 2.....	5
Imagem 3 – The Conceptual Reference Model	7
Imagem 4 – Modelo de Objetos de ADO.NET.....	11
Imagem 5 – Esquema do Laboratório implementado	13
Imagem 6 – Aba Home do VMware Workstation 14 Pro	16
Imagem 7 – Escolha do tipo de configuração da máquina virtual	16
Imagem 8 – Escolha da origem de instalação do sistema operativo.....	17
Imagem 9 – Escolha do sistema operativo e a versão a ser instalada	17
Imagem 10 – Atribuição do nome da máquina virtual e a sua localização.....	18
Imagem 11 – Escolha do Firmware para a máquina virtual	18
Imagem 12 – Configuração do processador.....	19
Imagem 13 – Atribuição da memória RAM	19
Imagem 14 – Escolha do tipo de rede a ser conectada	20
Imagem 15 – Escolha do tipo de disco para armazenamento	20
Imagem 16 – Seleção do disco de armazenamento	21
Imagem 17 – Atribuição da capacidade do disco de armazenamento	21
Imagem 18 – Especificação do ficheiro do disco e a sua localização	22
Imagem 19 – Resumo das configurações feitas para a máquina virtual	22
Imagem 20 – Máquina criada conforme as configurações	23
Imagem 21 – Escolha da fonte para instalação do sistema operativo.....	23
Imagem 22 – Instalação do sistema operativo	24
Imagem 23 – Passo 1 de configuração para a máquina virtual permitir Nesting virtualization	24
Imagem 24 – Passo 2 de configuração para a máquina virtual permitir Nesting virtualization	25
Imagem 25 – Instalação do Hyper-V Manager via PowerShell	25
Imagem 26 – Hyper-V Manager após a respetiva instalação	26
Imagem 27 – Criação de um switch virtual	26
Imagem 28 – Escolha do tipo de switch virtual a ser criado	27
Imagem 29 – Escolha do tipo de conexão para o switch virtual.....	27
Imagem 30 – Switch virtual criado com a configuração pretendida	28
Imagem 31 – Criação de uma máquina virtual	28
Imagem 32 – Atribuição do nome da máquina virtual e a respetiva localização	29
Imagem 33 – Escolha da geração para a máquina virtual.....	29
Imagem 34 – Atribuição da memória RAM	30
Imagem 35 – Escolha da conexão (switch virtual) para a placa de rede	30
Imagem 36 – Especificação do armazenamento virtual	30
Imagem 37 – Escolha da fonte para a instalação do sistema operativo	31
Imagem 38 – Resumo das configurações feitas para a máquina virtual	32
Imagem 39 – Verificação das três máquinas criadas	32
Imagem 40 – Autenticação à máquina virtual DC após a promoção para controlador de domínio	33
Imagem 41 – Propriedades da máquina virtual dentro do domínio criado	34
Imagem 42 – Adição das máquinas virtuais SQLSERVER e CLIENT no domínio	34

Imagem 43 – Verificação da máquina virtual SQLSERVER.....	35
Imagem 44 – Verificação da máquina virtual CLIENT.....	35
Imagem 45 – Criação de uma conta de utilizador no domínio	36
Imagem 46 – Atribuição da palavra-passe.....	36
Imagem 47 – Conta de utilizador criada na unidade organizacional	37
Imagem 48 – Escolha para instalação de uma instância nova do Microsoft SQL Server 2014	37
Imagem 49 – Aceitação dos termos de licença do Microsoft SQL Server 2014	38
Imagem 50 – Configuração de algumas regras para instalação do Microsoft SQL Server	38
Imagem 51 – Configuração de uma instância do Microsoft SQL Server 2014 Standard.....	39
Imagem 52 – Configuração de contas de utilizadores aos serviços do Microsoft SQL Server 2014 Standard	39
Imagem 53 – Configuração do modo de autenticação e adição de contas de serviço para a instância do Microsoft SQL Server.....	40
Imagem 54 – Resumo das configurações para a instalação do Microsoft SQL Server 2014 Standard Edition.....	40
Imagem 55 – Fim da instalação do Microsoft SQL Server 2014 Standard Edition	41
Imagem 56 – Autenticação ao servidor SQL Server com uma conta configurada para o efeito	41
Imagem 57 – Propriedades de conexão da conta autenticada e da instância do Microsoft SQL Server	42
Imagem 58 – Atribuição de funções à base de dados para a conta de serviço	42
Imagem 59 – Criação de uma nova base de dados	43
Imagem 60 – Definição do nome da base de dados.....	43
Imagem 61 – Base de dados, tabela e stored Procedures criadas	44
Imagem 62 – Atributos (campos) da tabela dbo.osmeusdados.....	44
Imagem 63 – Stored Procedure com os parâmetros e instruções para inserir e atualizar os dados na tabela	45
Imagem 64 – Stored Procedure com os parâmetros e instrução para consultar dados na tabela	45
Imagem 65 – Visualização dos serviços SQL Server em execução	46
Imagem 66 – Janela do Visual Studio 2015 Enterprise	46
Imagem 67 – Formulário inicial da aplicação.....	47
Imagem 68 – Formulário inicial da aplicação em execução após a verificação que está tudo correto/validado para acesso à base de dados	48
Imagem 69 – Caixa de diálogo apresentando uma mensagem de erro.	49
Imagem 70 – Formulário inicial da aplicação em carregamento para aceder ao formulário A MINHA CONTA	49
Imagem 71 – Formulário A MINHA CONTA no seu estado inicial.....	50
Imagem 72 – Execução da função osmeusdados()	51
Imagem 73 – Clique no botão sair para execução do evento btnSair_Click	51
Imagem 74 – Execução do método Genero()	52
Imagem 75 – Execução do método EstadoCivil()	53
Imagem 76 – Informação quando os dados são guardados.....	54
Imagem 77 – Verificação do registo inserido na base de dados	54
Imagem 78 – Informação quando os dados são atualizados.....	56

Imagem 79 – Verificação do registo atualiza na base de dados	56
Imagem 80 – Concretização do evento btnImagem_Click()	57
Imagem 81 – Concretização do evento btnImagem_Click()	57
Imagem 82 – Antes de executar o método Ver()	59
Imagem 83 – Após executar o método Ver()	59
Imagem 84 – Criação de um ficheiro .pdf após executar o método Imprimir()	61
Imagem 85 – Verificação do ficheiro .pdf criado após executar o método Imprimir()	61
Imagem 86 – Ficheiro .pdf aberto num leitor de ficheiros .pdf	62

INTRODUÇÃO

A implementação de um projeto com base em virtualização não é um processo tão simples quanto parece, mas a concretização efetiva deste processo exige do especialista conhecimentos sólidos que permitem garantir uma virtualização capaz de instanciar máquinas virtuais com capacidades de desempenho como se fosse uma máquina física. O acesso à uma base de dados alojada numa máquina virtual diferente da máquina que aloja a aplicação, é um dos feitos que permite concretizar a importância da tecnologia.

O objetivo deste projeto é desenvolver uma aplicação em ADO.NET que tenha acesso a uma base de dados SQL num ambiente virtualizado.

Neste projeto serão abordadas as seguintes tecnologias:

Virtualização processo que permite criar várias máquinas virtuais em uma ou mais máquinas físicas, partilhando recursos físicos e lógicos desta(s). *Cloud Computing* modelo de computação baseada na nuvem constituída por 5 características principais: Demanda em *Self-service*, acesso à rede em banda larga, espaço de recursos, rápida elasticidade e serviços mensuráveis. SQL linguagem padrão usada para acesso e manipulação dos dados em uma ou mais base de dados, realizando inúmeras operações nos dados registados nelas. SQL é normalmente usada pelos administradores de base de dados e programadores que escrevem scripts de integração de dados e analistas. ADO.NET é uma tecnologia .NET constituído por um conjunto de bibliotecas orientado a objeto que permite interagir com os recursos de dados.

ESTADO DA ARTE

Virtualização

Virtualização é um processo que permite criar instâncias isoladas de máquinas virtuais usando o *hardware* de uma máquina física. Este processo é uma evolução dos antigos *mainframes* que eram compartilhados por vários utilizadores num espaço de aplicações distintas. No período de 1980 a 1990 o processo de *mainframes* em 1970, foi ultrapassado com a criação e existência de computadores pessoais. Hoje em dia este processo tende a crescer cada vez mais. Para além de ser um processo para vários acessos em simultâneo, a Virtualização tem os principais interesses na segurança, confiabilidade, disponibilidade, custo, adaptabilidade, balanceamento de carga e suporte a aplicações. (Mattos, s.d.)

O conceito de virtualização não é novidade para o mundo das tecnologias da informação. Na década de 1960 começaram os processos para a construção de ambientes de máquinas virtuais quando a IBM (International Business Machines) desenvolveu o seu sistema operativo experimental M44/44X. foi a partir daí que a IBM desenvolveu mais sistemas para o suporte da virtualização, dentre eles o OS/370 (Goldberg 1973, Golcberg and MAger 1979). Na era, a implementação de um ambiente individual completo para cada utilizador com independência e separado de outros utilizadores é um processo dominante para ser fornecido. (Laureano & Maziero, s.d.)

A virtualização é um mecanismo de processamento virtual com base num *Software* para os efeitos. Ela pode ser aplicada para servidores, armazenamento, aplicações e redes, sendo este um método eficaz para a redução de despesas para tecnologia de informação e também oferece uma elevada eficiência e agilidades para variadas empresas. São benefícios a rápida implantação de cargas de trabalho, o maior desempenho e disponibilidades, automatização das operações. Todos estes benefícios resumem-se num processo simples para o seu gerenciamento, barata e fácil para operar. (VMmare, s.d.)

O processo da virtualização na sua implementação prática é concretizado um *software* de nome *Hypervisor*. *Hypervisor* é uma camada (intermediária) entre a máquina física e a máquina virtual. Este faz a gestão de todo o processo virtualizado, de maneiras a garantir um bom funcionamento da virtualização. Existem dois tipos de *Hypervisors*: Tipo 1 (Nativo/*Bare Metal*): é aquele que é implementado diretamente sobre o *hardware* da máquina física. Estes podem ser VMware ESX e ESXi, Microsoft Hyper-V, Citrix XenServer, Oracle VM. E Tipo 2 (Hospedado): é aquele que é implementado sobre um sistema operativo previamente instalado

e configurado sobre o *hardware* da máquina física. Estes podem ser VMware Server, Microsoft Virtual, Oracle VM VirtualBox, Red Hat Enterprise Virtualization, Parallels Desktop. (Informática.com, s.d.)

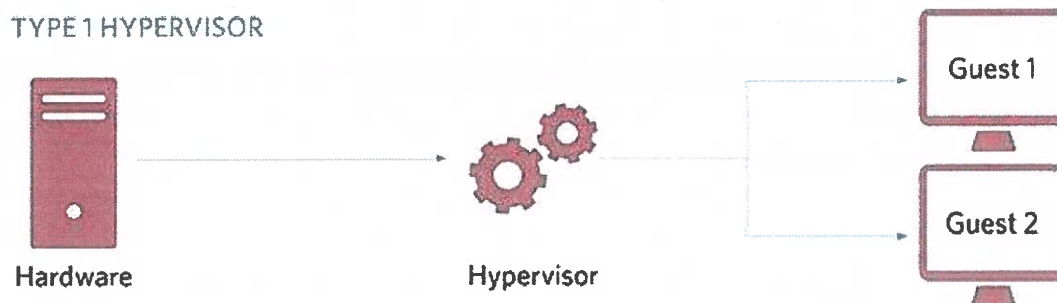


Imagem 1 – *Hypervisor* do Tipo 1.
Fonte: Margaret Rouse (2016)

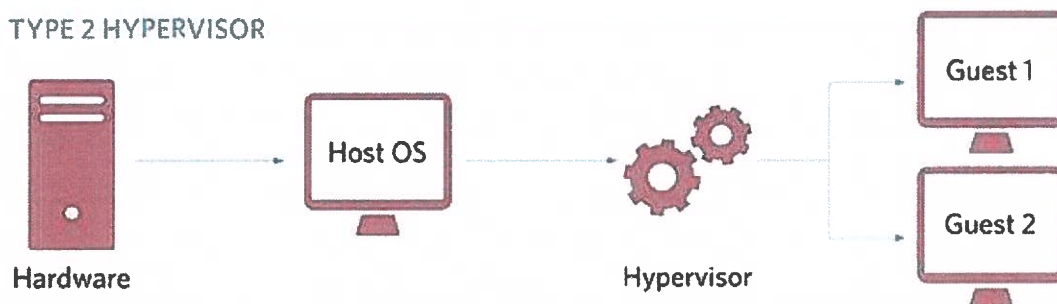


Imagem 2 – *Hypervisor* do Tipo 2.
Fonte: Margaret Rouse (2016)

A virtualização oferece também vantagens no melhor aproveitamento do *hardware*, dado que um dispositivo como uma placa de rede pode ser usado para várias máquinas virtuais. Na possibilidade de poder usar vários sistemas operativos de diferentes fabricantes em máquinas virtuais isoladas. Tal como nos outros processos, a virtualização também desvantagens como a sobre carga de tarefas, onde o excesso de máquinas virtuais ligadas simultaneamente pode causar mau desempenho da própria virtualização. Segurança, uma falha sobre o suporte do *Hypervisor*, é a razão para parar com a virtualização. (Oficina da Net, s.d.)

Cloud Computing

Cloud Computing é um modelo que permite o acesso em qualquer lugar, com exigência de um espaço partilhado de recursos computacionais configuráveis como rede, servidores, aplicações, entre outros, desde que estejam rapidamente provisionados e lançados com um mínimo de esforço de gestão e ou interação com o fornecedor do serviço. *Cloud Computing* é um modelo com as seguintes características principais: Demanda em *Self-service*, acesso à rede em banda larga, espaço de recursos, rápida elasticidade e serviços mensuráveis. *Cloud*

Computing é composto pelos serviços seguintes: *Software as a Service*, *Platform as a Service* e *Infrastructure as a Service*. Existem quatro modelos de implementação: *Private Cloud*, *Public Cloud*, *Community Cloud* e *Hybrid Cloud*. (Mell & Grance, s.d)

O conceito de *Cloud Computing* já tinha idealizadores desde os anos 60.

John McCarthy (citado por Silva, 2014) diz que:

a computação deve algum dia, ser organizada como um serviço de utilidade pública, bem como o sistema telefônico. Cada contribuinte precisará pagar apenas pela capacidade que utilizar de fato, mas possuirá acesso a todas as linguagens de programação características de um sistema muito amplo. (McCarthy, 1961, para: 10).

Como se pode perceber, o conceito de *Cloud Computing* utilizado hoje em dia como um serviço de pagamento de acordo com o uso (Pay as you use) já era idealizado em 1961. Joseph Carl Robnett Licklider (citado por Silva, 2014) diz que uma rede de computadores intergaláctica em que todos estariam conectados, tendo acesso aos programas e dados de qualquer lugar. (Joseph Carl Robnett Licklider, 1962, para: 14)” (Silva, 2014)

Cloud Computing usa as Taxonomias para cada tipo de serviço que providenciam informações nos serviços de *Cloud Computing* e cria um diálogo entre o Software de *Cloud Computing* e vendedores de produtos e assim como os desenvolvedores para aprofundar o entendimento e adoção das soluções de *Cloud Computing*. (Taxonomy, s.d.)

Sendo *Cloud Computing* um modelo de nuvem onde o foco está focado na redução custos e exigências operacionais para o elemento da tecnologia de informação e pela sua alta qualidade nos serviços oferecidos para os utilizadores, o Instituto de Padrões e Tecnologias do Governo Americano definiu 5 atores principais dentro do contexto e mercado de *Cloud Computing*: Cliente, uma entidade que mantém uma relação de negócio e usufrui de serviços prestados por um certo provedor/fornecedor de serviços *cloud*; Provedor, uma entidade que fornece os serviços *cloud* à clientes; Auditor, um terceiro que pode conduzir avaliação independente do serviço de *Cloud*, operações do sistema de informação, desempenho e segurança da implementação da *Cloud*; *Broker* uma entidade que gerência o uso, desempenho e entrega dos serviços de *Cloud* e negocia relações entre o provedor e o consumidor de *Cloud*; Transmissor de *Cloud*: o intermediário que providência conectividade e transporte dos serviços de *Cloud* a partir do Provedor e o Consumidor de *Cloud*. (Rosa, s.d.)

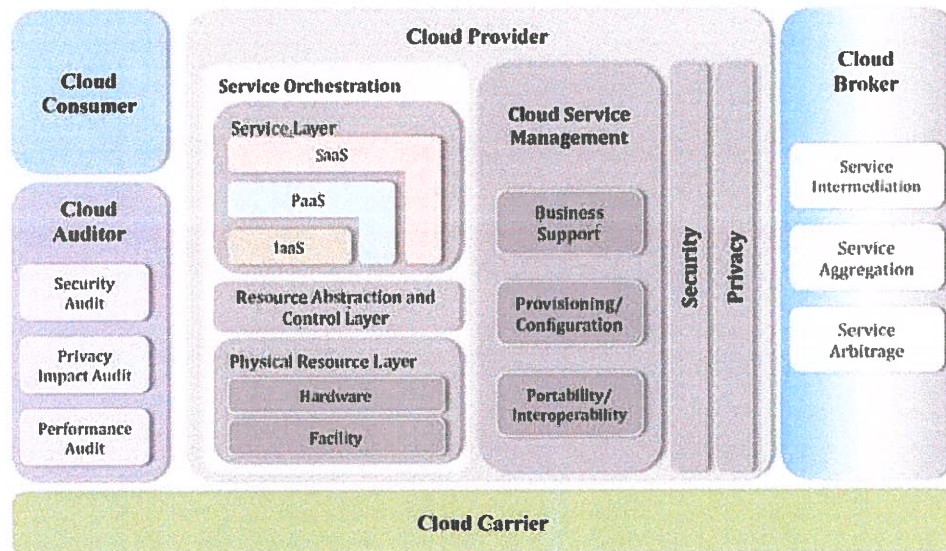


Imagem 3 – The Conceptual Reference Model
 Fonte: NIST, (s.d)

São os benefícios de *Cloud Computing* a redução de custos que permite poupar valores monetários de maneiras que só paga pelo que usa; o aumento de armazenamento onde o cliente só precisa utilizar o que realmente precisa e alargar ou reduzir quando necessário; automatização um dos grandes fatores que auxilia os técnicos, deixando-os sem a preocupação da manutenção dos equipamentos; mobilidade, com a infraestrutura armazenada na nuvem, só é preciso uma conexão à Internet para aceder ao sistema. *Cloud Computing* tem as seguintes limitações: segurança e privacidade dos dados dado que os dados não estão diretamente sobre a infraestrutura física do cliente, preocupa o seu estado atual de segurança; localização onde preocupa o local onde os dados são armazenados e como estão armazenados; a dependência da Internet também é um fator importante embora a tendência atual é a garantia e segurança do acesso a Internet; a migração de aplicações de negócio não é um processo simples de se realizar na maioria das vezes. (Antonio, s.d.)

SQL

SQL (*Structured Query Language* – Linguagem de Consulta Estruturada) é uma linguagem padrão usada para acesso e manipulação dos dados em uma ou mais base de dados, realizando inúmeras operações nos dados nela. SQL é normalmente usada pelos administradores de base de dados e programadores que escrevem scripts de integração de dados e analistas. SQL é usada para inserir, atualizar, selecionar, apagar dados contidos em linhas das tabelas, assim como a estruturação e reestruturação dos campos das tabelas. (Rouse, s.d.)

SQL como uma linguagem padrão para acessos e manipulações de dados e base de dados, tornou-se um padrão do American National Standards Institute (ANSI) em 1986 e da Organização Internacional de Normalização (ISO) em 1987. Esta linguagem executa consultas, recuperações, atualizações, inserções, divisões de dados, tabelas e bases de dados. Existem versões distintas da linguagem SQL, mas os comandos principais como *SELECT*, *UPDATE*, *DELETE*, *INSERT* funcionam de igual modo para todas as versões, diferenciando-se na sua sintaxe. As bases de dados são internamente geridas por um RDBMS (*Relational Database Management System* – Sistema de Gestão de Base de Dados Relacional). Como Microsoft SQL Server, IBM DB2, Oracle, MySQL e Microsoft Access. (W3Schools, n.d.)

A conceção do SQL permite aos utilizadores criarem bases de dados, inserirem e manipularem os dados e a recuperação de partes seleccionadas dos dados. Existem duas formas para concretização destas operações: a comunicação direta com SGBD (Sistema de Gestão de Base de Dados) através de uma linguagem de consulta e por intermédio de uma aplicação que envia instruções previamente definidas para o SGBD, que por sua vez executa as operações recebidas. SQL não é case sensitive (não é sensível a maiúsculo), sendo somente necessário uma correta sintaxe e alguns sistemas exigem que no final de cada instrução exista um ponto e vírgula. A linguagem SQL pode ser dividida em três partes principais: *DML (Data Manipulation Language – Linguagem de Manipulação de Dados)*: é utilizada para realizar operações como inclusões, consultas, alterações e exclusões de dados existentes num dado registo ou conjunto de registos e podem ser executadas em várias tabelas/bases dados diferentes ao mesmo tempo. A execução de cada operação obedece uma sintaxe própria (*Insert*, *Select*, *Update* e *Delete*); *DDL (Data Definition Language - Linguagem de Definição de Dados)*: as operações deste subconjunto permitem criar/alterar/apagar tabelas e seus elementos associados. São os seguintes comandos básicos da DDL: *CREATE*, *ALTER* e *DROP*; *DCL (Data Control Language - Linguagem de Controle de Dados)*: as operações deste subconjunto controlam os aspetos de autorização de dados e licenças dos utilizadores para controlar os devidos acessos para manipular os dados numa base de dados. As palavras-chaves da DCL são: *GRANT*, *REVOKE*. (Gomes, s.d.)

A grande utilização do SQL é fundamentada no modelo relacional de Codd (1970). A sua primeira versão teve como nome de SEQUEL (*Structured English Query Language*), definida por D. D. Chamberlin, em 1974 nos laboratórios de pesquisa da IBM (Califórnia). Um ano depois foi implementado um protótipo de aplicação dessa nova linguagem. Dois anos depois da sua revisão e ampliação, o nome SEQUEL foi alterado para SQL, por razões

jurídicas. A linguagem SQL foi inicial e especialmente concebido para ambiente relacional, podendo ser adaptado para um outro, tal como se vê nos dias de hoje. SQL hoje assume um papel muito importante nos SGBD, podendo ser usada como linguagem interativa de consulta (query AdHoc), linguagem de programação para acesso à base de dados (local e ou para outros servidores), linguagem cliente/servidor. Com SQL pode-se manipular objetos de classes distintas entre as funções do SGBD. Com SQL beneficia-se em vários termos. SQL como uma linguagem universal para os principais Sistemas de Gestão de Bases de Dados, ou seja, uma boa parte da linguagem é igual em todos dos Sistemas, embora haja alguns acréscimos ou ligeira regra na escrita de Sistema para Sistema, SQL não é case sensitive (não é sensível a letra Maiúscula/Minúscula); A portabilidade entre computadores; Redução de custos com treinamento; Inglês estruturado de alto nível; Consulta interativa; Múltiplas visões dos dados; Definição dinâmica dos dados. A limitação do SQL vê-se pela padronização que leva a uma limitação da criatividade, pois quem desenvolve aplicações fica limitado a soluções padronizadas. (Freitas, 2011)

ADO.NET

ADO.NET é uma tecnologia .NET constituído por um conjunto de bibliotecas orientado a objeto que permite interagir com os recursos de dados. Normalmente, a origem/fonte de dados é uma base de dados, como também poderia ser um ficheiro de texto, folha de cálculo do Excel ou mesmo um ficheiro XML. Existem vários *Data Providers* que permitem a conexão para diferentes recursos de dados, isto com base nos protocolos ou bases de dados a serem usados. ADO.NET é composto por vários objetos que trabalham com dados como: *SqlConnection* objeto que permite a gestão de uma conexão com a fonte de dados; *SqlCommand* o objeto que permite a comunicação entre o utilizador e a fonte de dados, enviando instruções para a fonte de dados; *SqlDataReader* é o objeto que oferece o acesso somente de leitura para os dados; *Dataset* o objeto que funciona de forma desconectada da fonte original dos dados, implementando assim o processo de leitura e escrita de e para fonte dos dados com auxílio do *SqlAdapter*. (Mayo, 2016)

ADO.NET (ActiveX Data Object .Net) é uma biblioteca de software na *framework* .NET está concebida para possibilitar os desenvolvedores códigos para obterem acessos desconectados às fontes de dados, o qual podem ser relacionais ou não relacional. Esta tecnologia fornece acesso conectado à uma conexão de base de dados usando o provedor de .NET-Managed, por outro lado, fornece acesso desconectado através dos *datasets* que são aplicações que usam as conexões de base de dados somente na consulta e atualização dos dados.

