

Projeto Global



WorkMatic

Ricardo Miguel Fernandes Tavares (N.º 2052)

Instituto Superior de Tecnologias Avançadas
Campus Lumiar, Lisboa

Documento submetido para a Licenciatura de: *Informática*
Coordenador e Orientador: Prof. Dr. Pedro Brandão
Orientador da Especialidade: Professor José Neves
Ano letivo 2017/2018

Dedicatória

Dedico este projeto, em primeiro lugar, à minha mãe que tem sofrido desde sempre para eu ser a pessoa que sou hoje.

Em seguida dedico aos “The Three Geeks” espero durante vários anos continuemos a poder chatear-nos mutuamente.

Por fim, mas não menos importante, dedico à minha colega Cândida Santos.

Agradecimentos

Agradeço aos professores e colaboradores do Instituto Superior de Tecnologias Avançadas (ISTEC), pela disponibilidade e apoio demonstrado ao longo dos 3 anos de curso, em particular à Dona Cristina Almeida, um anjo em forma de senhora.

Para além de dedicar este projeto à minha Mãe, também quero agradecer-lhe pelo seu amor incondicional. À Cândida Santos agradeço a força que me deu para iniciar esta aventura e à Joana Duarte para a finalizar. Aos “The Three Geeks” ... não tenho palavras para expressar a nossa amizade.

Não podia acabar os agradecimentos sem referir o estimado Professor José Neves, foi um grande prazer conhecê-lo.

Por fim, um muito obrigado a todos os que ajudaram a que este projeto fosse concluído.

Resumo

O objetivo deste projeto consiste em criar uma rede virtual, que suporta uma infraestrutura com várias máquinas virtuais num domínio e um programa que retira partido da mesma. O programa usa uma base de dados, que é armazenada e gerida pela máquina virtual, a qual tem o SQL instalado.

Ao usar o *VMware Workstation* é possível criar a rede virtual, que é composta por três servidores e um cliente. Um servidor com a função de controlador de domínio (ISTEC.LOCAL), que faz a gestão de toda a *Active Directory* e dos seus recursos. Um servidor com função de *routing*, que permite o acesso à rede interna e externa das restantes máquinas virtuais. Um servidor com função de SQL, que faz a gestão e armazenamento da base de dados usada pelo programa *WorkMatic*. Um cliente, com função de executar o *WorkMatic*, o qual contém o sistema operativo *Windows 8.1*.

Com a criação e respetiva configuração de toda a rede virtual é possível ter todas as máquinas a comunicar entre si e, em simultâneo. Conseguindo, assim, executar um programa onde é possível fazer a gestão de clientes e de encomendas.

Informação para o acesso ao domínio.

<i>Username</i>	<i>Password</i>
ISTEC\Administrator	P@ssw0rd
ISTEC\UserApp	P@ssw0rd
ISTEC\UserNotApp	P@ssw0rd

Palavras-Chave: virtualização, SQL, active directory, VMWare, windows server 2012

Abstract

The goal of this project is to create a virtual network, which supports an infrastructure with multiple virtual machines on a domain and a program that takes advantage of the this one. The program uses a database that is stored and managed by the virtual machine with has SQL installed.

To use VMware Workstation, it's possible to create a virtual network, which is composed of three servers and a client. A server that has a domain controller function (ISTEC.local), which makes the management of the entire Active Directory and its resources. A server with routing function, which allows access to internal and external network of other virtual machines. A server with SQL function, which makes the management and storage of the database used by the program WorkMatic. A customer with a function to execute the WorkMatic, which contains the Windows operating system 8.1.

With the creation and configuration of the entire virtual network it's possible to have all machines communicating between themselves and at the same time. So, it's possible to execute a program and make the management of customers and orders.

Information for the access to the domain.