Dataset é um componente para armazenamento de dados persistentes na memória para fornecer acesso desconectado através de um recurso de base de dados eficientemente e com melhor escalabilidade. A arquitetura do ADO.NET é fundamentada por dois elementos primários: *Dataset* que fornece um conjunto completo de dados incluindo tabelas relacionadas, funcionalidades como acesso para dados XML remoto, manipulação dinâmica dos dados, processamento de dados de uma maneira desconectada, uso de ferramentas como XSLT e XPath Query para operar nos dados; e *.NET framework data Provider* que é composto por alguns dos seguintes componentes para manipulação de dados: *Connection* que fornece a conectividade a fonte de dados; *Command* que executa as instruções CRUD (*create, retrieve, update e delete*) de uma base de dados; *DataReader* componente de leitura dos dados contidos numa tabela e o *DataAdapter* que é o intermediário entre a fonte de dados e o *dataset*, carregando e ajustando as alterações feitas no *dataset* para a fonte de dados. (ActiveX Data Object.NET, s.d.)

ADO.NET é um conjunto de classes do .NET concebido para oferecer um acesso simples das aplicações aos variados motores de bases de dados como Microsoft SQL Server, Microsoft Access, Oracle e outros. Esta tecnologia surge do antigo ADO (ActiveX data Objects) que foi utilizado à o Visual Basic 6. O desenvolvimento do ADO.NET surge do nada, nem do antigo ADO, senão a sua abreviação. Sendo de fácil adaptação para os programadores em ADO para ADO.NET. O desenvolvimento desta tecnologia oferece as seguintes características: Acesso desconectado onde para manipular os dados não é necessária uma ativa conexão a fonte de dados; a integração com XML que simplifica a interoperabilidade entre superfícies heterogêneas; fontes de dados diversificados, o que significa que os dados podem surgir de várias fontes; Caching que permite o armazenamento de dados em cache melhorando assim o desempenho das aplicações ASP.NET. (Húngaro, s.d.)

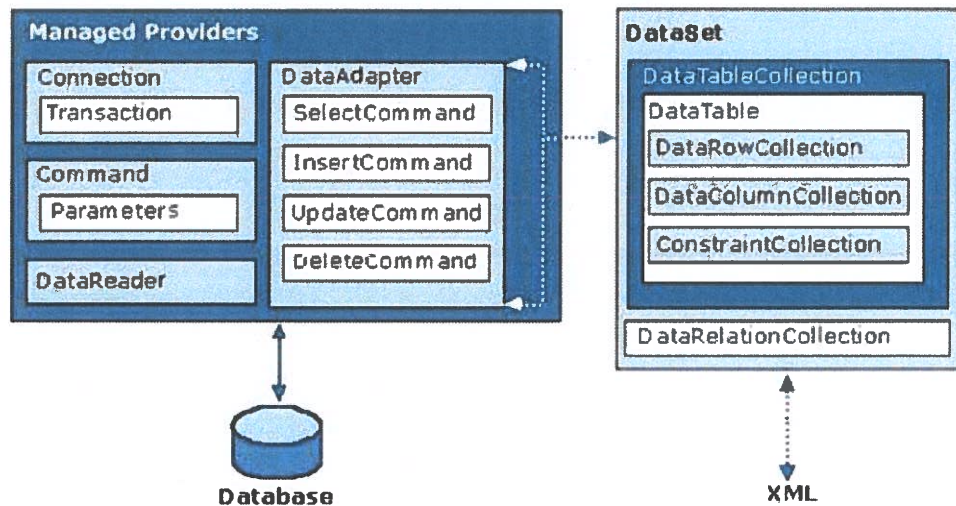


Imagem 4 – Modelo de Objetos de ADO.NET
 Fonte: Húngaro ibid, (s.d.)

ADO.NET é composto pelos seguintes *managed providers*: *OleDb*, *Odbc*, *OracleClient* e *SqlClient*. Cada objeto de um desses *Managed providers* escreve-se como no seguinte exemplo para um objeto *SqlClient* é *SqlConnection*; para o *OracleClient* é *OracleConnection* e adiante. A interoperabilidade, escalabilidade, produtividade e alto desempenho formam as vantagens da tecnologia ADO.NET. (Húngaro, s.d.)

A tecnologia ADO.NET fornece um conjunto de classes orientado a objeto, vários *data providers* funcionam com distintas fontes de dados, mas o modelo da programação para todos estes é igual. ADO.NET é de simples compreensão e uso pela sua natureza como orientado a objeto. As classes são gerenciadas, suporta o uso de XML que permite um acesso rápido de dados. O desempenho e a escalabilidade são fatores importantes a quando do desenvolvimento de aplicações web, quando desconectado com a fonte de dados, os dados em cache num ficheiro XML contribuem para um melhor desempenho e escalabilidade. (Mehra, 2009)

CONTEXTUALIZAÇÃO

A virtualização tem um grande desempenho e funcionalidade no mundo atual, tal que as entidades (empresas e particulares) tendem a fazer o uso desta tecnologia de maneiras que reduza o número físico de máquinas para funcionalidades na rede e outros fins.

Cloud Computing no mundo atual oferece infraestruturas tecnológicas que permite o acesso em qualquer lugar e momento dos seus recursos

Estas duas tecnologias estão a ganhar espaço no mercado informático devido às suas vantagens, tal que, muitas empresas estão no processo tendencial para migração da infraestrutura tradicional (física) para a moderna (*cloud*).

ADO.NET e SQL juntos, oferecem grandes benefícios para programação em conjunto com uma fonte de armazenamento de dados, permitindo assim criar uma instância de base de dados que tenha ligação com uma aplicação.

A junção destas quatro tecnologias permite aos *developers* e utilizadores finais usufruírem dos benefícios delas, de maneiras que mesmo estando separados fisicamente é possível haver acesso e comunicação do lado do servidor como do cliente.

DESENVOLVIMENTO

No desenrolar deste capítulo, serão apresentados os passos realizados para a concretização do objetivo do projeto que é através de uma aplicação ADO.NET numa máquina virtual aceder a uma base de dados SQL também alojada numa máquina virtual, ambas as máquinas pertencentes a uma rede com domínio.

Implementação do Ambiente Virtual (Laboratório)

A implementação deste laboratório foi feita usando um *Hypervisor* do Tipo 2, com base no seguinte esquema:

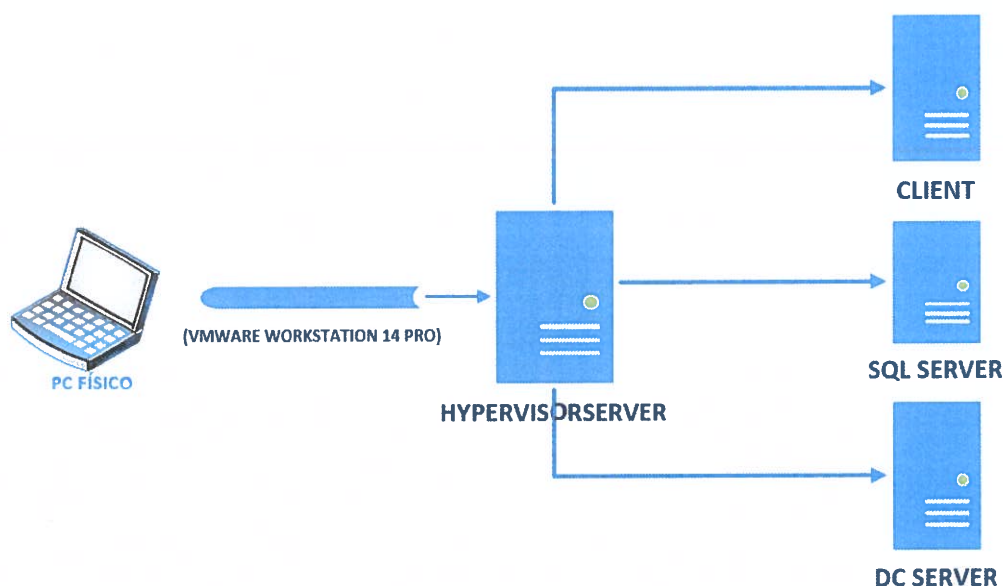


Imagem 5 – Esquema do Laboratório implementado

Fonte: do autor

Descrição do esquema acima:

PC FÍSICO é o computador físico usado para alojar a virtualização.

VMWARE WORKSTATION 14 PRO é o *software* que permitiu a criação da máquina virtual HYPERVISORSERVER, que por sua vez vai permitir virtualização do segundo nível para criação das máquinas CLIENT, SQLSERVER e DCSERVER.

HYPERVISORSERVER é máquina virtual que terá a funcionalidade de Hypervisor final, por que é nela onde serão criadas as máquinas CLIENT, SQLSERVER e DCSERVER.

DC SERVER é a máquina que terá a função de controlador de domínio na rede.

SQL SERVER é a máquina que vai alojar e servir os serviços do Microsoft SQL Server.

CLIENT é a máquina cliente da rede onde será testada a aplicação com acesso a base de dados alojada na máquina SQL SERVER, usando uma conta criada na máquina DC SERVER.

MÁQUINAS		PC FÍSICO	HYPERVISORS SERVER	DC SERVER	SQL SERVER	CLIENT
CPU		Intel(R) Core(TM)_i7-4710MQ CPU @ 2.50GHz				
RAM		16 GB	10 GB	1 GB	4 GB	1 GB
ARMAZENAMENTO	TIPO	SSD	ISCSI	VHDX	VHDX	VHDX
	CAPACIDADE	+250 GB	150 GB	<127 GB	<127 GB	<127 GB
INTERFACE DE REDE	TIPO	FÍSICO	NAT	PRIVATE	PRIVATE	PRIVATE
	NOME	Realtek PCIe GBE Family Controller	Intel(R) 82574L Gigabit Network Connection	Microsoft Hyper-V Network adapter	Microsoft Hyper-V Network adapter	Microsoft Hyper-V Network adapter
VIRTUALIZAÇÃO	VT	SIM	SIM	NÃO	NÃO	NÃO
	HYPERVISOR	VMware Workstation 14 Pro	HYPER-V	N/A	N/A	N/A

Tabela 1 - Características físicas/virtuais das máquinas no laboratório

Fonte: do autor

A tabela acima apresenta as características físicas e virtuais das máquinas pertencentes ao laboratório; estas características são suficientes para a concretização do objetivo pretendido.

Criação das Máquinas Virtuais

Com o *software* VMware Workstation 14 Pro instalado no PC FÍSICO criou-se a máquina HYPERVISORSERVER usando os seguintes passos:

Clicando sobre o botão Create a New Virtual Machine constante no espaço de trabalho da aba Home.

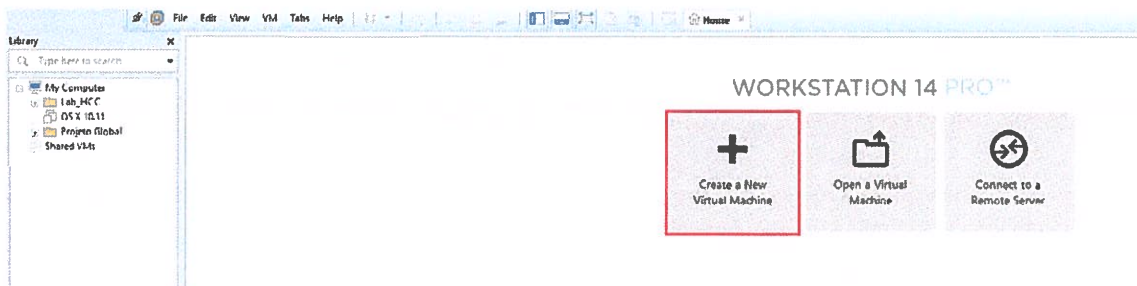


Imagem 6 – Aba Home do VMware Workstation 14 Pro
Fonte: do autor

Na caixa de diálogo que aparece escolhe-se a opção Costum (Advanced) e clicando em Next duas vezes (deixando assim as definições padrão sobre a compatibilidade do hardware da máquina virtual). A diferença destas duas opções é que a primeira tem passos reduzidos e mais simples para criação de uma máquina virtual, ao passo que a segunda permite ao criador configurar todo o hardware para a máquina virtual.

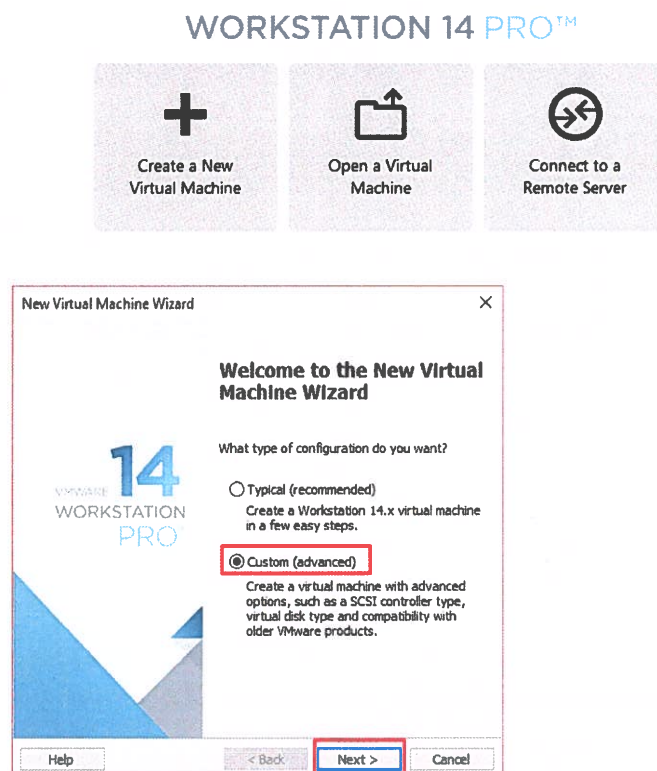


Imagem 7 – Escolha do tipo de configuração da máquina virtual
Fonte: do autor

Na secção do Guest Operating System Intallation, foi escolhida a opção I Will Install Operating System Later. É neste passo que o “especialista” (passo o termo) define quando e a fonte para a instalação do sistema operativo; neste caso foi escolhida a instalação mais tarde.

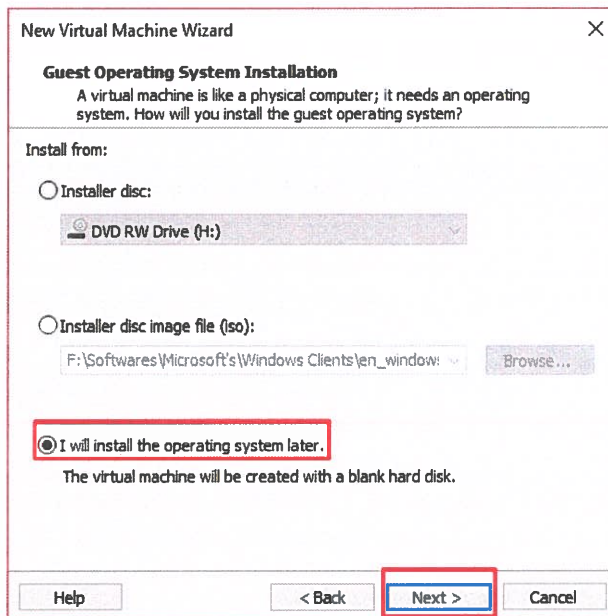


Imagem 8 – Escolha da origem de instalação do sistema operativo
Fonte: do autor

Na secção Select Guest Operating System, foi escolhida como Guest Operating System Microsoft Windows e a versão Windows Server 2012. Neste passo define-se o sistema operativo e a respetiva versão a ser instalada na máquina virtual.

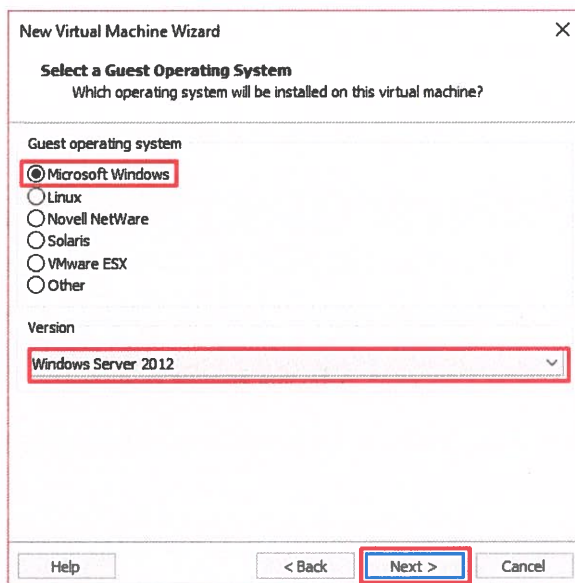


Imagem 9 – Escolha do sistema operativo e a versão a ser instalada
Fonte: do autor

Na secção Name The Virtual Machine atribuiu-se o nome e o caminho escolhido para máquina virtual. É neste passo onde atribui-se o nome da máquina virtual e a respetiva localização para armazenar o ficheiro de configuração da mesma.

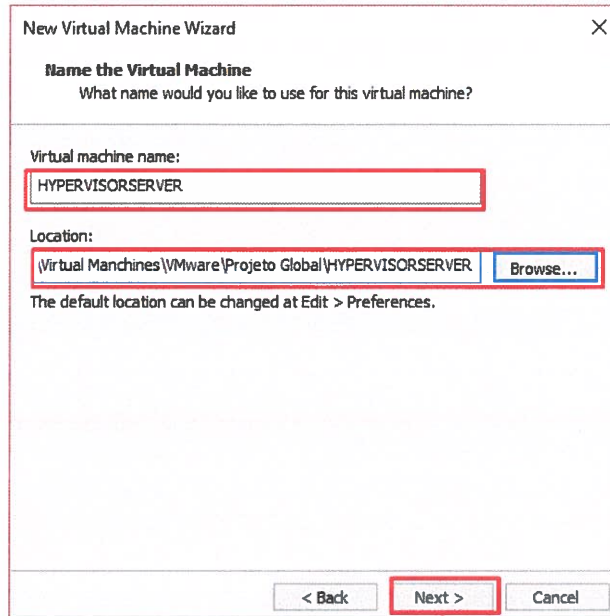


Imagem 10 – Atribuição do nome da máquina virtual e a sua localização
Fonte: do autor

Na secção Firmware Type é o passo para a escolha do *Firmware* para o *boot* da máquina virtual. Neste caso foi escolhida o *Firmware* do tipo BIOS.

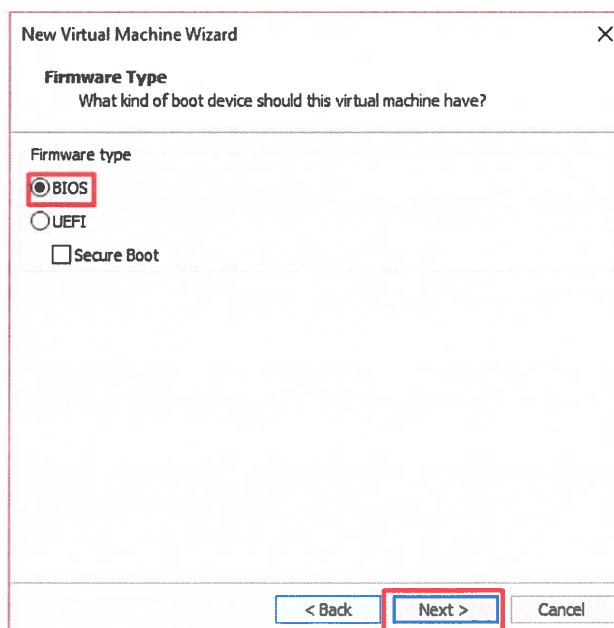


Imagem 11 – Escolha do *Firmware* para a máquina virtual
Fonte: do autor

Na secção Processor Configuration atribuiu dois como Number of Cores per Processor. Atribuição do número de núcleos para o processador virtual.

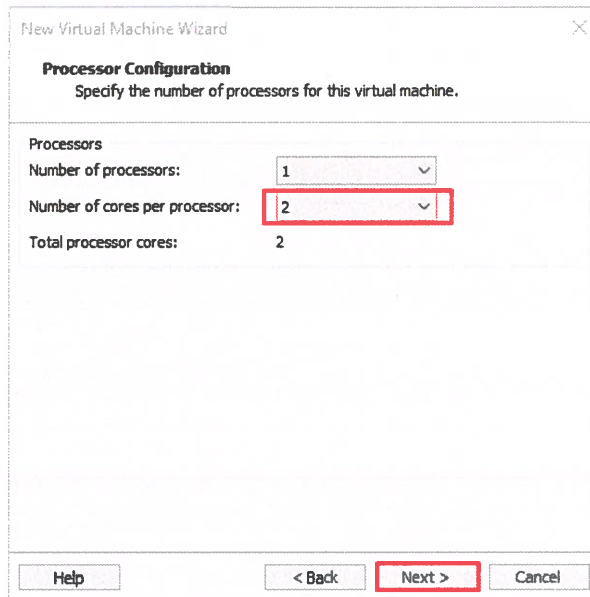


Imagem 12 – Configuração do processador
Fonte: do autor

Na secção Memory For Virtual Machine atribuiu-se 10240MB~10GB. É neste passo que se define a quantidade de memória RAM para a máquina virtual.

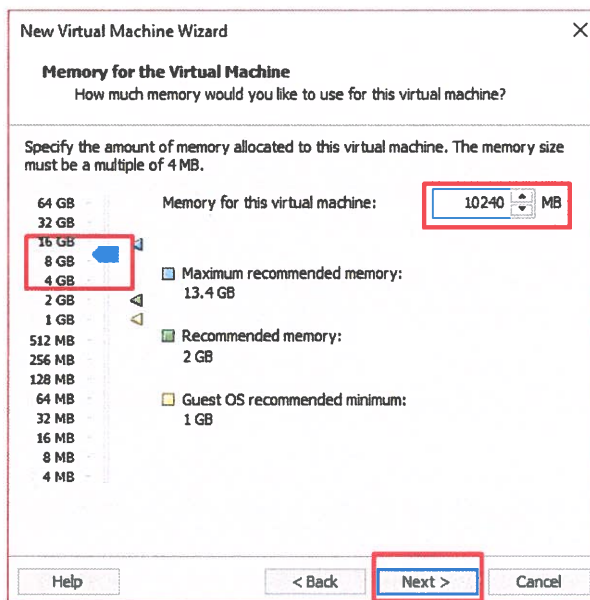


Imagem 13 – Atribuição da memória RAM
Fonte: do autor

Na secção Network Type escolheu-se a opção Use Network Address Translation (NAT). Neste passo é definido o tipo de placa de rede a ser usada pela máquina virtual, com o tipo de

placa de rede escolhido, a máquina virtual terá acesso à conexão de rede existente na placa de rede da máquina física.

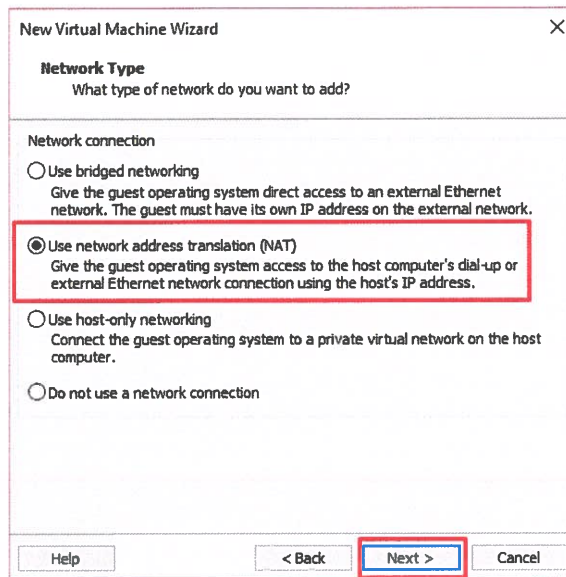


Imagem 14 – Escolha do tipo rede a ser conectada
Fonte: do autor

Na secção Select A Disk Type foi escolhida a opção SCSI (Recommended) para o tipo de disco.

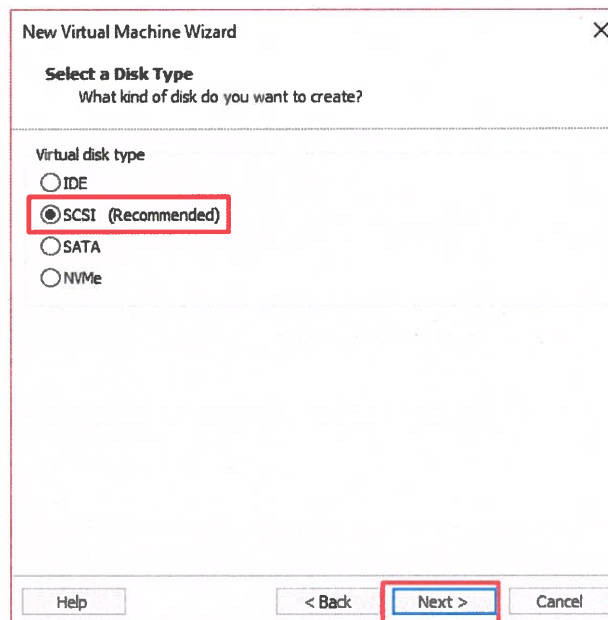


Imagem 15 – Escolha do tipo de disco para armazenamento
Fonte: do autor

Na secção Select a Disk escolheu-se a opção Create a new Virtual Disk para criar um novo disco virtual para a máquina. Neste passo realiza-se a escolha do disco para máquina

virtual, podendo ser um disco novo, um já existente ou usar diretamente o disco da máquina física.

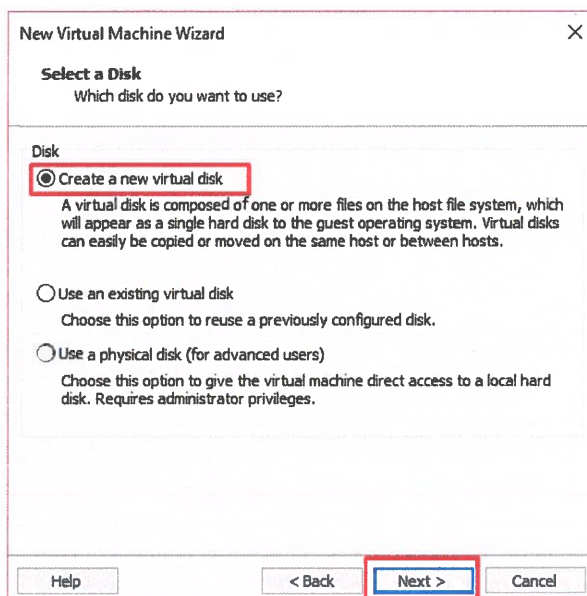


Imagem 16 – Seleção do disco de armazenamento
Fonte: do autor

Na secção Specify Disk Capacity definiu-se 150GB e escolheu-se a opção Store Virtual Disk as a Single File. É aqui onde definiu-se a capacidade de armazenamento do disco virtual e se se pretende criar um disco único ou ter o dividido em múltiplos ficheiros.

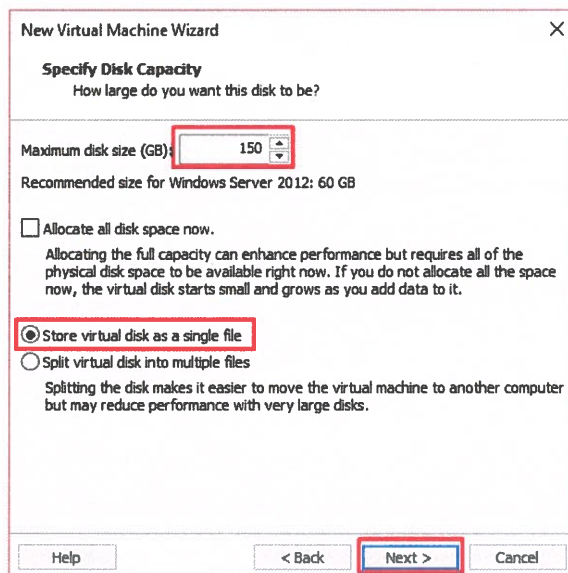


Imagem 17 – Atribuição da capacidade do disco de armazenamento
Fonte: do autor

Na secção Specify Disk File definiu-se o caminho onde será a guardado o disco virtual. Neste caso foi guardado na mesma localização onde estarão os restantes ficheiros da máquina virtual.

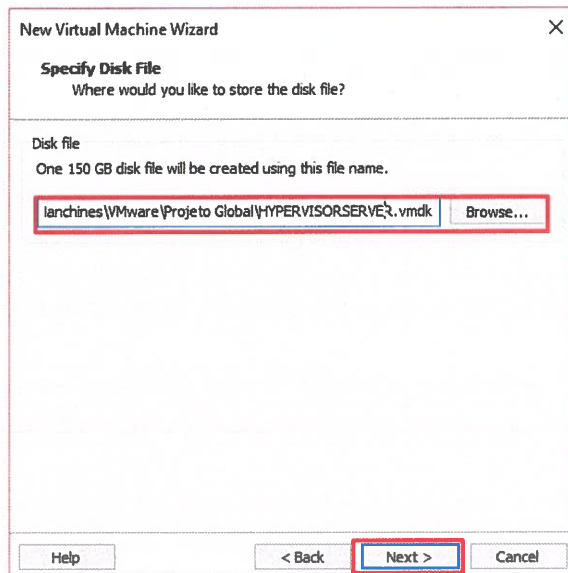


Imagem 18 – Especificação do ficheiro do disco e a sua localização
Fonte: do autor

Na secção Ready To Create Virtual Machine é possível verificar as configurações feitas para a máquina, é possível retificar caso haja a necessidade. Verificando-se que está tudo definido como pretendido pode agora clicar em Finish para criar a máquina virtual.

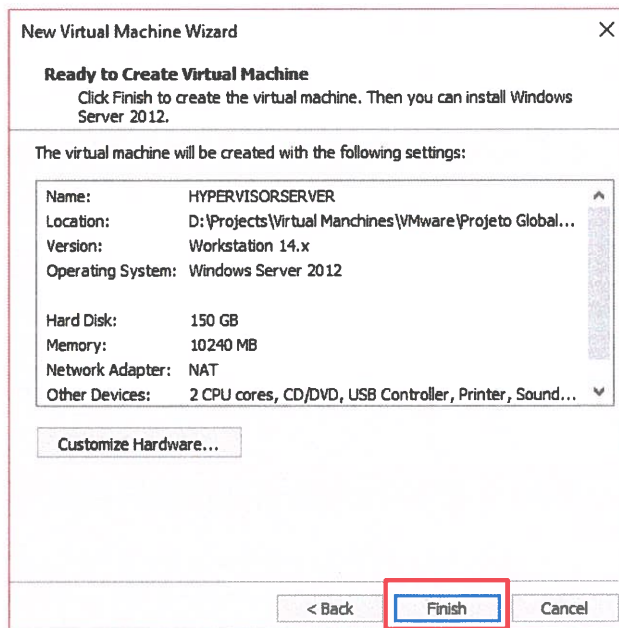


Imagem 19 – Resumo das configurações feitas para a máquina virtual
Fonte: do autor

Após clicar em Finish é possível ver a máquina virtual criada com as configurações pretendidas. A partir deste ponto, já é possível instalar um sistema operativo na máquina virtual.

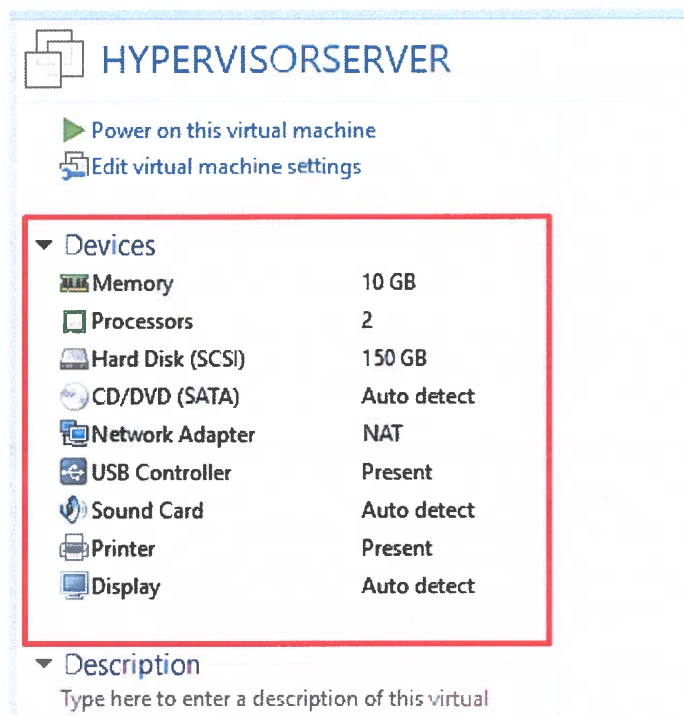


Imagem 20 – Máquina criada conforme as configurações
Fonte: do autor

Instalação do Sistema Operativo Windows Server 2012 R2 Standard

No ambiente de trabalho da máquina virtual, na secção em Edit em Virtual Machine Settings, na caixa de diálogo que aparece, clique em CD/DVD (SATA) e escolhe-se a opção Use ISO image File, para carregar o ISO do sistema operativo Windows Server 2012 R2 Standard e clica-se em Ok.

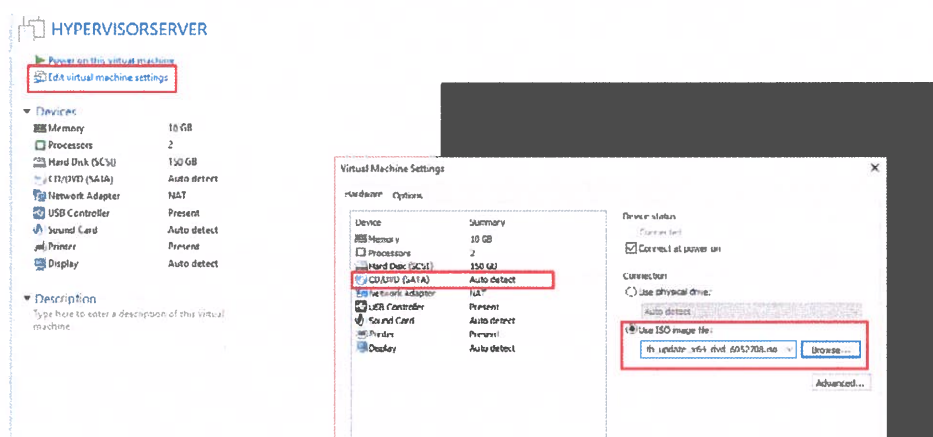


Imagem 21 – Escolha da fonte para instalação do sistema operativo
Fonte: do autor

Processo de instalação do sistema operativo seguindo os passos recomendados para instalação e configuração de uma nova instância e concluiu-se com a instalação do VMware Tools.

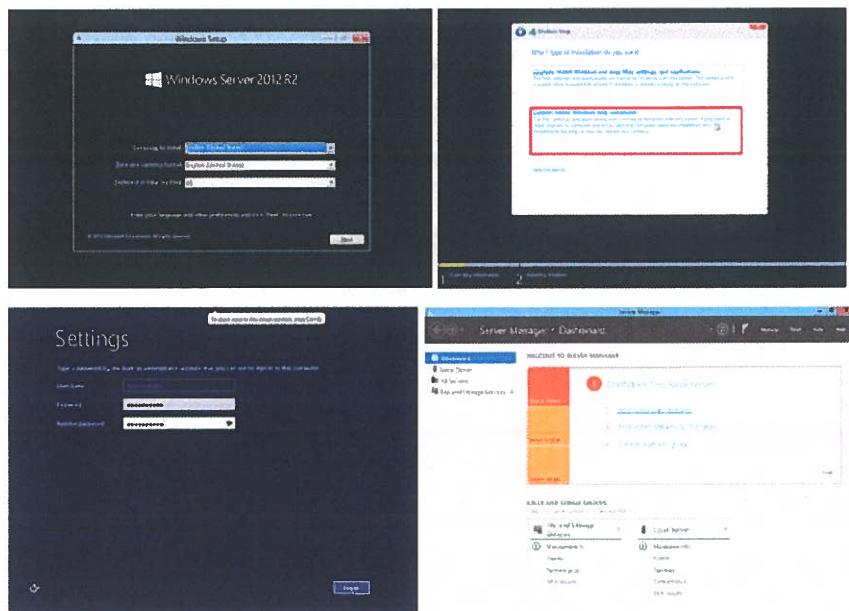


Imagem 22 – Instalação do sistema operativo
Fonte: do autor

Configuração da Máquina Virtual para permitir Virtualização sobre ela

Com a máquina desligada, nas configurações da máquina sobre o Processors na secção Virtualization Engine ativa-se a opção Virtualize Intel VT-x/EP Tor AMD-V/RVI e Virtualize CPU Performance Counters e clica-se em Ok.

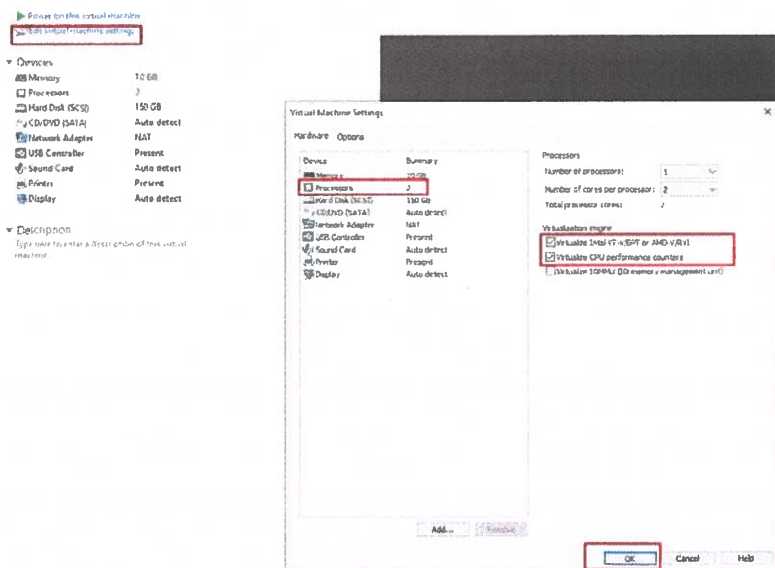
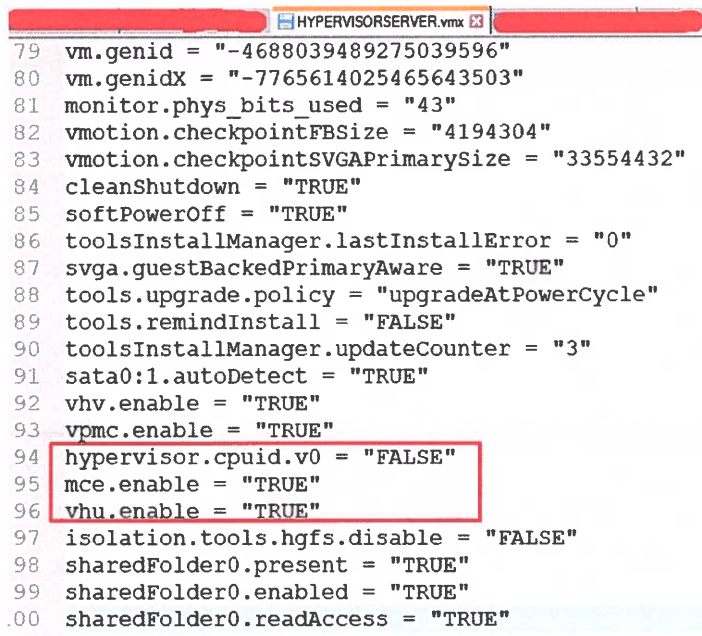


Imagem 23 – Passo 1 de configuração para a máquina virtual permitir Nesting virtualization
Fonte: do autor

No ficheiro *Hypervisorserver.vmx* constante no local onde são armazenados os ficheiros de configuração da máquina (ver pág. 16, imagem 10.), adicionar as seguintes linhas conforme a imagem abaixo e guarda o ficheiro com as linhas adicionadas.



```
79 vm.genid = "-4688039489275039596"
80 vm.genidX = "-7765614025465643503"
81 monitor.phys_bits_used = "43"
82 vmotion.checkpointFBSize = "4194304"
83 vmotion.checkpointSVGAPrimarySize = "33554432"
84 cleanShutdown = "TRUE"
85 softPowerOff = "TRUE"
86 toolsInstallManager.lastInstallError = "0"
87 svga.guestBackedPrimaryAware = "TRUE"
88 tools.upgrade.policy = "upgradeAtPowerCycle"
89 tools.remindInstall = "FALSE"
90 toolsInstallManager.updateCounter = "3"
91 sata0:1.autoDetect = "TRUE"
92 vhw.enable = "TRUE"
93 vpmc.enable = "TRUE"
94 hypervisor.cpuid.v0 = "FALSE"
95 mce.enable = "TRUE"
96 vhu.enable = "TRUE"
97 isolation.tools.hgfs.disable = "FALSE"
98 sharedFolder0.present = "TRUE"
99 sharedFolder0.enabled = "TRUE"
00 sharedFolder0.readAccess = "TRUE"
```

Imagem 24 – Passo 2 de configuração para a máquina virtual permitir Nesting virtualization
Fonte: do autor

Instalação o Hyper-V Manager via powershell com a seguinte linha de comando:

Install-WindowsFeature -Name Hyper-V -IncludeManagementTools -Restart

Hyper-V Manager é o software de virtualização da Microsoft. Também pode ser instalado pelo *wizard* através do Server Manager.

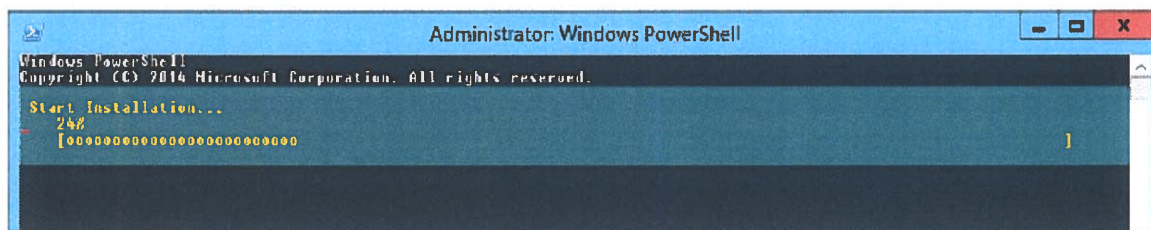


Imagem 25 – Instalação do Hyper-V Manager via PowerShell
Fonte: do autor

Após a instalação do Hyper-V Manager, criou-se um Private Virtual Switch no Hyper-V Manager (*switch* privado) para ser usado pelas máquinas virtuais DCSENER, SQLSERVER e CLIENT.

Com um clique no submenu do Tools no Server Manager, em Hyper-V Manager aparece o gestor de Hyper-v.

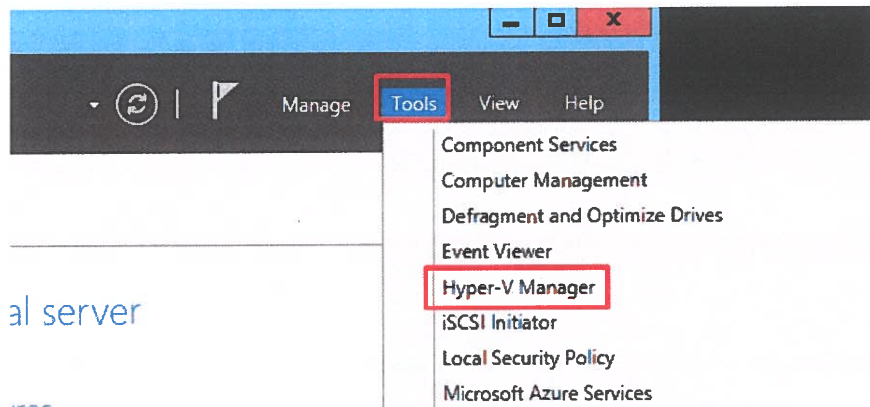


Imagem 26 – Hyper-V Manager após a respetiva instalação
Fonte: do autor

É nesta janela onde se pode criar e configurar máquinas virtuais e todos os seus hardwares.

Com um clique em Virtual Switch Manager constante em Actions na secção direita do Hyper-V Manager.

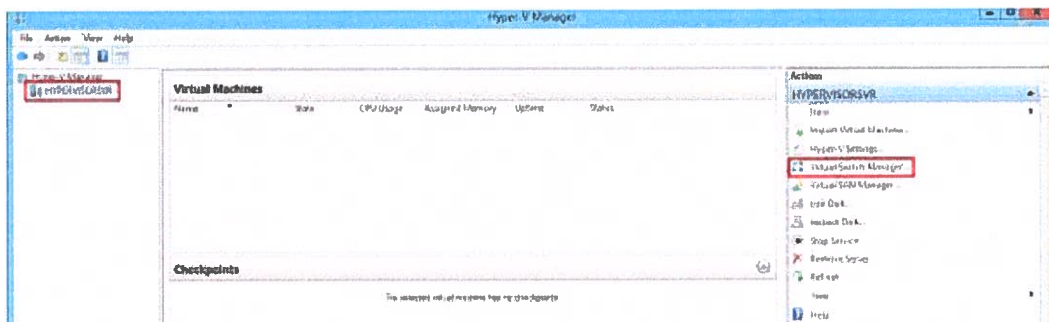


Imagem 27 – Criação de um switch virtual
Fonte: do autor

Surge a janela do Virtual Switch Manager onde faz-se as operações com as placas de redes virtuais. Na secção do Virtual Switches (a esquerda) clica-se em New Virtual Network Switch e no lado direito escolher o tipo de virtual switch Private e clicar em Create Virtual switch para proceder ao passo seguinte.