<i>Username</i>	<i>Password</i>
ISTEC\Administrator	P@ssw0rd
ISTEC\UserApp	P@ssw0rd
ISTEC\UserNotApp	P@ssw0rd

Keywords: Virtualization, SQL, active directory, VMWare, windows server 2012

Abreviaturas

VM - *Virtual Machine* (Máquina Virtual)

IBM - *International Business Machines*

SaaS - *Software as a Service*

PaaS - *Platform as a Service*

IaaS - *Infrastructure as a Service*

SQL - *Structured Query Language*

WPF - *Windows Presentation Foundation*

SEQUEL - *Structured English Query Language*

LINQ - *Language Integrated Query*

IDE - *Integrated Development Environment*

IT - *Internet Technology*

ISTEC - Instituto Superior de Tecnologias Avançadas

SSD – *Solid-State Drive*

RAM - *Random Access Memory*

GB – *Gigabyte*

Bios - *Basic Input / Output System*

IT - *Information Technology*

NAT - *Network Address Translation*

RRAS - *Routing and Remote Access Service*

DVD - *Digital Video Disc*

Índice

Dedicatória.....	III
Agradecimentos	V
Resumo	VII
Abstract.....	IX
Abreviaturas.....	XI
Índice.....	XIII
Lista de Imagens	XV
Lista de tabelas.....	XVI
Introdução	1
Capítulo I	3
Estado de Arte.....	3
Virtualização	3
Hypervisors	4
VMware Workstation.....	5
Microsoft Windows Server 2012	5
Cloud Computing.....	6
SQL.....	7
.NET Framework	8
WPF	8
Microsoft Visual Studio.....	9

LINQ.....	9
Capítulo II.....	11
Contextualização.....	11
Capítulo III.....	13
Configurações do Laboratório	13
Desenvolvimento do Laboratório	14
Instalação do VMWare Workstation Pro 12.....	14
Instalação das Máquinas Virtuais	15
Configuração do Domain Controller (DC01)	17
Servidor Routing.....	18
Servidor SQL	19
Máquina virtual CLIENT.....	20
Capítulo IV.....	23
Desenvolvimento de <i>WorkMatic</i>	23
Ficheiros “MainWindow.xaml” e “MainWindow.xaml.cs”	24
Ficheiros “FormWindows.xaml” e “FormWindows.xaml.cs”	24
Ficheiros “OrderWindows.xaml” e “OrderWindows.xaml.cs”	24
Ficheiros “ProductWindow.xaml” e “ProductWindow.xaml.cs”	25
Ficheiro “WorkMatic.edmx”	25
Classe “ExportPdf.cs”	26
Classe “MoneyConverter.cs”	26
Conclusão.....	27

Referências Bibliográficas	29
Anexos	31
Anexo I - Código do ficheiro WorkMatic.sql.....	31
Anexo II - Código do ficheiro ClientId.sql.....	32
Anexo III - Código do ficheiro MainWindow.xaml	32
Anexo IV - Código do ficheiro MainWindow.xaml.cs.....	32
Anexo V - Código do ficheiro OrderWindow.xaml	34
Anexo VI - Código do ficheiro OrderWindow.xaml.cs.....	36
Anexo VII - Código do ficheiro FormWindow.xaml	41
Anexo VIII - Código do ficheiro FormWindow.xaml.cs.....	43
Anexo IX - Código do ficheiro ProductWindow.xaml.....	48
Anexo X - Código do ficheiro ProductWindow.xaml.cs.....	49
Anexo XI - Código da classe ExportPdf.cs.	52
Anexo XII - Código da classe MoneyConverter.cs.	56
Anexo XIII - Código do ficheiro WorkMatic.edmx.....	56

Lista de Imagens

Imagem 1: Hypervisor tipo 1	4
Imagem 2: Hypervisor tipo2	5
Imagem 3 Instalação VMware 1	14
Imagem 4 Instalação VMware 2	14

Lista de tabelas

Tabela 1 Estrutura das VM.....	16
--------------------------------	----

Introdução

Neste projeto pretendo desenvolver uma rede virtual com um domínio Microsoft e um programa em WPF/C#.

A rede virtual é composta por três servidores e um cliente. Do domínio Microsoft (ISTEC.LOCAL) pretendo aproveitar todas as suas vantagens, autenticação, centralização, segurança, entre outras.

A criação deste programa tem como objetivo ajudar a solucionar o problema de gestão de clientes e encomendas das pequenas empresas de distribuição de garrafas de gás. Este programa permite as empresas poderem consultar, adicionar, modificar os dados referentes aos clientes, produtos, encomendas e criar faturas.