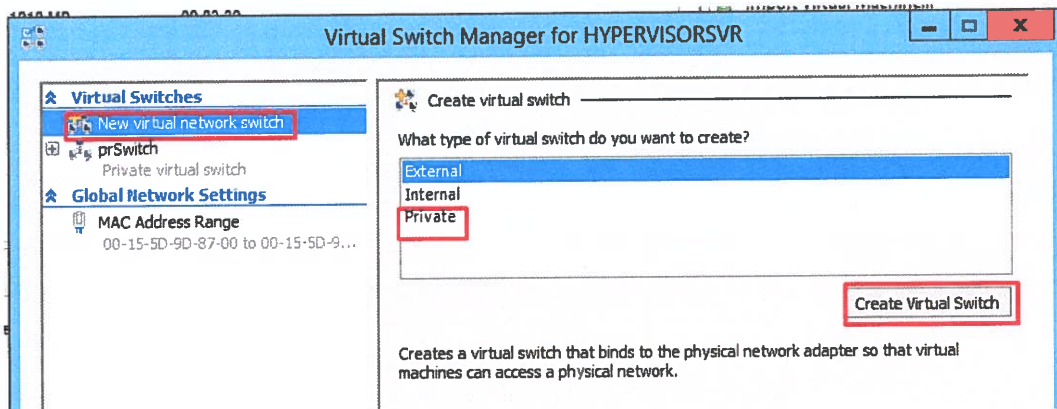


Imagem 28 – Escolha do tipo de switch virtual a ser criado
 Fonte: do autor

Na mesma janela, aparece as propriedades do virtual switch a ser criado, onde atribui-se do nome da switch, neste caso é prSwitch e confirma-se o tipo Private Network, depois destes pequenos passos clica-se em Apply e Ok.

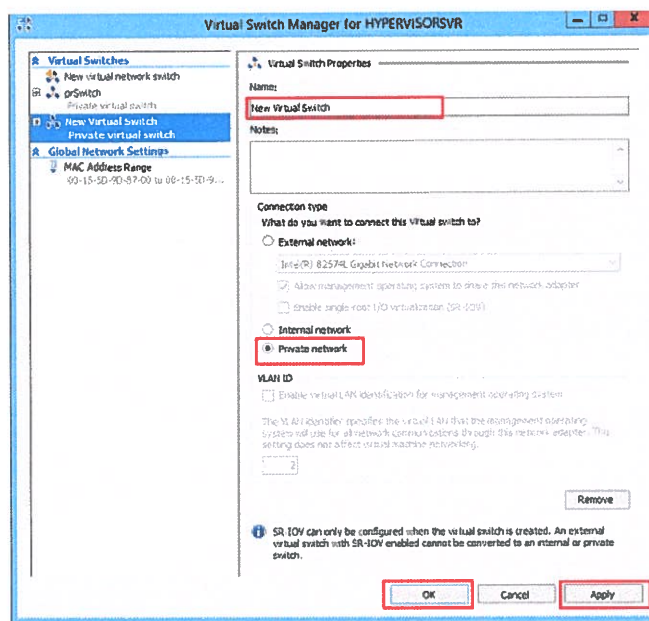


Imagem 29 – Escolha do tipo de conexão para o switch virtual
 Fonte: do autor

Olhando de volta para o lado esquerdo verifica-se o virtual switch criado com as propriedades especificadas. Este *virtual switch* vai permitir a comunicação entre as máquinas DCSEVER, SQLSERVER e CLIENT de maneira privada, tal que nem o PC FÍSICO e o HYPERVISORSERVER poderão comunicar-se com as mesmas.

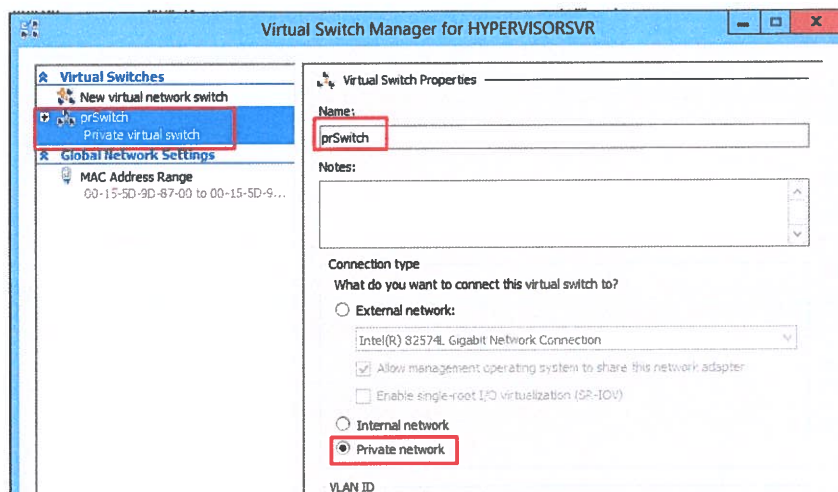


Imagem 30 – Switch virtual criado com a configuração pretendida
 Fonte: do autor

Depois de criado o virtual switch, é a hora para a criação das máquinas virtuais DC, SQLSERVER e CLIENT. Procedimentos usados para uma máquina foi igual para todas, adaptando-se as configurações para cada máquina conforme a tabela 2.

Com o clique no botão direito do rato sobre a instância do server Hyper-V, clica-se em New e no submenu em Virtual Machine.

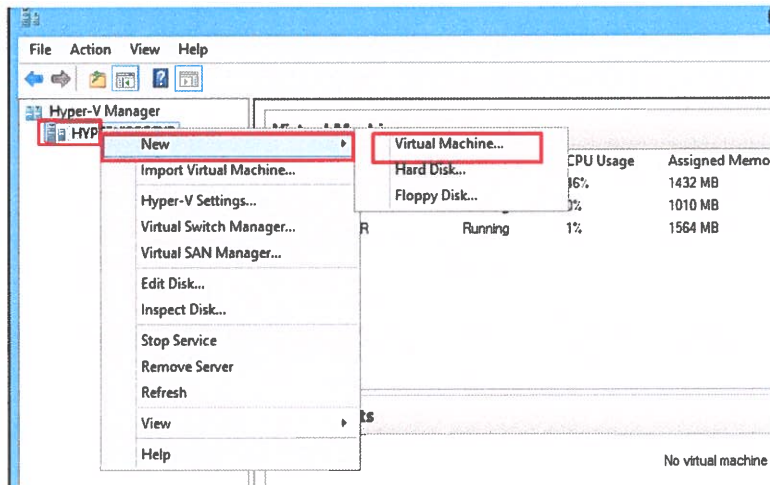


Imagem 31 – Criação de uma máquina virtual
 Fonte: do autor

É apresentada a janela New Virtual Machine Wizard onde faz-se as configurações de cada máquina no momento da sua criação. Saltando o passo do Before You Begin (pequena informação sobre o processo de criação de uma máquina virtual), na secção Specify Name and Location atribuiu-se um nome para a máquina virtual e a localização para armazenar os

ficheiros da máquina virtual (semelhante ao realizado na pág. 16 imagem 10). Neste caso deixou-se a localização predefinida.

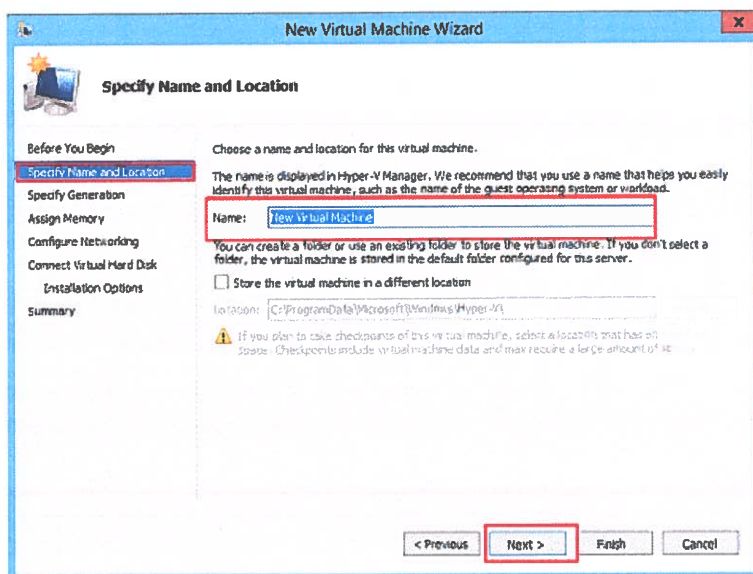


Imagem 32 – Atribuição do nome da máquina virtual e a respetiva localização
Fonte: do autor

O passo seguinte é a escolha da geração da máquina virtual. Geração 1 é quando a máquina virtual fornece o mesmo *hardware* virtual para a máquina virtual como nas versões anteriores do Hyper-V; Geração 2 a máquina virtual fornece suporte para as características como Secure Boot, iSCSI Boot, entre outras, o sistema operativo convidado deve ter pelo menos as versões do Windows Server 2012 ou Windows 8 de 64 bit. Na secção Specify Generation escolheu-se a opção Generation 1.

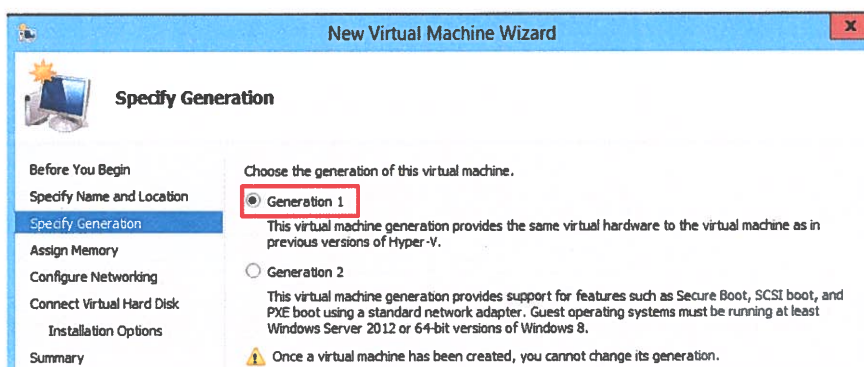


Imagem 33 – Escolha da geração para a máquina virtual
Fonte: do autor

Na secção Assign Memory atribui-se a memória RAM para a máquina, neste caso definiu-se 1024MB e o uso da memória dinâmica.

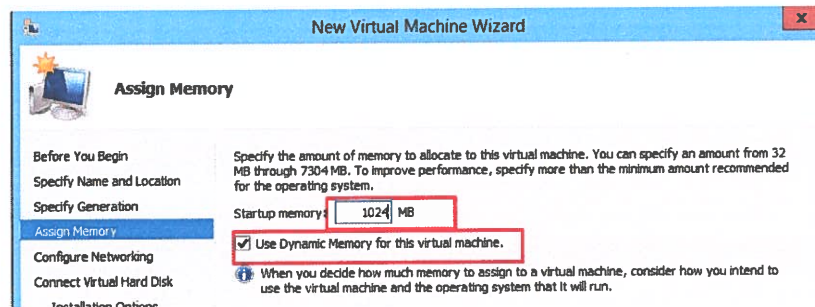


Imagem 34 – Atribuição da memória RAM
Fonte: do autor

Na secção Configure Network escolheu-se o virtual switch criado “prSwitch” (pág. 25 e 26).

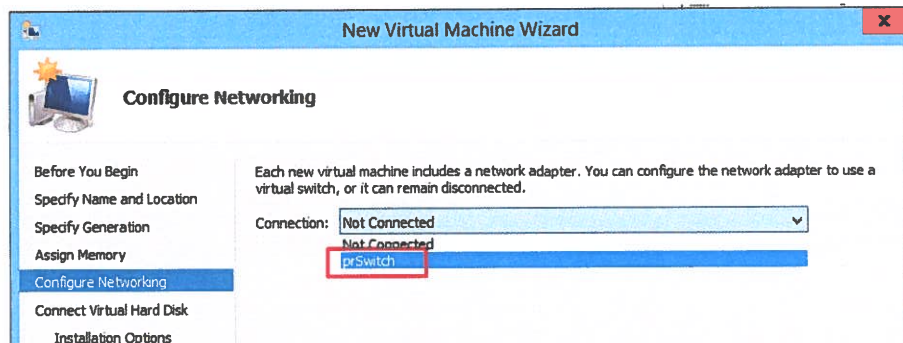


Imagem 35 – Escolha da conexão (switch virtual) para a placa de rede
Fonte: do autor

Na secção Connect Virtual Hard Disk escolheu-se a opção Create a Virtual Hard Disk na localização predefinida com o máximo de 127GB. Pode-se escolher outras opções como Use na existing virtual hard disk para o uso de um disco virtual VHD/VHDX já existente ou Attach a virtual hard disk later para adicionar um disco mais tarde.

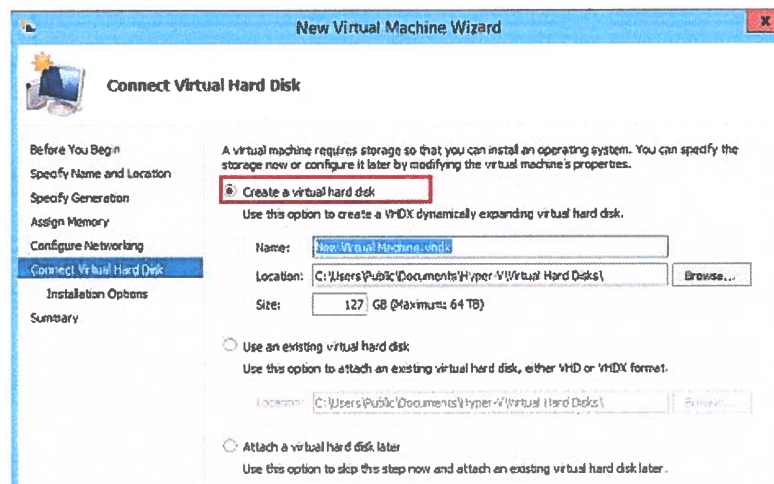


Imagem 36 – Especificação do armazenamento virtual
Fonte: do autor

Na subsecção Installation Options escolhe-se uma das quatro opções:

- Install na operating system later, opção que permite instalar o sistema operativo depois da máquina ser criada;
- Install na operating system from a bootable CD/DVD-ROM, instalação do sistema operativo através do boot com CD/DVD-ROM;
- Install na operating system from a bootable floppy disk, instalação do sistema operativo através do boot com um floppy disk;
- Install na operating system from a network-based Installation server, instalação do sistema operativo através da rede baseada no servidor de instalação;

Foi escolhida a opção Install Operating System Later.

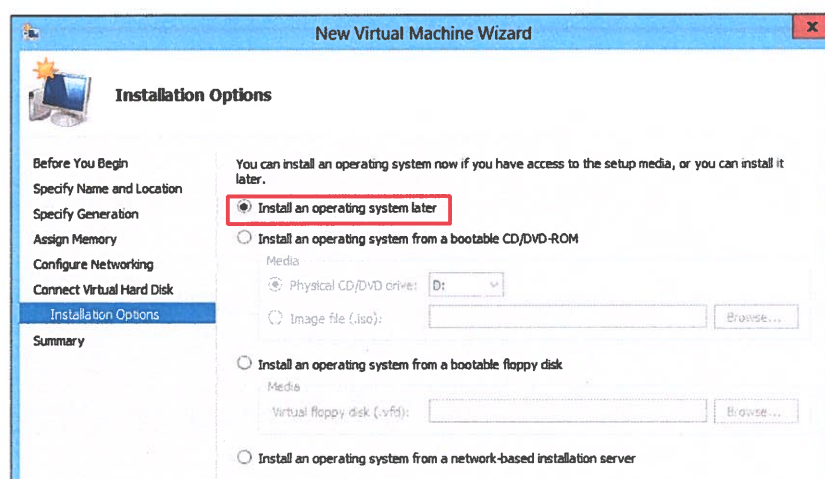


Imagem 37 – Escolha da fonte para a instalação do sistema operativo
Fonte: do autor

Após cumprido com os passos anteriores, na secção Completing the New Virtual Machine Wizard, é apresentado resumidamente as configurações da máquina conforme pretendido e clica-se em Finish para a criação da máquina virtual.

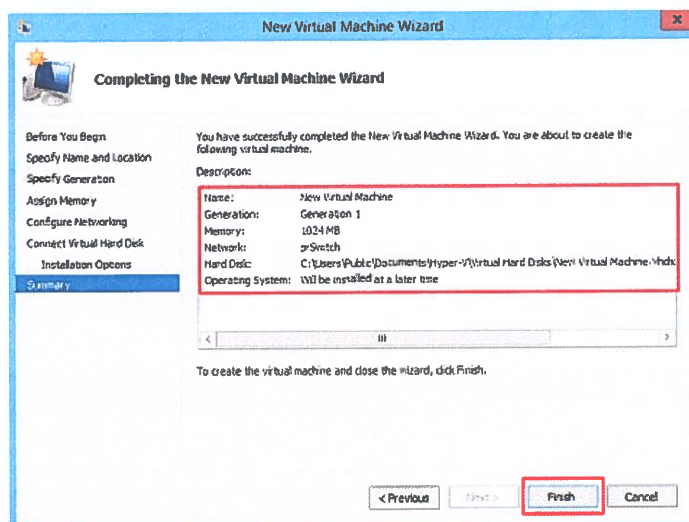


Imagem 38 – Resumo das configurações feitas para a máquina virtual
Fonte: do autor

De volta no Hyper-V Manager, na secção central encontram-se as máquinas criadas DC, SQLSERVER, CLIENT.

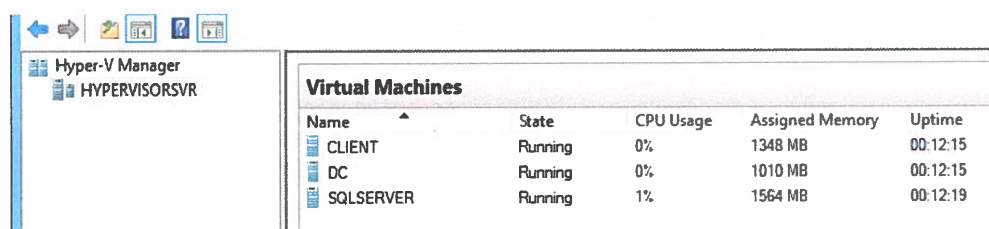


Imagem 39 – Verificação das três máquinas criadas
Fonte: do autor

Tabela 2 - Configurações das máquinas virtuais
Fonte: do autor

Máquina Virtual	Memória (Dinâmica)	Interface de Rede	Sistema Operativo	IP	DNS	Subnet Mask	Gateway
DC	1GB	prSwitch	Windows Server 2012 R2 Standard	192.168.1.1	127.0.0.1	255.255.255.0	N/A
SQLSERVER	4GB			192.168.1.2	192.168.1.1		
CLIENT	1GB		Windows 8.1 Pro	192.168.1.10			

Na tabela acima, verifica-se as principais configurações realizadas para as máquinas virtuais.

Promoção da Máquina Virtual DC a controlador de domínio

Este é o primeiro processo a ser realizado quando se pretende ter uma rede com domínio, é o processo onde se define qual máquina terá a principal função na rede como gerir contas de utilizadores, controlos de acessos e outras funcionalidades. Este processo pode ser

realizado por duas vias: através do assistente de instalação existente no Server Manager ou através da linha de comandos usando o PowerShell. Para este projeto foi usado o PowerShell para execução dos comandos abaixo:

Para instalação dos serviços associados à Active Directory e as ferramentas de gestão, foi usada a primeira linha de comandos, para criação de florestas, domínios e promoção do computador a controlador de domínio a segunda linha de comando.

Add-WindowsFeature AD-Domain-Services -IncludeManagementTools

Install-ADDSForest -DomainName “oproducer.local”

Após o reinício da máquina DC, passar-se-á doravante autenticar-se com a conta Administrador de Domínio.

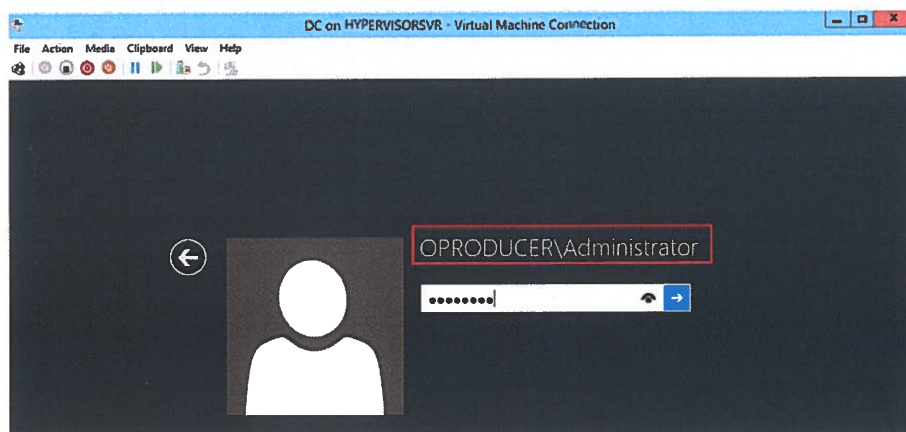


Imagem 40 – Autenticação à máquina virtual DC após a promoção para controlador de domínio
Fonte: do autor

No Server Manager na secção do Local Server constata-se que a máquina já pertence a um domínio.

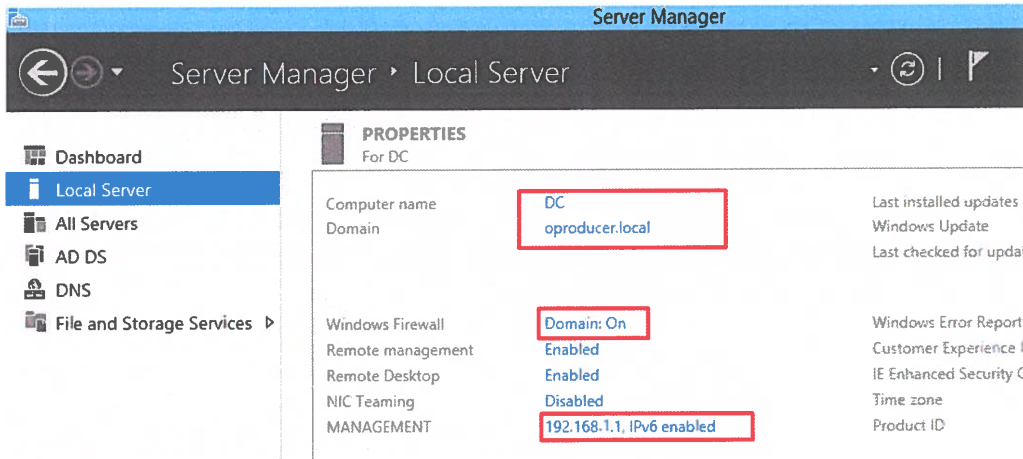


Imagem 41 – Propriedades da máquina virtual dentro do domínio criado
 Fonte: do autor

Considerando que as máquinas SQLSERVER e CLIENT já tenham o sistema operativo instalado e configurado (placas de rede e outros dispositivos necessários), foram adicionadas ao domínio recentemente criado “oproducer.local”.

Na Systems Properties na aba Computer Name com um clique no botão Change que abre uma outra caixa de diálogo onde na secção Member of escolheu-se a opção Domain e introduziu-se as credenciais para acesso ao domínio. Tendo concluído com sucesso esta operação, realizou-se o mesmo na máquina CLIENT.

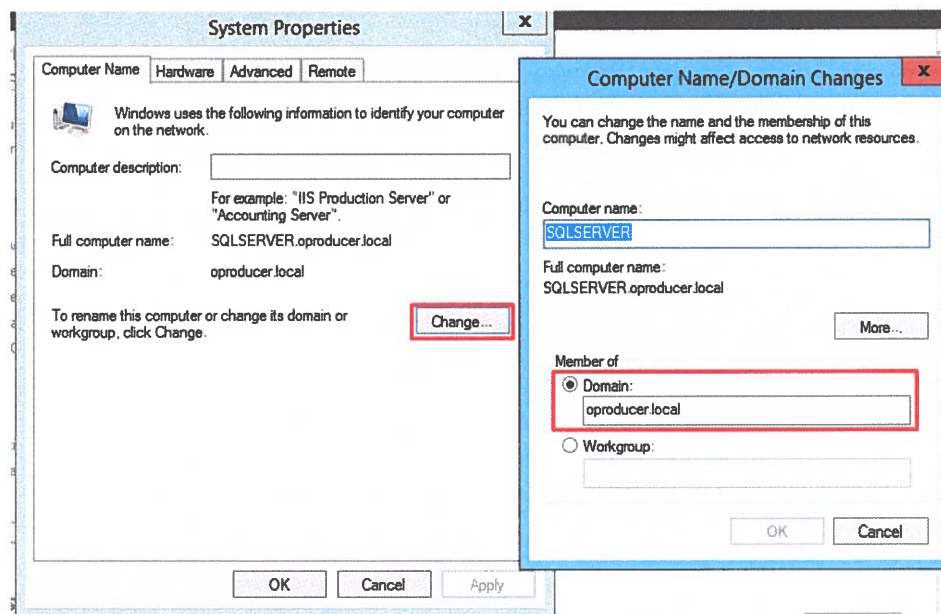


Imagem 42 – Adição das máquinas virtuais SQLSERVER e CLIENT no domínio
 Fonte: do autor

A partir deste momento as três máquinas virtuais já se encontram comunicáveis no mesmo domínio.

Máquina virtual SQLSERVER foi movida da unidade organizacional Computer para SERVER. Esta última foi criada manualmente para armazenar as máquinas do tipo servidor.

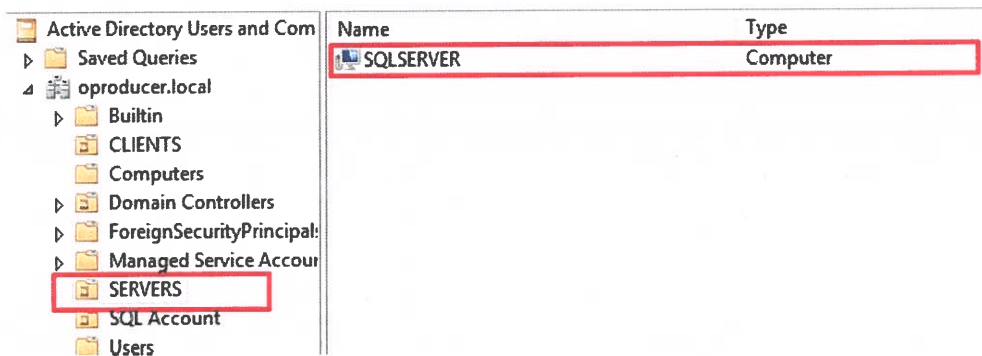


Imagem 43 – Verificação da máquina virtual SQLSERVER
Fonte: do autor

Máquina virtual CLIENT movida da unidade organizacional Computer para CLIENTS. Esta última também foi criada manualmente para armazenar as máquinas do tipo cliente.

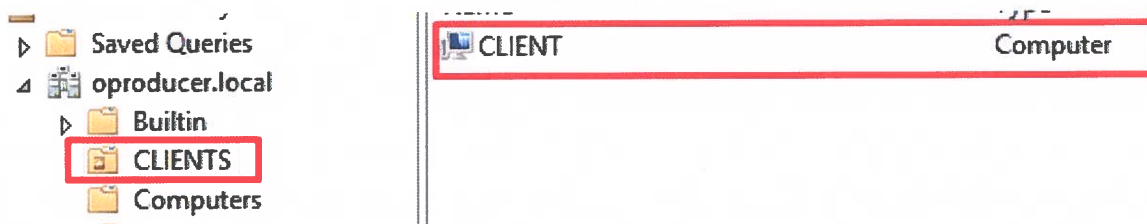


Imagem 44 – Verificação da máquina virtual CLIENT
Fonte: do autor

Criação de Conta de Utilizador no Domínio (SQL User)

Foi criada uma conta de utilizador “SQL User”, que será usada para acesso ao Servidor SQL (instância de base de dados) via máquina Client.

Esta conta foi criada na unidade organizacional “SQL Account”, que serve para alojar a(s) conta(s) de utilizador(es) relacionada(s) com os serviços do SQL Server.

Com um clique com o botão direito do rato dentro da unidade organizacional SQL Account, na janela que aparece para criação de um novo objeto (Utilizado, atribuiu-se o nome, o user logon name.

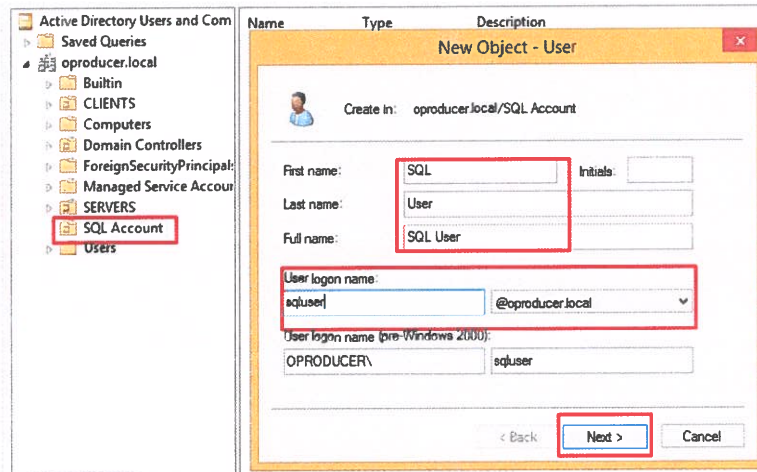


Imagem 45 – Criação de uma conta de utilizador no domínio
 Fonte: do autor

O próximo passo foi atribuir palavra-passe para esta conta de utilizador, definir algumas propriedades para a palavra-passe e verificado que tudo foi feito como pretendido, clicou-se em Finish para concretização do processo de criação de utilizador.

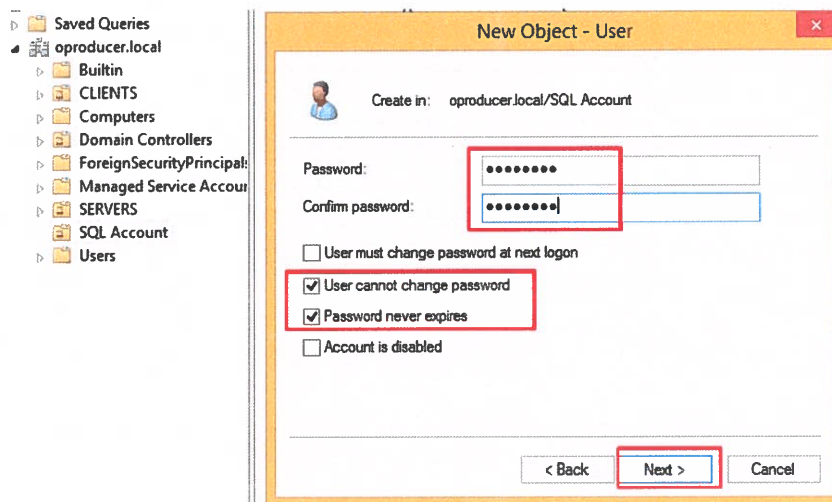


Imagem 46 – Atribuição da palavra-passe
 Fonte: do autor

Dentro da unidade organizacional SQL Account é possível ver a conta criada. Esta conta está simplesmente configurada para acesso a qualquer máquina do domínio, sem qualquer permissão sobre a administração do sistema..

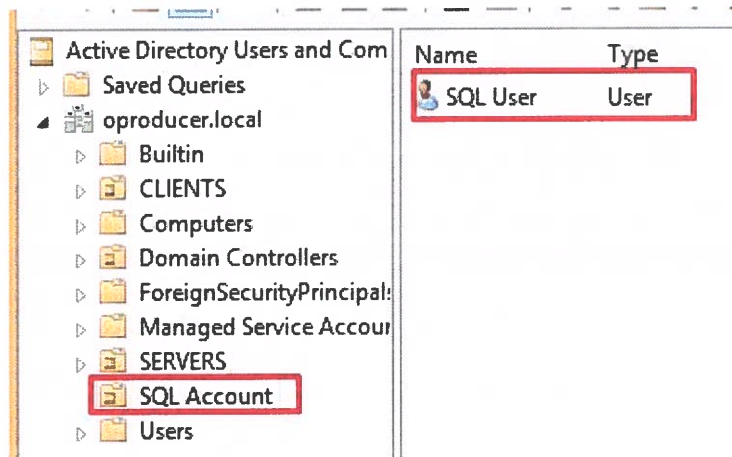


Imagem 47 – Conta de utilizador criada na unidade organizacional
 Fonte: do autor

Instalação do Microsoft SQL Server 2014 Standard Edition na máquina SQLSERVER

Foi instalado o Microsoft SQL Server 2014 Standard Edition na máquina SQLSERVER para poder alojar a base de dados para o uso numa aplicação ADO.NET.

Com o duplo clique sobre o setup.exe do Microsoft SQL Server 2014 Standard Edition, aparece a janela SQL Server Installation Center. Com um clique em Installation no lado esquerdo, escolhe-se a opção New SQL Server Stand-alone Installation or add features to an existing Installation, para começar o processo de instalação de uma instância nova.

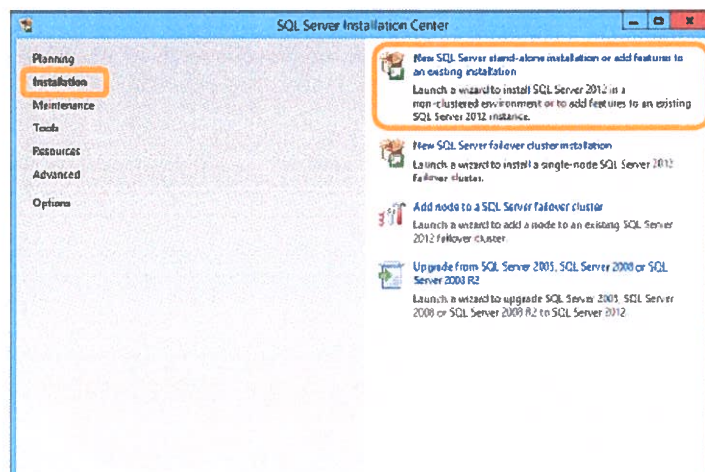


Imagem 48 – Escolha para instalação de uma instância nova do Microsoft SQL Server 2014
 Fonte: do autor

Na janela SQL Server 2014 Setup (a janela onde é feita toda a configuração para instalação dos serviços Microsoft SQL Server). Na imagem abaixo fez-se a aceitação dos termos de licença do software. Muitos passos não serão abordados, devido o automatismo existente nas suas execuções.

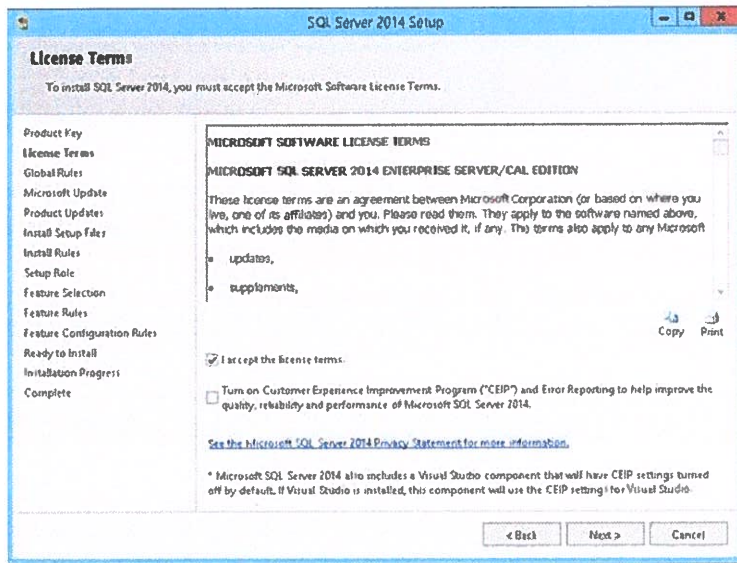


Imagem 49 – Aceitação dos termos de licença do Microsoft SQL Server 2014
Fonte: do autor

Verificação de regras importantes para um ótimo funcionamento do servidor de SQL Server.

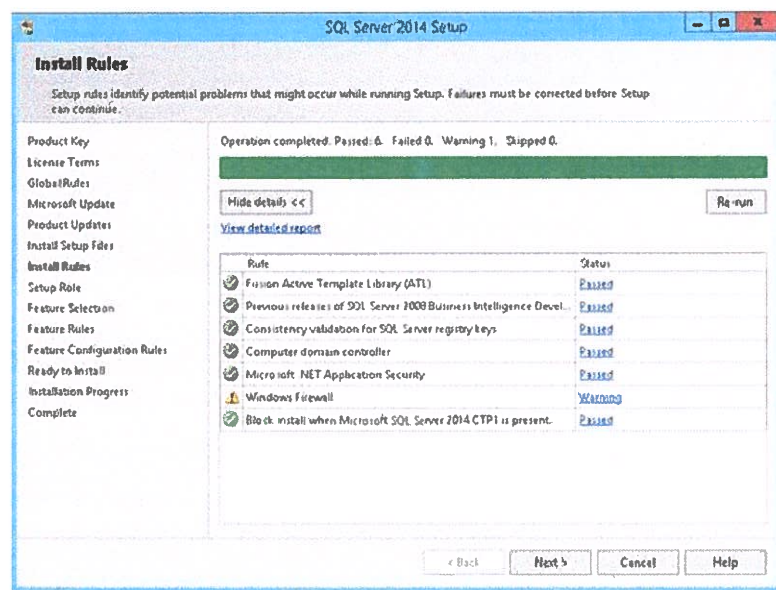


Imagem 50 – Configuração de algumas regras para instalação do Microsoft SQL Server
Fonte: do autor

Configuração da Instância (escolha do nome da instância)

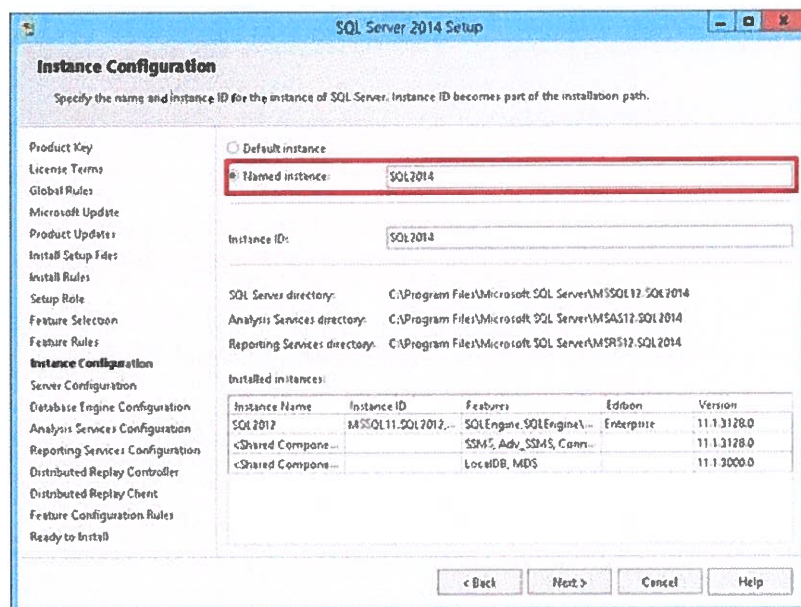


Imagem 51 – Configuração de uma instância do Microsoft SQL Server 2014 Standard
 Fonte: do autor

Configuração do Servidor, definição das contas que usarão os serviços de uso automático no Servidor. É neste passo onde define-se as contas de serviço para os serviços Microsoft SQL Server.

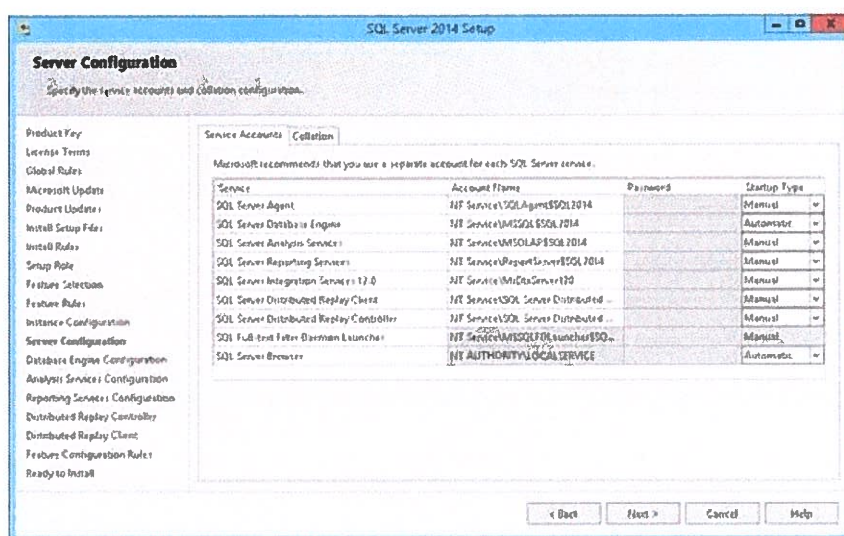


Imagem 52 – Configuração de contas de utilizadores aos serviços do Microsoft SQL Server 2014 Standard
 Fonte: do autor

Definição do modo de autenticação e a palavra-passe, as contas para o uso do Microsoft SQL Server, pode-se escolher o modo de autenticação pela conta do Windows que realizou o login ou autenticação com conta do Windows e autenticação do SQL Server, define-se a palavra-passe para a conta “sa” caso escolhe-se a segunda opção. Neste passo foram adicionadas as contas Domain Administrator e SQL User.

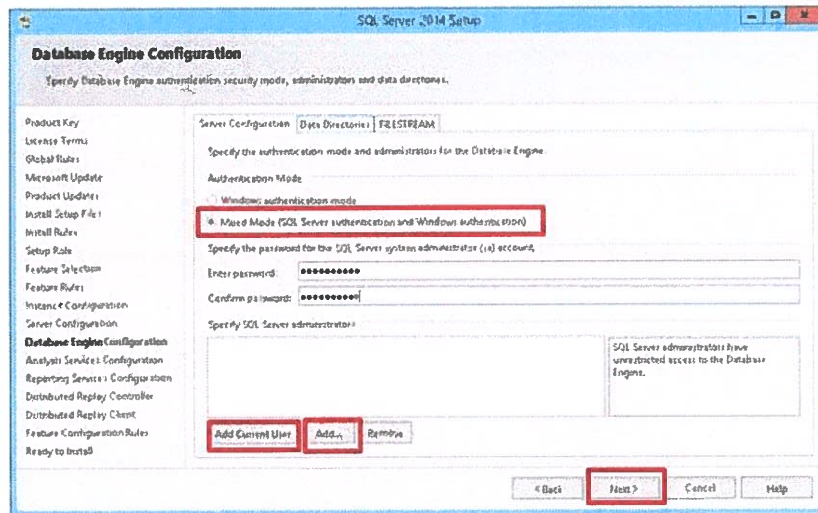


Imagem 53 – Configuração do modo de autenticação e adição de contas de serviço para a instância do Microsoft SQL Server
Fonte: do autor

Resumo de toda a configuração realizada, é aqui onde verifica-se se foi configurado tudo conforme pretendido, depois disto, é instalado o SQL Server.

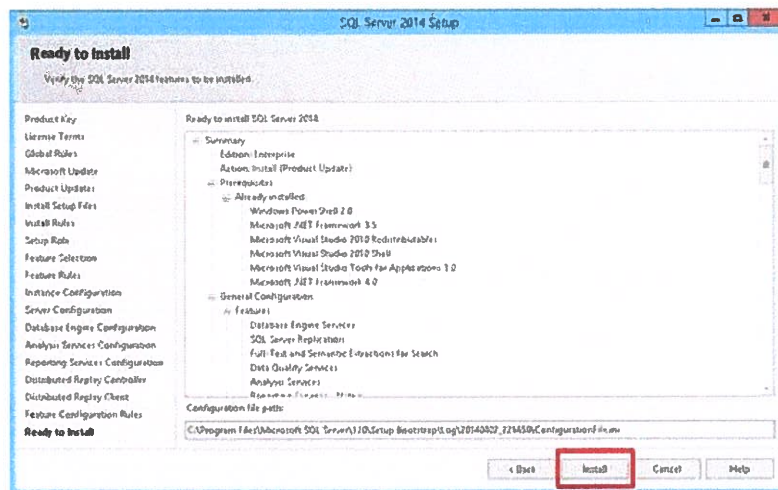


Imagem 54 – Resumo das configurações para a instalação do Microsoft SQL Server 2014 Standard Edition
Fonte: do autor

Após a correta instalação, é apresentada da janela abaixo mostrando tudo que foi instalado com o respetivo estado (sucesso ou insucesso).

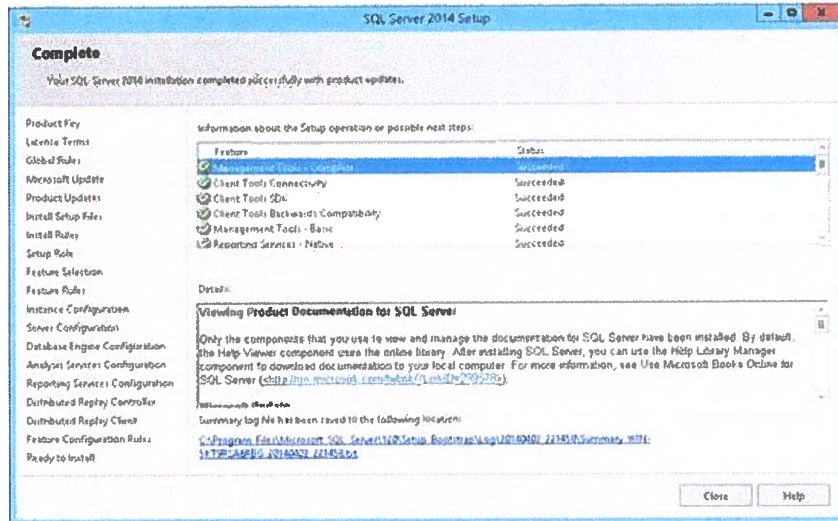


Imagem 55 – Fim da instalação do Microsoft SQL Server 2014 Standard Edition
 Fonte: do autor

Ambiente do Microsoft SQL Server 2014 para o início da sessão, neste caso com o Windows Authentication (autenticação através da conta do Windows que fez o login).