O nome do programa é WorkMatic. Este nome nasce da junção das palavras *Work* (trabalho, em inglês) e *Automatic* (Automático, em inglês) e, doravante, será assim referido.

Capítulo I

Estado de Arte

Virtualização

Virtualização é uma tecnologia, que tem o propósito de simular uma plataforma *hardware* com um componente de *software*. Para isso, o *software* cria uma máquina virtual (*virtual machine*, VM) que simula um computador tradicional, com todas as suas características. Possibilitando assim a instalação de sistemas operativos e de todo o tipo de *software* que funcionam nas máquinas tradicionais. (Ferreira, 2015, p. 81)

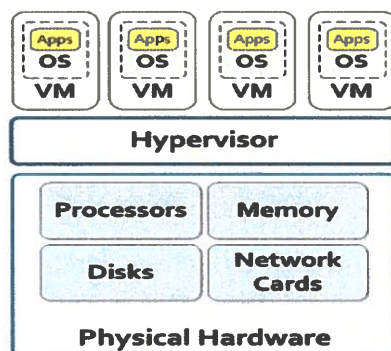
O conceito memória virtual remota a 1950, quando um grupo na Universidade de *Manchester* introduziu a substituição automática de página no sistema Atlas. A implementação deste princípio de paginação criou um método onde se pode armazenar os dados e transmiti-los para cima e para baixo na hierarquia da memória, o que não existia no Atlas. Proporcionou, assim, o primeiro protótipo funcional de memória virtual. O termo máquina virtual surgiu por volta da década de 60. Contribuindo para isso a IBM, que por volta de 1967, lançou o *System/360* modelo 67, o primeiro sistema com memória virtual integral. O modelo 67 foi o primeiro a integrar o conceito de auto virtualização conseguida através de um conjunto de instruções programadas no processador. Este método foi aperfeiçoado em modelos posteriores. Durante a década de 60 do século XX, a IBM produziu vários avanços no campo da virtualização conseguindo que um computador executasse máquinas virtuais. A virtualização é uma tecnologia existente há décadas, apesar de durante as décadas de 60 e 70, a virtualização ser popular tanto na área da pesquisa como no mercado comercial. Durante os anos 80 e 90, esse interesse praticamente desapareceu. Uma das causas, foi o aparecimento dos computadores pessoais. (Brandão, 2018, pp. 1,2)

Nos anos 90, os computadores e servidores físicos dominaram o mercado. Durante vários anos a maioria das empresas adquiriu equipamentos de um único fabricante (marca), que não possibilitavam a utilização de *hardware* de fabricantes diferentes. Quando as empresas começaram a atualizar as infraestruturas de IT depararam-se com um problema. Embora no

mercado existe-se uma vasta oferta de serviços mais económicos, oferecidos por vários fabricantes, as empresas estavam limitadas pelo *hardware* do fabricante que utilizam. Esta situação prejudicou as empresas tanto financeiramente como na evolução da sua infraestrutura de IT. A solução surgiu de uma ideia antiga, a virtualização, mas desta vez com várias soluções atuais e robustas. Ao produzir diversos recursos a partir de um único servidor ou máquina, a virtualização melhora a escalabilidade permitindo assim uma menor utilização de servidores, menos custos na manutenção das infraestruturas e menor consumo de energia. Desde então, até a atualidade, a virtualização é cada vez mais uma realidade incontornável no mundo das IT, sendo já dominante no mesmo. (Red Hat, Inc., 2019, p. 2)

Hypervisors

Hypervisor tipo 1 ou *Baremetal*, é virtualização ao nível do *hardware* o que significa que as máquinas virtuais são instaladas diretamente na máquina física. (Tanamachi, de Oliveira, & Galvani, 2013, p. 2).



Type 1

Imagem 1: Hypervisor tipo 1

Fonte: http://ibm.com/developerworks/br/websphere/techjournal/1206_dejesus/1206_dejesus.html

Hypervisor tipo 2, é virtualização ao nível do sistema operativo o que significa que as máquinas virtuais são instaladas num sistema operativo já instalado na máquina física. (Tanamachi, de Oliveira, & Galvani, 2013, pp. 2,3).