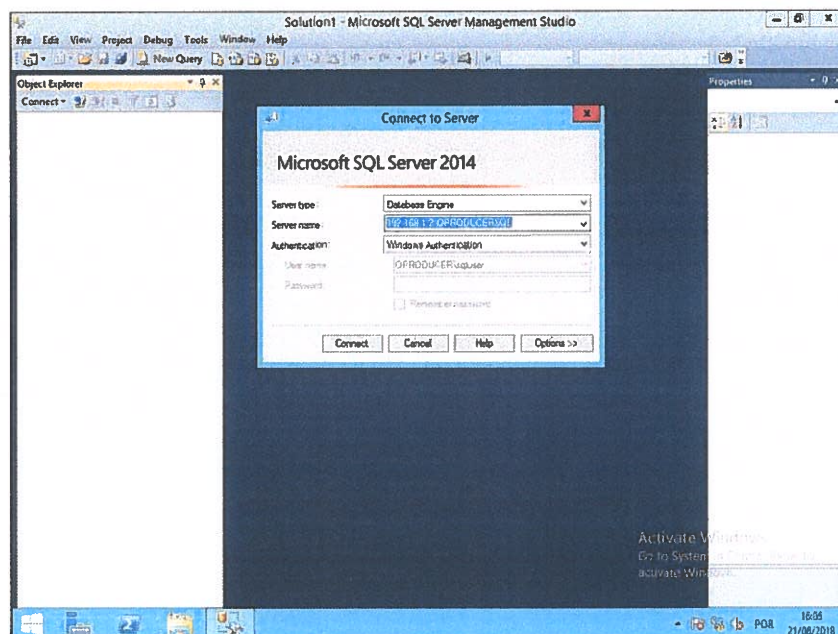


Imagem 56 – Autenticação ao servidor SQL Server com uma conta configurada para o efeito
 Fonte: do autor

Verificação das propriedades da conexão da conta SQL User

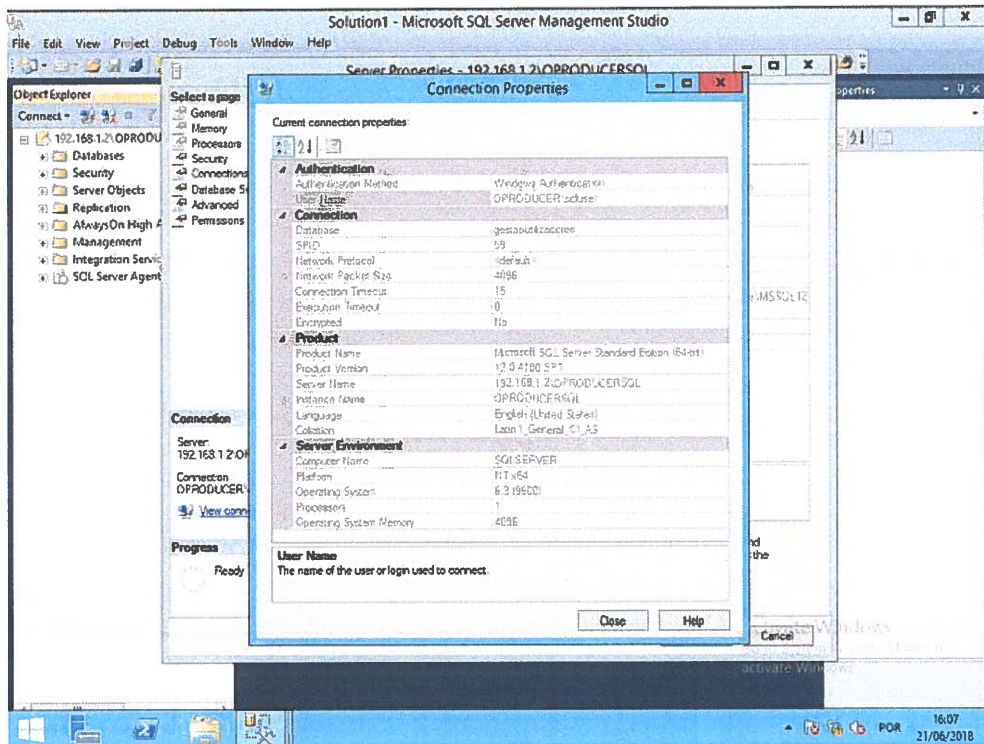


Imagem 57 – Propriedades de conexão da conta autenticada e da instância do Microsoft SQL Server
 Fonte: do autor

Atribuição de funções para a conta SQL User, de maneira que possa gerir a base de dados. É neste passo onde definem-se as limitações para as contas de utilizadores à base de dados.

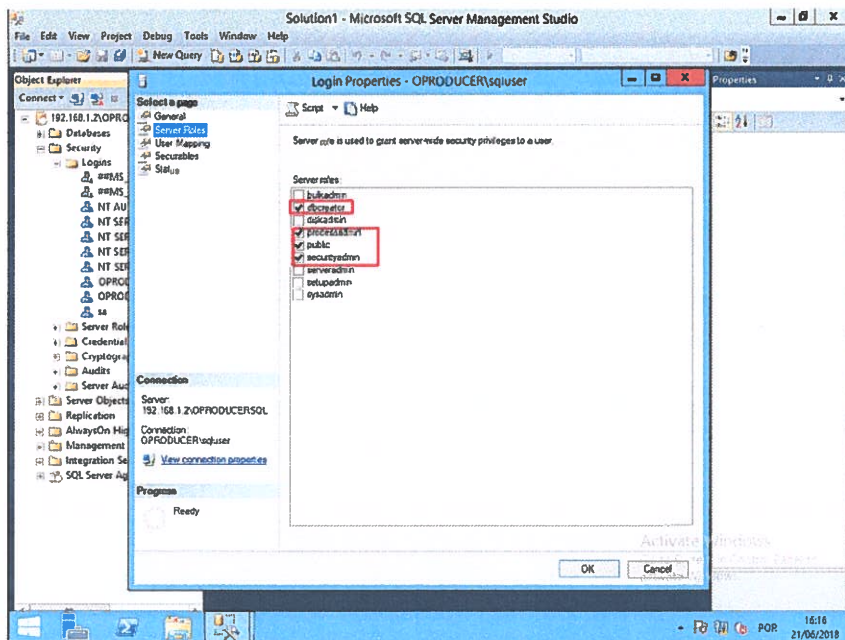


Imagem 58 – Atribuição de funções à base de dados para a conta de serviço
 Fonte: do autor

Criação da base de dados

Com a sessão já iniciada, na secção do Object Explorer, com um clique no botão direito do rato sobre Database, e no submenu que aparece clica-se em New Database...

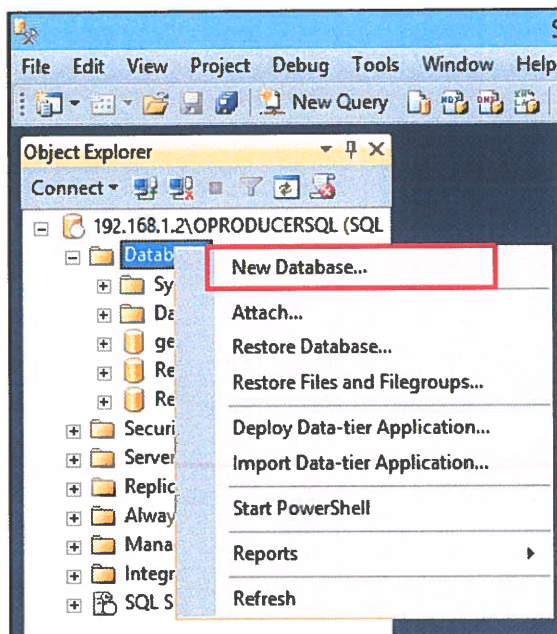


Imagem 59 – Criação de uma nova base de dados
Fonte: do autor

Atribuir o nome da base de dados e clicar em Ok. A base de dados e os restantes ficheiros serão armazenados numa localização predefinida.

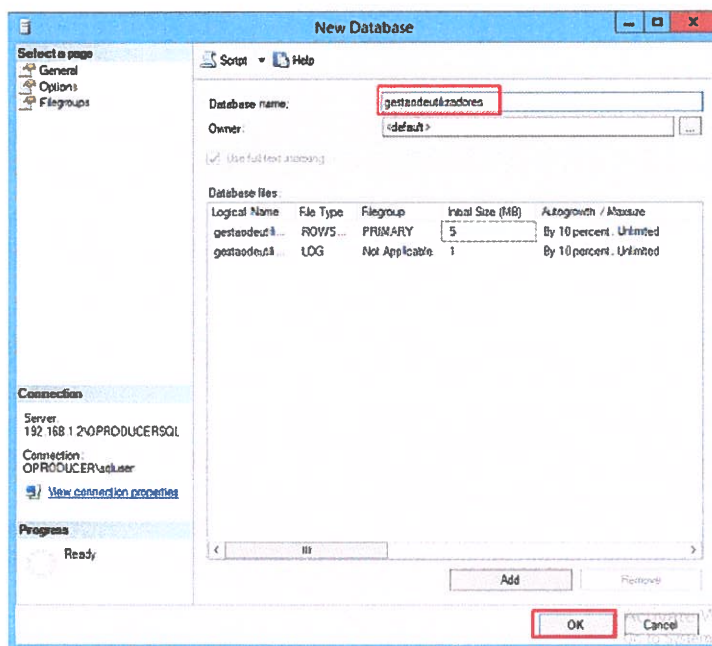


Imagem 60 – Definição do nome da base de dados
Fonte: do autor

Esta base de dados é composta por uma tabela e dois stored Procedures

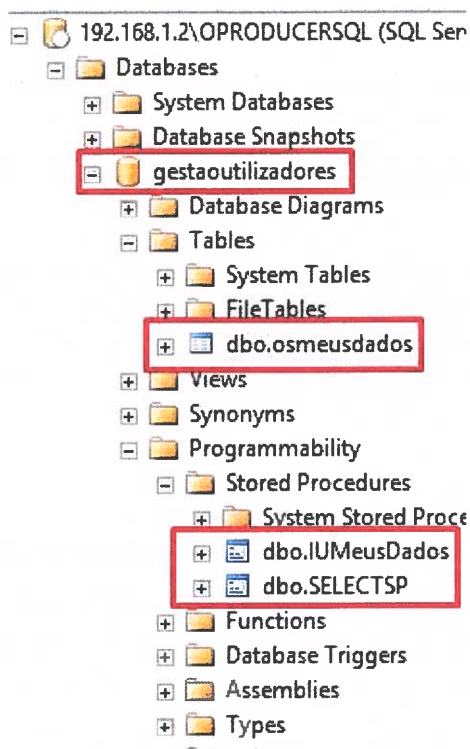


Imagem 61 – Base de dados, tabela e stored Procedures criadas
Fonte: do autor

Atributos da tabela dbo.osmeusdados

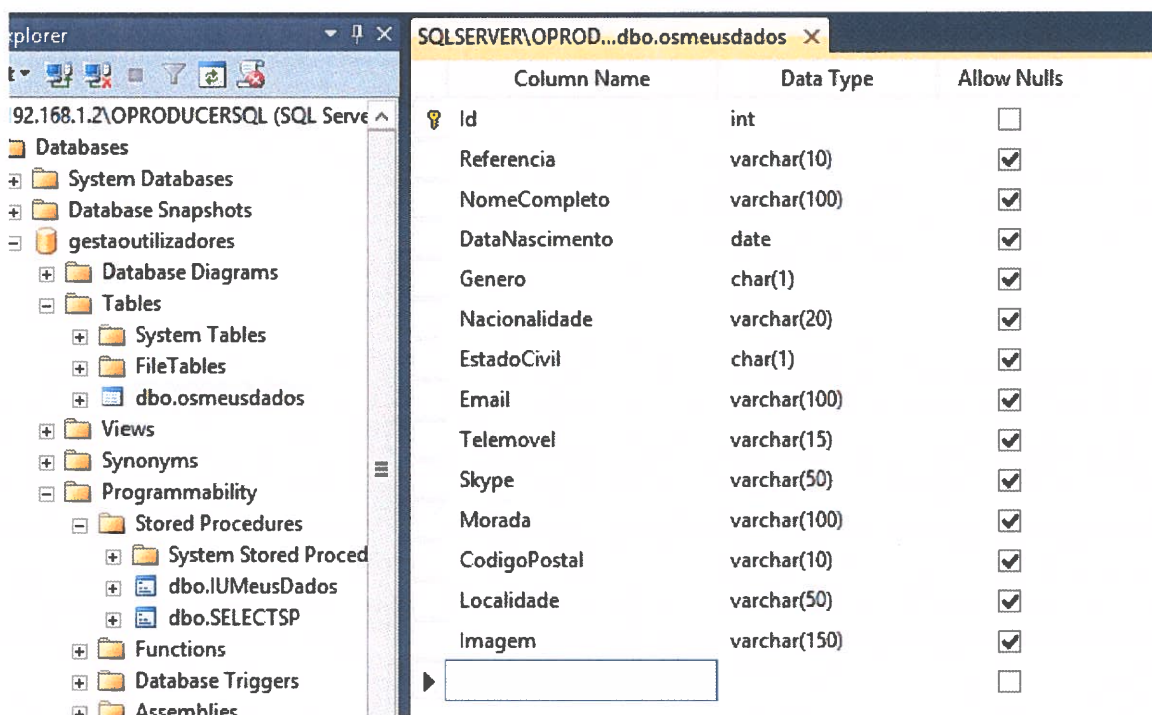


Imagem 62 – Atributos (campos) da tabela dbo.osmeusdados
Fonte: do autor

Stored Procedures para inserir e atualizar os dados na tabela. Esta stored procedure terá a dupla função de inserir e atualizar todos os dados na base de dados. Através de uma variável que recebe um valor através da aplicação, automaticamente é escolhida a instrução sql para inserir ou para atualizar.

```

SQLQuery3.sql - 19...DUCER\sqluser (60) X SQLSERVER\OPROD...dbo.osmeusdados
USE [gestaoutilizadores]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[InsMeusDados]
    @IU CHAR(1),
    @Id VARCHAR(15),
    @NomeCompleto VARCHAR(100),
    @DataNascimento DATE,
    @Genero CHAR(1),
    @Nacionalidade VARCHAR(20),
    @EstadoCivil CHAR(1),
    @Email VARCHAR(100),
    @Telefone1 VARCHAR(15),
    @Skype VARCHAR(20),
    @Morada VARCHAR(50),
    @CodigoPostal VARCHAR(10),
    @Localidade VARCHAR(30),
    @Imagem VARCHAR(120)
AS
BEGIN
    SET NOCOUNT ON;
    IF @IU='I'
    BEGIN
        INSERT INTO [dbo].[osmeusdados] ([Referencia], [NomeCompleto], [DataNascimento], [Genero], [Nacionalidade],
        [EstadoCivil], [Email], [Telefone1], [Skype], [Morada], [CodigoPostal], [Localidade], [Imagem])
        VALUES (LEFT(REPLICATE(' ', 10), 10), @NomeCompleto, @DataNascimento, @Genero, @Nacionalidade, @EstadoCivil, @Email,
        @Telefone1, @Skype, @Morada, @CodigoPostal, @Localidade, @Imagem)
        SELECT Referencia FROM [osmeusdados] WHERE Id = @@Identity
    END
    IF @IU='U'
    BEGIN
        UPDATE [dbo].[osmeusdados] SET [NomeCompleto] = @NomeCompleto, [DataNascimento] = @DataNascimento, [Genero] = @Genero,
        [Nacionalidade] = @Nacionalidade, [EstadoCivil] = @EstadoCivil, [Email] = @Email, [Telefone1] = @Telefone1, [Skype] = @Skype,
        [Morada] = @Morada, [CodigoPostal] = @CodigoPostal, [Localidade] = @Localidade, [Imagem] = @Imagem WHERE Referencia = @Id
    END
END

```

Imagem 63 – Stored Procedure com os parâmetros e instruções para inserir e atualizar os dados na tabela
Fonte: do autor

Stored Procedure para ver dados na tabela. Esta stored Procedure permite visualizar um registo na base de dados, apresentando o registo que tenha um Id igual ao valor que está armazenado na variável @Id. Esta instrução retorna um registo.

```

SQLQuery9.sql - 19...DUCER\sqluser (57) X
USE [gestaoutilizadores]
GO
/***** Object: StoredProcedure [dbo].[SELECTSP] Script Date:
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author: <Author,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====
ALTER PROCEDURE [dbo].[SELECTSP]
    -- Add the parameters for the stored procedure here
    @Id VARCHAR(15)
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    SELECT * FROM [osmeusdados] WHERE Referencia=@Id
END

```

Imagem 64 – Stored Procedure com os parâmetros e instrução para consultar dados na tabela
Fonte: do autor

Verificação dos serviços SQL Server. Através da janela de Services/Serviços do Microsoft Windows é possível verificar os serviços SQL Server.

SQL Full-text Filter Daemon Launcher (OPRODUCERSQL)	Service to la...	Running	Automatic	NT Service...
SQL Server (OPRODUCERSQL)	Provides sto...	Running	Automatic	OPRODUC...
SQL Server Agent (OPRODUCERSQL)	Executes jo...	Running	Automatic	OPRODUC...
SQL Server Analysis Services (OPRODUCERSQL)	Supplies onl...	Running	Automatic	OPRODUC...
SQL Server Browser	Provides SQ...	Running	Automatic	Local Service
SQL Server Reporting Services (OPRODUCERSQL)	Manages, e...	Running	Automatic	OPRODUC...
SQL Server VSS Writer	Provides th...	Running	Automatic	Local Syste...

Imagem 65 – Visualização dos serviços SQL Server em execução
Fonte: do autor

Criação e desenvolvimento da aplicação ADO.NET

A aplicação foi desenvolvida usando o Visual Studio 2015 Enterprise.

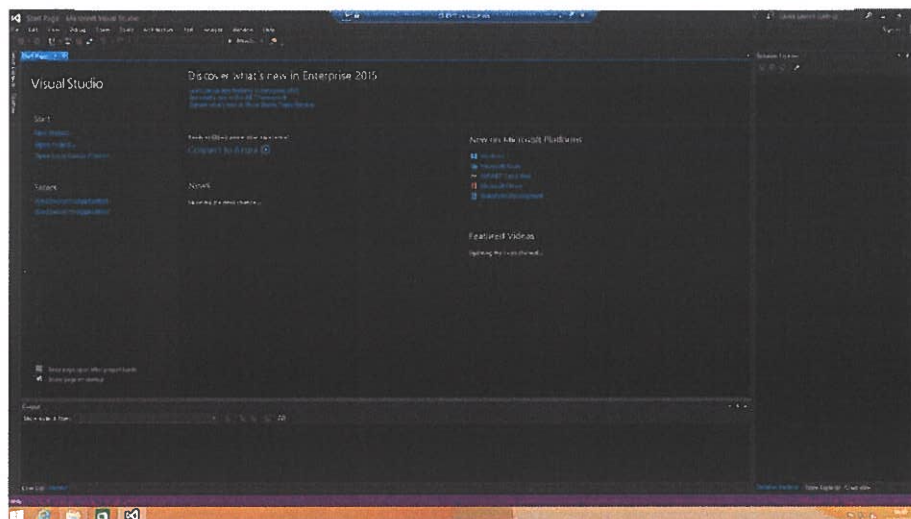


Imagem 66 – Janela do Visual Studio 2015 Enterprise
Fonte: do autor

Esta aplicação consiste numa pequena plataforma para criação de um perfil para criar um curriculum vitae, semelhante ao conhecido Curriculum Europass.

A aplicação é composta por dois formulários: Splashscreen e Os Meus Dados

Splashscreen é o formulário de entrada, que permite fazer a abertura da aplicação e tem também a função de verificar o estado da conexão com o servidor SQLSERVER, nomeadamente com a instância de base de dados.

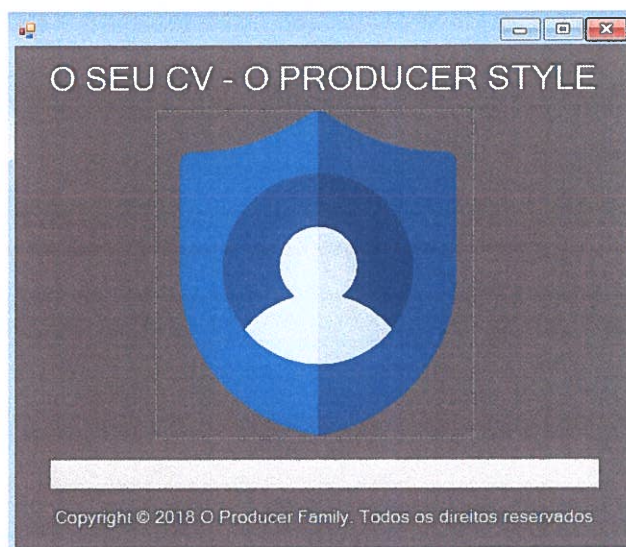


Imagem 67 – Formulário inicial da aplicação
Fonte: do autor

Código para o uso das bibliotecas do ADO.NET no formulário splashscreen. Estas linhas são as principais para que sejam acedidas facilmente os recursos do ADO.NET, com o foco principal a linha `using System.Windows.Forms;`, que permite usar os recursos pertencentes ao SQL Server.

```
using System;  
using System.Windows.Forms;  
using System.Data.SqlClient;
```

O bloco de código abaixo serve para criar a conexão “connectionstring” e instanciar a mesma ao objeto `SqlConnection Conn`. É através destas linhas que a aplicação consegue aceder a base de dados.

```
public static string MyConnection = "Data Source=SQLSERVER;Initial  
Catalog=gestaoutilizadores;Integrated Security=True;Connect  
Timeout=15;MultiSubnetFailover=False";  
SqlConnection Conn = new SqlConnection(MyConnection);
```

O bloco de código abaixo serve para verificar se está tudo correto com a conexão. No caso positivo, a conexão é aberta, o objeto `timer1` é inicializado e logo é fechada a conexão, ver imagem 68; no caso negativo, é apresentada uma mensagem de erro “Sem Conexão com o Servidor de Base de Dados SQL Server”, ver Imagem 69.

```
private void splashscreen_Load(object sender, EventArgs e)  
{  
    try  
    {  
        Conn.Open();
```

```

        this.timer1.Start();
        Conn.Close();
    }
    catch (Exception ex)
    {
        string zx = ex.Message;
        DialogResult Retry = MessageBox.Show("Sem Conexão com o Servidor de Base de
        Dados SQL Server", "CONEXÃO COM O SERVIDOR DE BASE DE DADOS SQL
        SERVER", MessageBoxButtons.RetryCancel, MessageBoxIcon.Information);
        if (Retry == DialogResult.Retry)
        {
            splashscreen_Load(sender, e);
        }
        else
        {
            Application.Exit();
        }
    }
}

```

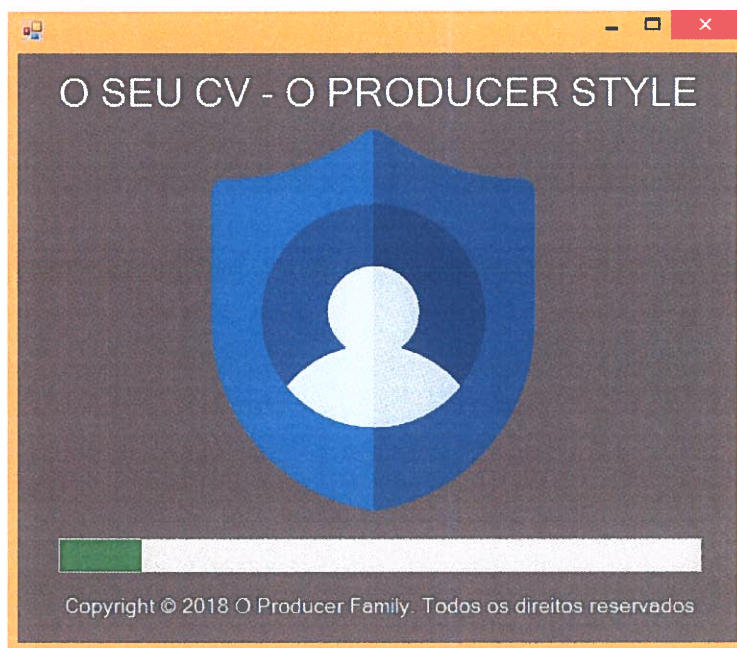


Imagem 68 – Formulário inicial da aplicação em execução após a verificação que está tudo correto/validado para acesso à base de dados

Fonte: do autor

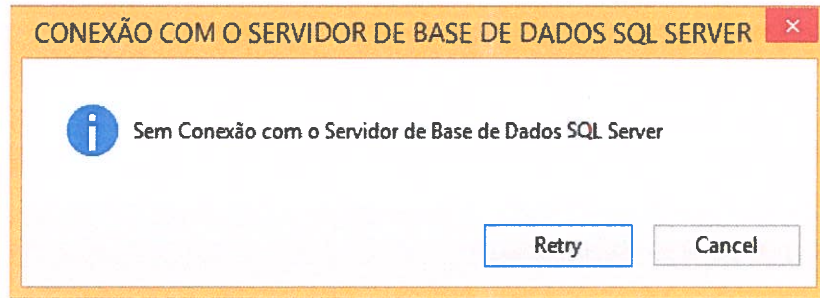


Imagem 69 – Caixa de diálogo apresentando uma mensagem de erro.
Fonte: do autor

A caixa de diálogo acima (informando ao utilizador que não há conexão com o servidor SQLSERVER). Na maioria dos casos ou é porque o utilizador autenticou-se na máquina CLIENT com uma conta de utilizador que não tem acesso à base de dados ou é porque não há comunicação com a máquina SQLSERVER

O bloco de código abaixo serve para verificar se a propriedade Value do objeto progressBar1 é igual a 100. No caso de ser igual a 100, o objeto timer1 pára e é apresentado o formulário osmeusdados substituindo assim o formulário splashscreen.

```
private void timer1_Tick(object sender, EventArgs e)
{
    this.progressBar1.Increment(1);
    if (progressBar1.Value == 100)
    {
        timer1.Stop();
        osmeusdados osmeusdados = new osmeusdados();
        this.Hide();
        osmeusdados.Show();
    }
}
```

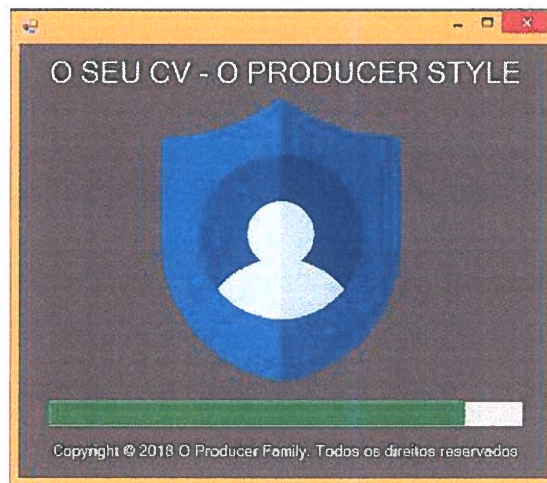


Imagem 70 – Formulário inicial da aplicação em carregamento para aceder ao formulário A MINHA CONTA
Fonte: do autor

Imagem 71 – Formulário A MINHA CONTA no seu estado inicial
Fonte: do autor

Código para o uso de biblioteca do ADO.NET no formulário osmeusdados. Dado que este formulário usa mais recursos, logo mais bibliotecas são necessárias.

```
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.Data.SqlClient;
using System.IO;
using iTextSharp.text;
using iTextSharp.text.pdf;
```

O bloco de código abaixo serve para criar a conexão “connectionstring” e instanciar a mesma ao objeto SqlConnection Conn. É através destas linhas que a aplicação consegue aceder a base de dados.

```
public static string MyConnection = "Data Source=SQLSERVER;Initial Catalog =
gestaoutilizadores; Integrated Security = True; Connect Timeout = 15;
MultiSubnetFailover=False";
SqlConnection Conn = new SqlConnection(MyConnection);
```

As instruções abaixo servem para definir os primeiros passos as serem executados (automaticamente) logo que é apresentado o formulário osmeusdados pela primeira vez.

```
public osmeusdados()
```

```

{
InitializeComponent();
Genero();
EstadoCivil();
btnAtualizar.Visible = false;
btnGuardar.Visible = true;
btnImprimir.Enabled = false;
}

```



Imagem 72 – Execução da função osmeusdados()

Fonte: do autor

O bloco de código abaixo serve para terminar a execução da aplicação quando é clicado no botão sair

```

private void btnSair_Click(object sender, EventArgs e)
{
Application.Exit();
}

```

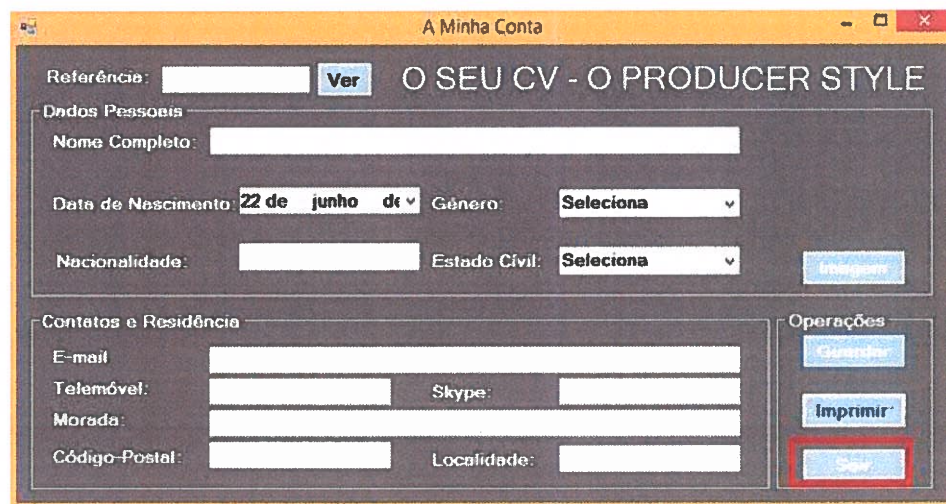


Imagem 73 – Clique no botão sair para execução do evento btnSair_Click

Fonte: do autor

O excerto de código abaixo serve para criar e instanciar uma lista de textos para o controlo txtDropGenero, permitindo assim o utilizador escolher a opção pretendida

```
protected void Genero()
{
    DataTable dt = new DataTable();
    dt.Columns.Add("Id", typeof(char));
    dt.Columns.Add("Genero", typeof(string));
    dt.Rows.Add("0", "Selecione");
    dt.Rows.Add("F", "Feminino");
    dt.Rows.Add("M", "Masculino");
    this.txtDropGenero.ValueMember = "Id";
    this.txtDropGenero.DisplayMember = "Genero";
    this.txtDropGenero.DataSource = dt;
}
```

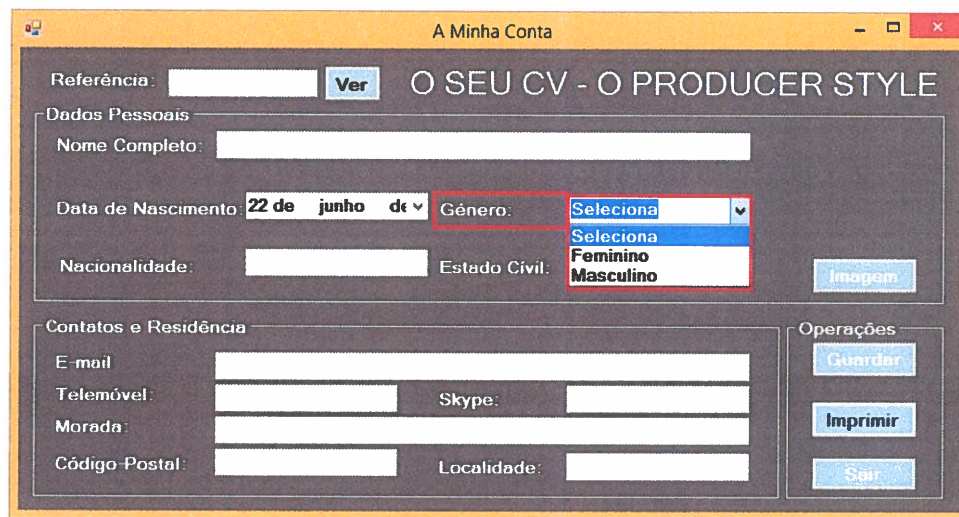


Imagem 74 – Execução do método Genero()
Fonte: do autor

As linhas de código abaixo servem para criar e instanciar uma lista de textos para o controlo txtDropEstadoCivil, permitindo assim o utilizador escolher a opção pretendida

```
protected void EstadoCivil()
{
    DataTable dt = new DataTable();
    dt.Columns.Add("Id", typeof(char));
    dt.Columns.Add("EstadoCivil", typeof(string));
    dt.Rows.Add("0", "Selecione");
    dt.Rows.Add("C", "Casado/a");
    dt.Rows.Add("D", "Divorciado / a");
    dt.Rows.Add("S", "Solteiro / a");
    dt.Rows.Add("V", "Viúvo / a");
    this.txtDropEstadoCivil.ValueMember = "Id";
    this.txtDropEstadoCivil.DisplayMember = "EstadoCivil";
    this.txtDropEstadoCivil.DataSource = dt;
}
```

}

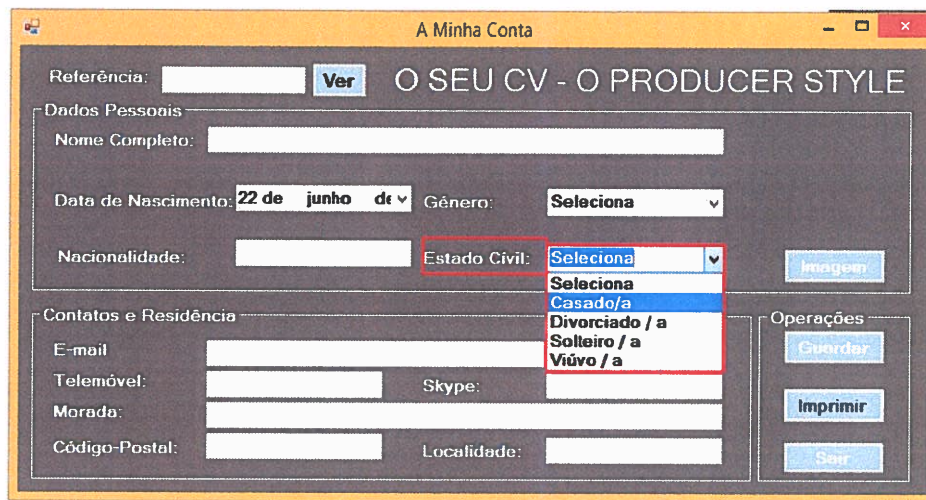


Imagem 75 – Execução do método EstadoCivil()
Fonte: do autor

O bloco de código abaixo serve para inserir um novo registo na base de dados, através da stored Procedure, conforme descrito acima.

```
private void btnGuardar_Click(object sender, EventArgs e)
{
    SqlCommand cmd = new SqlCommand("IUMeusDados");
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Parameters.Add("@IU", SqlDbType.Char).Value = "I";
    cmd.Parameters.Add("@Id", SqlDbType.Int).Value = 0;
    cmd.Parameters.Add("@NomeCompleto", SqlDbType.VarChar).Value =
txtNomeCompleto.Text;
    cmd.Parameters.Add("@DataNascimento", SqlDbType.Date).Value =
txtDataNascimento.Text;
    cmd.Parameters.Add("@Genero", SqlDbType.Char).Value =
txtdropGenero.SelectedValue;
    cmd.Parameters.Add("@Nacionalidade", SqlDbType.VarChar).Value =
txtNacionalidade.Text;
    cmd.Parameters.Add("@EstadoCivil", SqlDbType.Char).Value =
txtdropEstadoCivil.SelectedValue;
    cmd.Parameters.Add("@Email", SqlDbType.VarChar).Value = txtEmail.Text;
    cmd.Parameters.Add("@Telemovel", SqlDbType.VarChar).Value = txtTelemovel.Text;
    cmd.Parameters.Add("@Skype", SqlDbType.VarChar).Value = txtSkype.Text;
    cmd.Parameters.Add("@Morada", SqlDbType.VarChar).Value = txtMorada.Text;
    cmd.Parameters.Add("@CodigoPostal", SqlDbType.VarChar).Value =
txtCodigoPostal.Text;
    cmd.Parameters.Add("@Localidade", SqlDbType.VarChar).Value = txtLocalidade.Text;
    cmd.Parameters.Add("@Imagem", SqlDbType.VarChar).Value = txtLImagem.Text;
    cmd.Connection = Conn;
    try
    {
        Conn.Open();
        txtReferencia.Text = cmd.ExecuteScalar().ToString();
    }
}
```

```

Conn.Close();
    MessageBox.Show("Dados Guardados! A sua Referência é: " +
        txtReferencia.Text + " recomendamos que guarde-a para reutilizar na próxima
        vez.", "Informação", MessageBoxButtons.OK,
        MessageBoxIcon.Exclamation);
Ver();
    MessageBox.Show("Antes de Imprimir os seus dados, verifique que introduziu tudo
    corretamente. Obrigado!", "Atenção", MessageBoxButtons.OK,
    MessageBoxIcon.Information);
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}
}

```

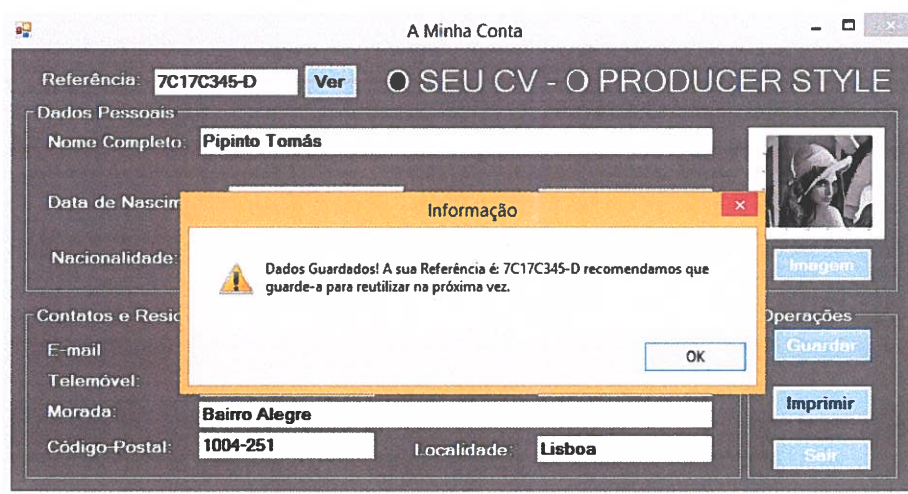


Imagem 76 – Informação quando os dados são guardados.
Fonte: do autor

Caixa de diálogo apresenta a informação quando os dados são guardados na base de dados, isto é, após a execução do evento btnGuardar_Click. Com a mensagem apresentada, certifica-se que os dados foram inseridos com sucesso; caso contrário é apresentada uma mensagem com o erro específico. Pode-se ver também os dados inseridos na base dados.

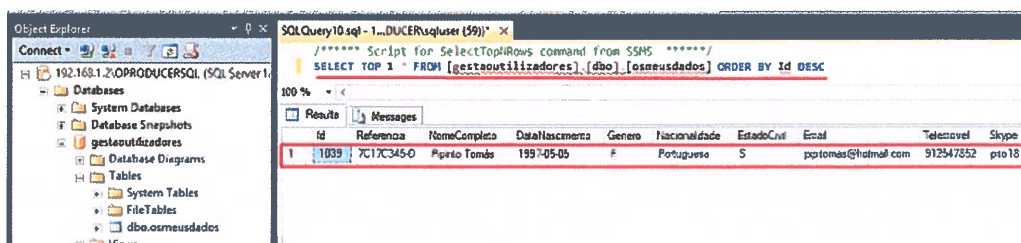


Imagem 77 – Verificação do registo inserido na base de dados
Fonte: do autor

O bloco de código abaixo serve para atualizar um registo já existente na base de dados com a Referência introduzida no controlo txt.Referencia.Text, de igual modo com recurso da stored Procedure.

```
private void btnAtualizar_Click(object sender, EventArgs e)
{
    SqlCommand cmd = new SqlCommand("IUMeusDados");
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Parameters.Add("@IU", SqlDbType.Char).Value = "U";
    cmd.Parameters.Add("@Id", SqlDbType.VarChar).Value = txtReferencia.Text;
    cmd.Parameters.Add("@NomeCompleto", SqlDbType.VarChar).Value =
txtNomeCompleto.Text;
    cmd.Parameters.Add("@DataNascimento", SqlDbType.Date).Value =
txtDataNascimento.Text;
    cmd.Parameters.Add("@Genero", SqlDbType.Char).Value =
txtdropGenero.SelectedValue;
    cmd.Parameters.Add("@Nacionalidade", SqlDbType.VarChar).Value =
txtNacionalidade.Text;
    cmd.Parameters.Add("@EstadoCivil", SqlDbType.Char).Value =
txtdropEstadoCivil.SelectedValue;
    cmd.Parameters.Add("@Email", SqlDbType.VarChar).Value = txtEmail.Text;
    cmd.Parameters.Add("@Telemovel", SqlDbType.VarChar).Value = txtTelemovel.Text;
    cmd.Parameters.Add("@Skype", SqlDbType.VarChar).Value = txtSkype.Text;
    cmd.Parameters.Add("@Morada", SqlDbType.VarChar).Value = txtMorada.Text;
    cmd.Parameters.Add("@CodigoPostal", SqlDbType.VarChar).Value =
txtCodigoPostal.Text;
    cmd.Parameters.Add("@Localidade", SqlDbType.VarChar).Value = txtLocalidade.Text;
    cmd.Parameters.Add("@Imagem", SqlDbType.VarChar).Value = txtLImagem.Text;
    cmd.Connection = Conn;
    try
    {
        Conn.Open();
        cmd.ExecuteNonQuery();
        MessageBox.Show("Dados Atualizados!", "Informação", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        MessageBox.Show("Antes de imprimir os seus dados, verifique que introduziu tudo
corretamente. Obrigado!", "Atenção", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        Conn.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

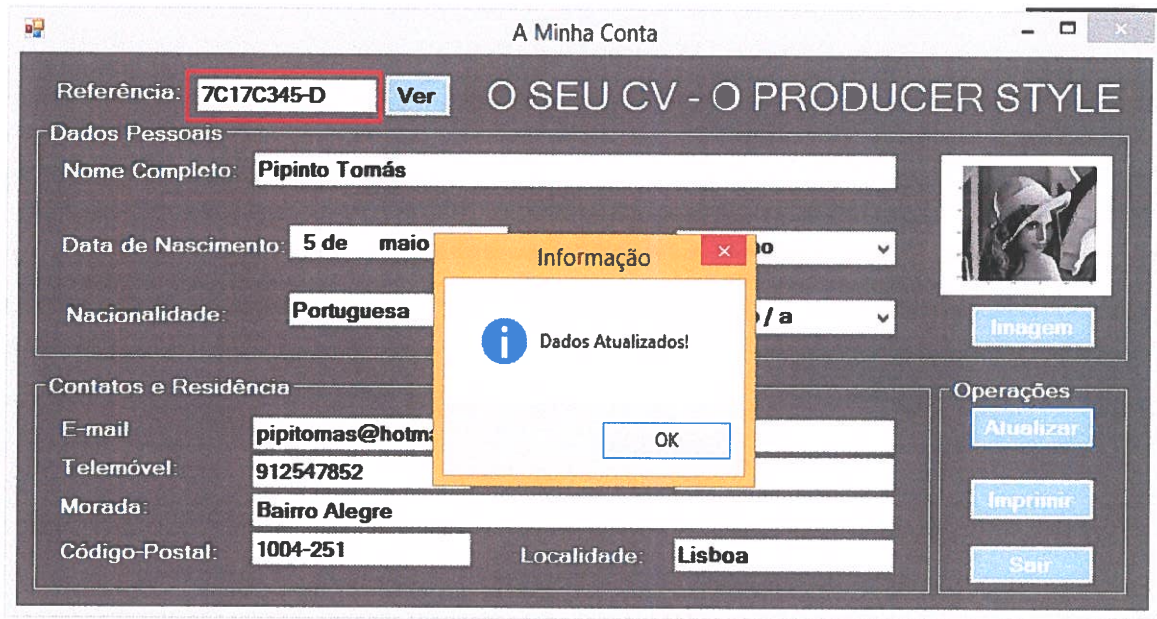


Imagem 78 – Informação quando os dados são atualizados.
Fonte: do autor

Caixa de diálogo apresentado na imagem acima, é apresentada após a concretização do evento `btnAtualizar_Click`. Com a mensagem apresentada, certifica-se que os dados foram atualizados com sucesso; caso contrário é apresentado uma mensagem com o erro específico.

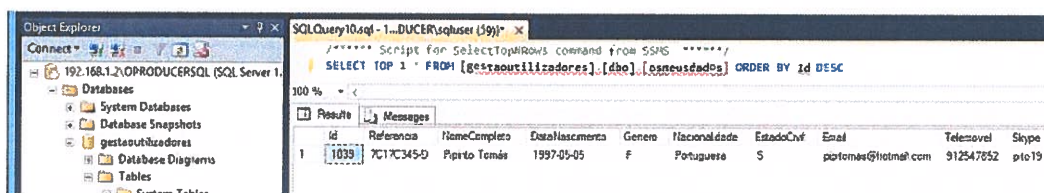


Imagem 79 – Verificação do registo atualiza na base de dados
Fonte: do autor

O bloco de código abaixo serve para inserir um ficheiro de imagem `.png` ou `.jpg`, associando assim este ficheiro com o nome introduzido no controlo `txtNomeCompleto.Text`, guardando a imagem numa pasta especificada e o caminho da mesma na base de dados através do controlo `txtLIimagem.Text`.