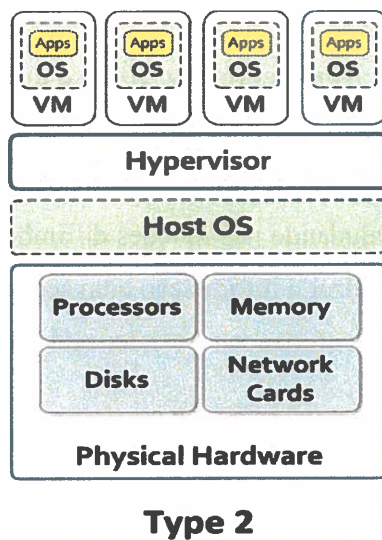


Imagem 2: Hypervisor tipo2

Fonte: http://ibm.com/developerworks/br/websphere/techjournal/1206_dejesus/1206_dejesus.html

VMware Workstation

O *VMware Workstation* é um *Hypervisor* tipo 2, criado pela empresa VMware, é o *software* do seu gênero mais usado hoje em dia, devido a todas as ferramentas que disponibiliza. Com o *VMware Workstation* que já vai na versão 15 é possível virtualizar sistemas operativos *Windows* em máquinas físicas com sistemas operativos *Linux* e vice-versa. (VMware, Inc, 2018, pp. 1,2)

Microsoft Windows Server 2012

O *Microsoft Windows Server 2012* é um sistema operativo para servidores, versátil, completo e poderoso, desenvolvido através de vários aperfeiçoamentos feitos a partir do sistema operativo *Windows Server 2008 R2* pela Microsoft. O *Windows Server 2012* e o *Windows 8* foram desenvolvidos no mesmo projeto. Um como sistema operativo para servidores e o outro para clientes e, por esse motivo, partilham recursos. Esses recursos estão no código base que têm em comum e em várias outras áreas do sistema operativo, como no aspeto gráfico, segurança, rede, armazenamento e a gestão do sistema. Assim, é possível aplicar

uma boa parte do conhecimento sobre *Windows 8*, ao utilizar o *Windows Server 2012*. (Stanek, 2014)

Active Directory no *Windows Server 2012*, é uma fonte de informação hierática que contém dados sobre utilizadores, ficheiros e outros objetos da rede, tais como pastas partilhadas, impressoras, etc... A informação é disponibilizada tanto para administradores como para os utilizadores, ajudando nas funções de ambos. Os administradores, graças a esta ferramenta, podem disponibilizar a informação com segurança e regras. Para os utilizadores, a informação para a qual estão autorizados a aceder está mais acessível. (Rosa, 2013, pp. 77,78)

Cloud Computing

Cloud Computing é um modelo que permite o acesso ubíquo, conveniente e a pedido, através da rede, a um conjunto de recursos de computação partilhada (rede, servidores, armazenamento, aplicações, serviços e etc.), que podem ser rapidamente provisionados ou liberados, com um mínimo de esforço e sem interação com o fornecedor. (Mell & Grance, 2011, p. 2)

O *Cloud Computing* tem cinco características:

- *On-demand self-service* (Autosserviço a pedido) – Possibilidade de se poder obter recursos de computação sem interagir com outra pessoa, basta usar o *service provider*;
- *Broad network access* (acesso generalizado à rede) – Todos os serviços têm de estar disponíveis em rede e ser facilmente acedidos através de ferramentas comuns, como por exemplo, o *browser web* podendo ser utilizado em multiplataformas;
- *Resource pooling* (acesso partilhado a recursos) - Os serviços são partilhados por vários clientes, podendo os mesmos associar e dissociar dinamicamente os serviços conforme as suas necessidades. Normalmente, o cliente não tem controlo da localização física dos serviços, mas mesmo assim, é possível saber de modo abstrato que *datacenter* está a usar;
- *Rapid elasticity* (Elasticidade rápida) – Os serviços são disponibilizados ou revogados de forma flexível, em alguns casos automaticamente conforme a necessidade do cliente;
- *Measured service* (Serviço medido) – Os sistemas *Cloud* são automaticamente controlados e otimizados através de mecanismos de medida. Assim, possibilitando uma melhor transparência na faturação onde o cliente só paga o que utiliza. (Ferreira, 2015, pp. 16-18)

Três modelos de serviço:

- *Software as a Service (SaaS) (Software como um Serviço) – A possibilidade de aceder a uma aplicação que funciona numa infraestrutura de computação. A capacidade de poder aceder as aplicações através de vários dispositivos, usando normalmente um browser. O utilizador não gere nem controla a infraestrutura de computação;*
- *Platform as a Service (PaaS) (Plataforma como um Serviço) – A capacidade de colocar na Cloud aplicações criadas ou compradas pelo utilizador, desde que as suas linguagens, bibliotecas, serviços e ferramentas, sejam suportadas pelo fornecedor;*
- *Infrastructure as a Service (IaaS) (Infraestrutura como um Serviço) – A hipótese de instalar e gerir recursos fundamentais de computação, como processador, armazenamento, memória entre outro. Sendo possível ainda instalar software sobre os recursos que escolhemos. (Ferreira, 2015, pp. 19,20)*

Quatro modelos de implementação:

- *Private Cloud (Cloud privada) – Infraestrutura de Cloud que é providenciada para uso de uma única organização. Pode ser detida, gerida e manipulada pela organização, por uma entidade exterior ou uma combinação de ambas;*
- *Public Cloud (Cloud pública) – Infraestrutura de Cloud que é providenciada para utilização do público em geral. Pode ser detida, gerida e manipulada por uma organização ou várias organizações;*
- *Hybrid Cloud (Cloud híbrida) – Infraestrutura de Cloud que é uma combinação de dois ou mais modelos de implementação (privada, pública ou comunitária), mesmo estando separadas partilham tecnologia *standard* ou proprietária o que permite portabilidade de dados ou aplicações;*
- *Community Cloud (Cloud Comunitária) – Infraestrutura Cloud que é providenciada para o uso exclusivo de uma comunidade de utilizadores que partilham determinadas preocupações como, por exemplo, Universidades. Pode ser detida, gerida e manipulada por uma ou mais organizações da comunidade. (Ferreira, 2015, pp. 21,22)*

SQL

A Structured Query Language (SQL) teve origem num artigo publicado no Association of Computer Machinery (ACM) journal, com título de “A Relational Model of Data for Large Shared Data Banks”, escrito por F. Codd, em junho de 1970. O modelo, criado por F. Codd é, ainda hoje, usado como apoio para qualquer Sistema Gestor de Base de Dados Relacional (SGBDR). A primeira empresa a implementar uma linguagem a partir do artigo do F. Codd foi a IBM, que criou a Linguagem Structured English Query Language (SEQUEL). Depois, de vários desenvolvimentos deu origem ao SQL o qual foi implementado comercialmente pela Oracle Corporation. Existem 5 gerações de linguagens de programação, estas são:

1ª Geração – Código Máquina

2ª Geração – Assembly

3ª Geração – Pascal, C, Cobol, Fortran, Basic

4ª Geração – SQL

5ª Geração – C++, Java Delphi, Visual Basic

O facto da linguagem SQL ser de 4ª Geração é devido a ter como objetivo a manipulação de dados em tabelas relacionais, logo todos os comandos da linguagem SQL correspondem a instruções que manipulam ou acedem a dados existentes na base de dados. (Damas, 1999, pp. 2-4)

.NET Framework

.NET Framework, foi desenvolvida pela *Microsoft Corporation*, com o objetivo de centralizar todas as suas ferramentas de desenvolvimento de aplicações numa única plataforma. Esta plataforma, pode ser usada em qualquer dispositivo onde esteja instalada. (Microsoft Corporation, 2018, pp. 1-4)

WPF

O *Windows Presentation Foundation* (WPF) é uma interface de estrutura gráfica, que foi criada pela *Microsoft Corporation* e está integrada nas *.NET Framework* mais recentes. A função do WPF é dar suporte ao desenvolvimento gráfico das aplicações. Permitindo um melhor controlo e personalização de todos os elementos gráficos e tornando as aplicações mais atrativas e seguras. (Microsoft Corporation, 2018, pp. 1,2)

Microsoft Visual Studio

Microsoft Visual Studio é um Ambiente de Desenvolvimento Integrado (*Integrated Development Environment* ou IDE, em inglês) criado pela *Microsoft Corporation* para o desenvolvimento de *software*. Permite o uso integrado da plataforma *.NET Framework* e o uso de várias linguagens de programação. Tem inúmeros recursos para os programadores desenvolverem os seus *softwares* e atualmente funciona em vários sistemas operativos. (Microsoft Corporation, 2018, pp. 1,2)