```
private void btnImagem_Click(object sender, EventArgs e)
{
    string gId = Guid.NewGuid().ToString();
    string fgId = gId.Substring(0, 10);
    OpenFileDialog InserirImagem = new OpenFileDialog();
    InserirImagem.Filter = "JPG File (*.jpg)*.jpg|PNG File (*.png)*.png|All Files (*.*)*.*";
    if (InserirImagem.ShowDialog() == DialogResult.OK)
    {
        string strImagem = InserirImagem.FileName.ToString();
    }
}
```



```

txtImagem.ImageLocation = strImagem;
if (InserirImagem.CheckFileExists)
    File.Copy(InserirImagem.FileName, @"\\CLIENT\Ficheiros\Imagens\" +
txtNomeCompleto.Text+fgId + ".png");
txtLImagem.Text = @"\\CLIENT\Ficheiros\Imagens\" + txtNomeCompleto.Text +
fgId + ".png";
}
}

```

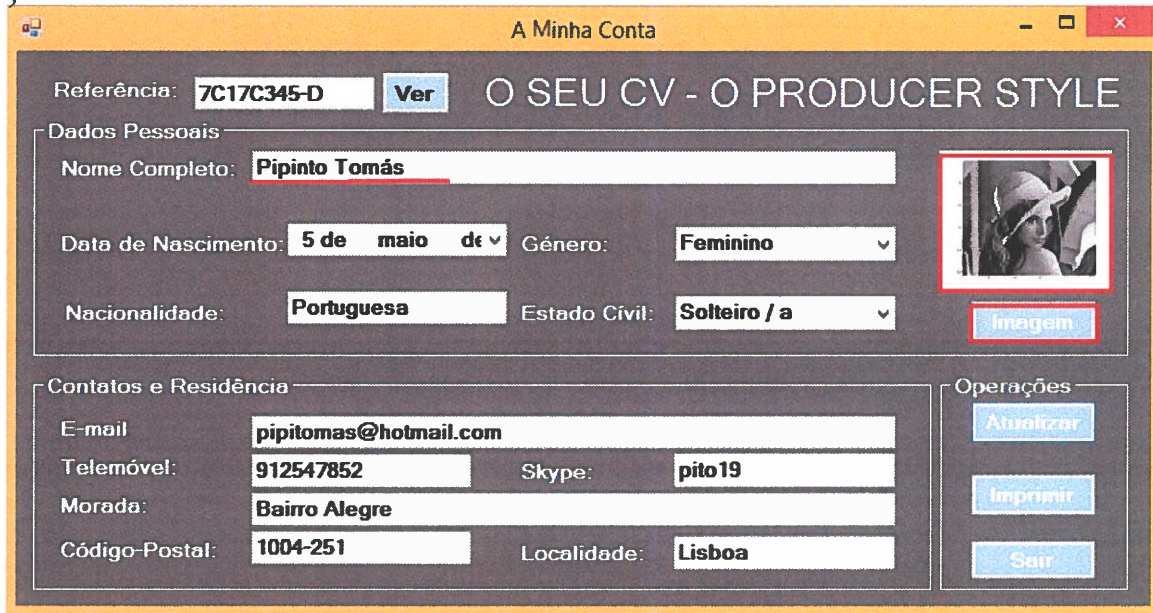


Imagem 80 – Concretização do evento btnImagem_Click()
Fonte: do autor

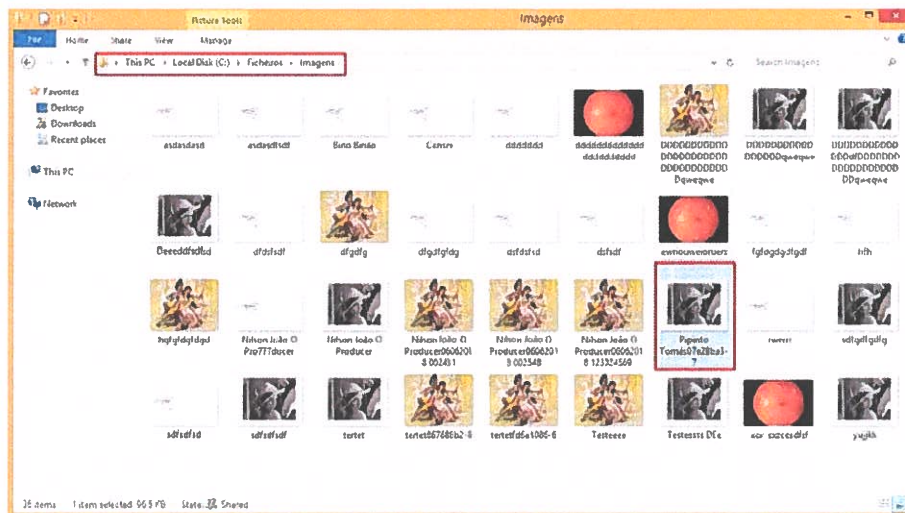


Imagem 81 – Concretização do evento btnImagem_Click()
Fonte: do autor

O bloco de código abaixo serve para fazer busca à base de dados de um registo inserido/atualizado, com base na Referência introduzida no controlo txt.Referencia.Text

```
protected void Ver()
```

```

{
    if (txtReferencia.Text != "")
    {
        SqlCommand cmd = new SqlCommand("SELECTSP");
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.Parameters.Add("@Id", SqlDbType.VarChar).Value = txtReferencia.Text;
        cmd.Connection = Conn;
        Conn.Open();
        SqlDataReader sdr = cmd.ExecuteReader();
        if (sdr.Read())
        {
            txtReferencia.Text = sdr["Referencia"].ToString();
            txtNomeCompleto.Text = sdr["NomeCompleto"].ToString();
            txtDataNascimento.Text = sdr["DataNascimento"].ToString();
            txtdropGenero.SelectedValue = sdr["Genero"].ToString();
            txtNacionalidade.Text = sdr["Nacionalidade"].ToString();
            txtdropEstadoCivil.SelectedValue = sdr["EstadoCivil"].ToString();
            txtEmail.Text = sdr["Email"].ToString();
            txtTelemovel.Text = sdr["Telemovel"].ToString();
            txtSkype.Text = sdr["Skype"].ToString();
            txtMorada.Text = sdr["Morada"].ToString();
            txtCodigoPostal.Text = sdr["CodigoPostal"].ToString();
            txtLocalidade.Text = sdr["Localidade"].ToString();
            txtImagem.ImageLocation = sdr["Imagem"].ToString();
            btnAtualizar.Visible = true;
            btnGuardar.Visible = false;
            btnImprimir.Enabled = true;
        }
        else
        {
            txtNomeCompleto.Text = "";
            txtDataNascimento.Text = "";
            txtdropGenero.SelectedValue = "0";
            txtNacionalidade.Text = "";
            txtdropEstadoCivil.SelectedValue = "0";
            txtEmail.Text = "";
            txtTelemovel.Text = "";
            txtSkype.Text = "";
            txtMorada.Text = "";
            txtCodigoPostal.Text = "";
            txtLocalidade.Text = "";
            txtImagem.ImageLocation = "";
            btnAtualizar.Visible = false;
            btnGuardar.Visible = true;
            btnImprimir.Enabled = false;
        }
        Conn.Close();
    }
}

```

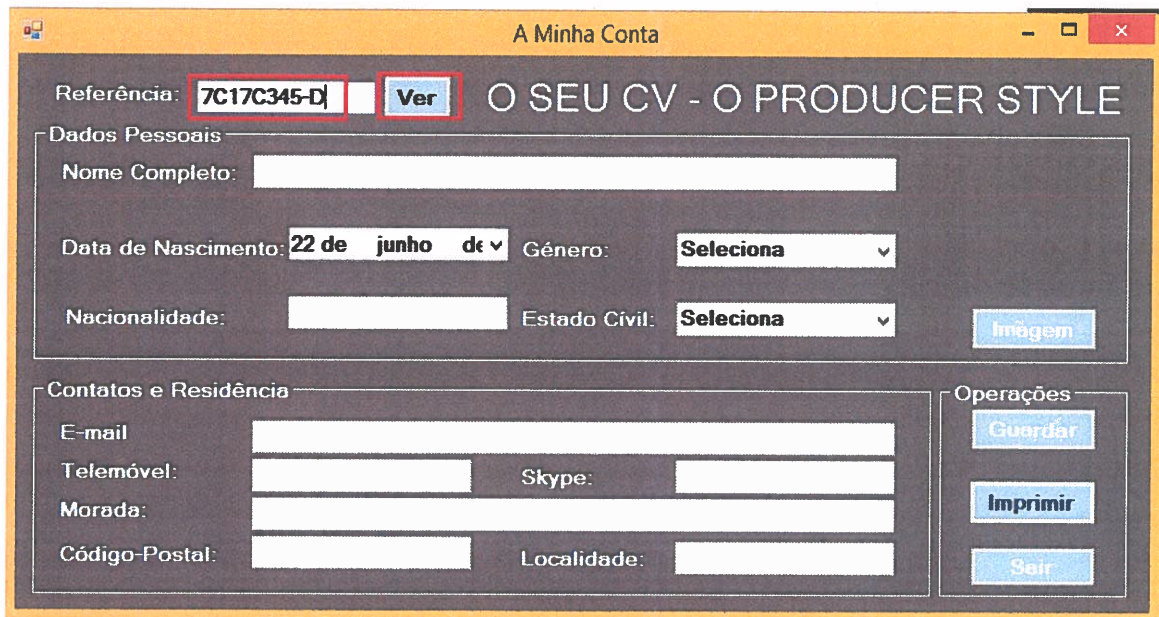


Imagem 82 – Antes de executar o método Ver()
Fonte: do autor

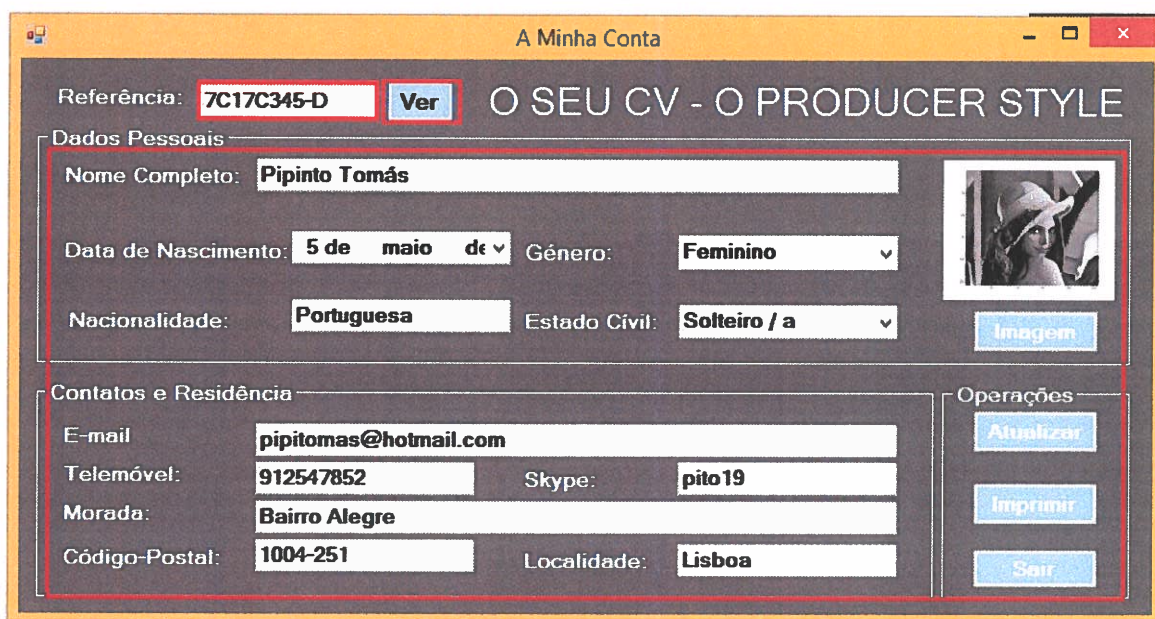


Imagem 83 – Após executar o método Ver()
Fonte: do autor

O bloco de código abaixo serve para gerar um ficheiro .pdf apresentando os dados inseridos por um utilizador e termina de imediato a execução da aplicação.

```
private void Imprimir()
{
    Document doc = new Document();
    PdfWriter.GetInstance(doc, new FileStream("//CLIENT/Ficheiros/Pdfs/" +
    txtNomeCompleto.Text + ".pdf", FileMode.Create));
    doc.Open();
}
```

```

Paragraph p1 = new Paragraph("OS MEUS DADOS");
Paragraph p2 = new Paragraph(" ");
Paragraph p24 = new Paragraph(" ");
Paragraph p25 = new Paragraph(" ");
Paragraph p26 = new Paragraph("Referência: " + txtReferencia.Text);
Paragraph p3 = new Paragraph("Nome Completo: " + txtNomeCompleto.Text);
Paragraph p4 = new Paragraph("Data de Nascimento: " + txtDataNascimento.Text);
Paragraph p5 = new Paragraph("Gênero: " + txtdropGenero.Text);
Paragraph p6 = new Paragraph("Nacionalidade: " + txtNacionalidade.Text);
Paragraph p7 = new Paragraph("Estado Civil: " + txtdropEstadoCivil.Text);
Paragraph p8 = new Paragraph("E-mail: " + txtEmail.Text);
Paragraph p9 = new Paragraph("Skype: " + txtSkype.Text);
Paragraph p10 = new Paragraph("Telemóvel: " + txtTelemovel.Text);
Paragraph p11 = new Paragraph("Morada: " + txtMorada.Text);
Paragraph p12 = new Paragraph("Código Postal: " + txtCodigoPostal.Text);
Paragraph p13 = new Paragraph("Localidade: " + txtLocalidade.Text);
Paragraph p14 = new Paragraph(" ");
Paragraph p15 = new Paragraph(" ");
Paragraph p16 = new Paragraph(" ");
Paragraph p17 = new Paragraph("Lisboa, " + DateTime.Today.ToString());
Paragraph p18 = new Paragraph(" ");
Paragraph p19 = new Paragraph(" ");
Paragraph p20 = new Paragraph(" ");
Paragraph p21 = new Paragraph("Assinar aqui:");
Paragraph p22 = new Paragraph(" ");
Paragraph p23 = new Paragraph("_____");
iTextSharp.text.Image ad = default(iTextSharp.text.Image);
ad = iTextSharp.text.Image.GetInstance(txtLImagem.Text);
ad.ScaleToFit(150, 150);
doc.Add(p1);
doc.Add(ad);
doc.Add(p2);
doc.Add(p26);
doc.Add(p3);
doc.Add(p4);
doc.Add(p5);
doc.Add(p6);
doc.Add(p7);
doc.Add(p8);
doc.Add(p9);
doc.Add(p10);
doc.Add(p11);
doc.Add(p12);
doc.Add(p13);
doc.Add(p14);
doc.Add(p15);
doc.Add(p16);
doc.Add(p17);
doc.Add(p18);
doc.Add(p19);

```

```

doc.Add(p20);
doc.Add(p21);
doc.Add(p22);
doc.Add(p23);
doc.Add(p24);
doc.Add(p25);
doc.Close();
MessageBox.Show("Foi criado um ficheiro pdf com os seus dados em
'C:/Ficheiros/Pdfs'. Obrigado", "Informação", MessageBoxButtons.OK,
MessageBoxIcon.Information);
Application.Exit();
}

```

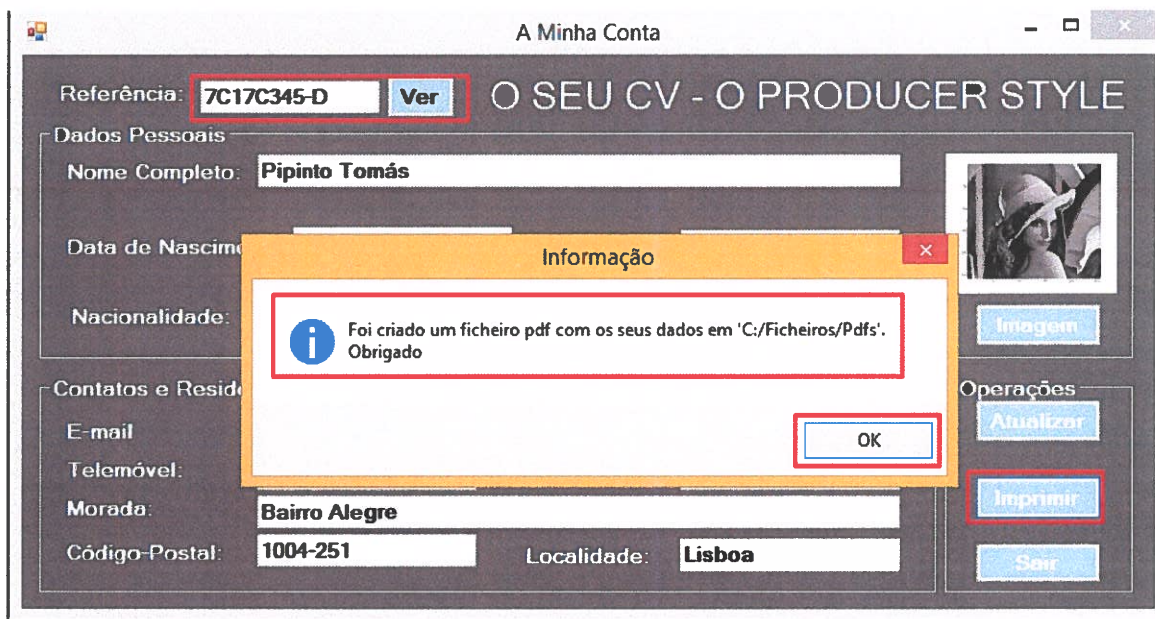


Imagem 84 – Criação de um ficheiro .pdf após executar o método Imprimir()
 Fonte: do autor

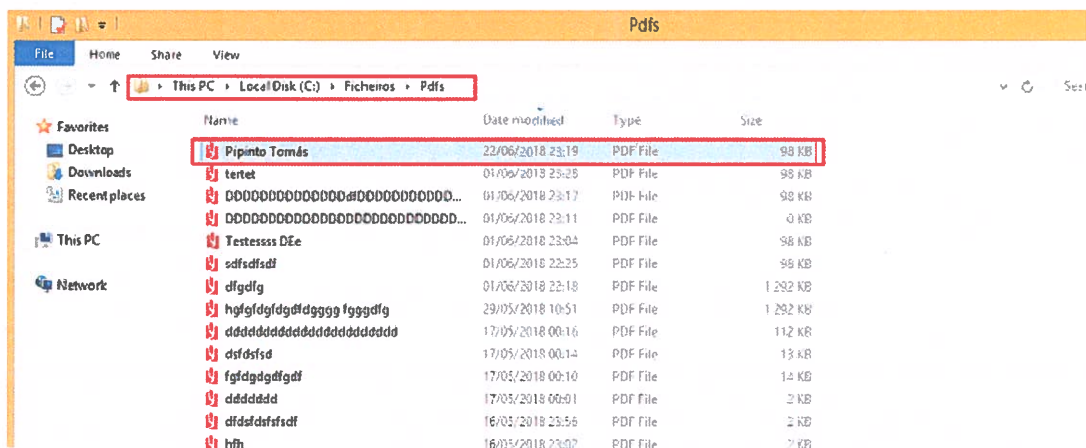


Imagem 85 – Verificação do ficheiro .pdf criado após executar o método Imprimir()
 Fonte: do autor

OS MEUS DADOS



Referência: 7C17C345-D
Nome Completo: Pipinto Tomás
Data de Nascimento: 5 de maio de 1997
Gênero: Feminino
Nacionalidade: Portuguesa
Estado Civil: Solteiro / a
E-mail: pipitomas@hotmail.com
Skype: pito19
Telemóvel: 912547852
Morada: Bairro Alegre
Código Postal: 1004-251
Localidade: Lisboa

Lisboa, 22/06/2018 00:00:00

Assinar aqui.

Imagem 86 – Ficheiro .pdf aberto num leitor de ficheiros .pdf
Fonte: do autor

O bloco de código abaixo serve para executar a função `Imprimir()`, clicando no botão `Imprimir`

```
private void btnImprimir_Click(object sender, EventArgs e)
{
    Imprimir();
}
```

O bloco de código abaixo serve para executar a função `Ver()`, clicando no botão `Ver`

```
private void btnVer_Click(object sender, EventArgs e)
{
    Ver();
}
```

CONCLUSÃO

Com a concretização do objetivo do projeto, desde a implementação do laboratório até a execução da aplicação com a tecnologia ADO.NET, chegou-se a conclusão que a Virtualização possibilita a criação e alojamento de variadas instâncias de máquinas virtuais, é uma tecnologia muito útil para ambientes de teste e ou grandes produções, SQL devido a sua padronização universal a gestão de bases de dados tende a ser mais simples e o ADO.NET sendo uma tecnologia ainda por ser mais explorada, permite o desenvolvimento de aplicações e serviços aplicativos para uso empresarial ou particular.

O objetivo do projeto foi concretizado quando através da aplicação na máquina CLIENT é possível aceder à base de dados alojada na máquina SQLSERVER e também permitir gerar um ficheiro .pdf com apresentação dos dados inseridos.

REFERÊNCIA BIBLIOGRÁFICAS

- ActiveX Data Object.NET (ADO.NET)*. (s.d.). Obtido de Techopedia:
<https://www.techopedia.com/definition/24703/activex-data-object-net-ado-net>
- Antonio, A. M. (s.d.). *Benefícios e Limitações da Cloud Computing*. Obtido de PMG Academy:
<http://www.pmgacademy.com/pt/blog/artigos/beneficios-cloud-computing>
- Freitas, M. d. (9 de Junho de 2011). *Conceitos do SQL*. Obtido de Marcel Mesmo:
<http://marcelmesmo.blogspot.pt/2011/06/conceitos-do-sql.html#.WwkpNkgvzIV>
- Gomes, P. E. (s.d.). *Linguagem SQL*. Obtido de Banco de Dados:
<http://ehgomes.com.br/disciplinas/bdd/sql.php>
- Húngaro, L. (s.d.). *Desenvolvimento - ADO.NET*. Obtido de Linha de Código:
<http://www.linhadecodigo.com.br/artigo/435/uma-visao-geral-do-adonet.aspx>
- Informática.com. (s.d.). *O que é Hypervisor e quais são os tipos de Hypervisor que existem?*
Obtido de <https://faqinformatica.com/o-que-e-hypervisor-tipos-de-hypervisor/>
- Laureano, M. A., & Maziero, C. A. (s.d.). Obtido de
https://www.researchgate.net/profile/Carlos_Maziero/publication/237681120_Virtualizacao_Conceitos_e_Aplicacoes_em_Seguranca/links/0a85e539b24b4524e7000000/Virtualizacao-Conceitos-e-Aplicacoes-em-Seguranca.pdf
- Mattos, D. M. (s.d.). Obtido de
https://www.researchgate.net/profile/Diogo_Menezes3/publication/255657320_Virtualizacao_VMWare_e_Xen/links/5617ce6f08aeccf998eae4eb/Virtualizacao-VMWare-e-Xen.pdf
- Mayo, J. (7 de Junho de 2016). *Lesson 01: Introduction to ADO.NET*. Obtido de C# Station:
<https://csharp-station.com/Tutorial/AdoDotNet/Lesson01>
- Mehra, P. (20 de Julho de 2009). *What is ADO.NET?* Obtido de C-Sharp Corner:
<https://www.c-sharpcorner.com/uploadfile/puranindia/what-is-ado-net/>
- Mell, P., & Grance, T. (s.d.). *The NIST Definition of Cloud Computing*. Obtido de National Institute of Standards and Technology:
<https://www.nist.gov/document-13184>

- Net, O. d. (s.d.). *Quais são as vantagens e as desvantagens de uma virtualização?* Obtido de Oficiãna da Net: <https://www.oficinadanet.com.br/post/13055-quais-sao-as-vantagens-e-as-desvantagens-de-uma-virtualizacao>
- Rosa, D. (s.d.). *Os principais atores da nuvem segundo o NIST*. Obtido de Jornada para Nuvem: <http://jornadaparanuvem.com.br/fundamentos-de-cloud-computing/os-principais-atores-da-nuvem-segundo-o-nist/>
- Rouse, M. (s.d.). *SQL (Structured Query Language)*. Obtido de Search SQL Server: <https://searchsqlserver.techtarget.com/definition/SQL>
- Silva, D. (22 de OUTUBRO de 2014). *Infográfico: A História de Cloud computing*. Obtido de Blog Cloud Market - Tudo Sobre Cloud Computing: <https://www.cloudmarket.com.br/blog/cloud-computing/infografico-historia-cloud-computing/>
- Taxonomy, C. (s.d.). *About Cloud Taxonomy*. Obtido de Cloud Taxonomy: <http://cloudtaxonomy.opencrowd.com/taxonomy/about/>
- VMware. (s.d.). *Virtualização*. Obtido de <https://www.vmware.com/br/solutions/virtualization.html>
- W3Schools. (s.d.). *Introdução ao SQL*. Obtido de W3Schools: https://www.w3schools.com/sql/sql_intro.asp