LINQ

O *Language Integrated Query* (LINQ) é um conjunto de tecnologias que permite a integração de recursos de consulta de dados na linguagem C# que foi introduzida com o *.NET Framework 3.5*. O LINQ possui várias funcionalidades que permitem melhorar a segurança e facilitar as consultas às bases de dados quando estamos a implementar programas em C#. Anteriormente, era necessário recorrer a outras linguagens para fazer estas consultas, o que prejudicava quer os programas quer os programadores. Os programas ficam mais seguros porque as consultas às bases de dados são requisitadas e terminadas com recurso a um único *software*. Para os programadores, as vantagens são a simplificação dos procedimentos (Microsoft Corporation, 2015, pp. 1-5)

Capítulo II

Contextualização

A escolha para as tecnologias usadas neste projeto resultou dos 3 anos de estudos no ISTECS. Durante estes anos no ISTECS, ao frequentar as diversas disciplinas, aprendi a usar parte das tecnologias que vou utilizar neste projeto. Ao desenvolver os meus conhecimentos aprendi a utilizar novas tecnologias para enriquecer este projeto.

Recorrendo aos conhecimentos adquiridos, resolvi criar o programa *WorkMatic* para pequenas empresas de distribuição de garrafas de gás. Estas empresas precisam de gerir facilmente os seus clientes e encomendas. O *WorkMatic* permite às empresas criar fichas de clientes, onde será possível adicionar, modificar e consultar os dados. Este programa está assente numa rede virtual desenvolvida para o efeito.

Capítulo III

Configurações do Laboratório

Para desenvolver o projeto usei os seguintes componentes de *hardware* e *software*:

Memoria:	- 16 GB RAM
Armazenamento:	- 512 GB (SSD)
Configurações na BIOS:	- Tecnologia de Virtualização ativada
<i>Software</i> :	- <i>VMware Workstation Pro</i> , versão 12 - <i>Windows Server 2012 R2</i> - <i>SQL Server 2012 Management Studio</i> - <i>Visual Studio 2017</i> - <i>Windows 8.1 Pro</i>

Desenvolvimento do Laboratório

Instalação do VMWare Workstation Pro 12

1. Para iniciar a instalação do *VMWare* é necessário fazer o *download* do ficheiro “*setup*” no *site* da *VMware*, executar o “*setup*” com um duplo clique e pressionar o botão “*Next*”.

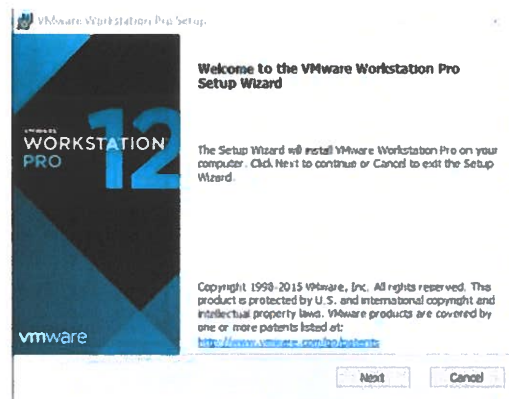


Imagem 3 Instalação VMWare 1

Fonte do autor

2. Passar todas as etapas sem fazer alterações, só pressionado o botão “*Next*” até finalizar a instalação.

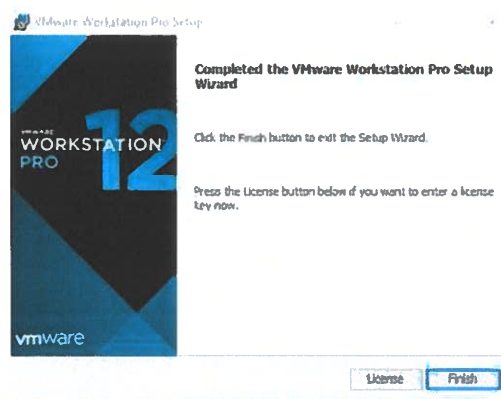


Imagem 4 Instalação VMWare 2

Fonte do autor

Instalação das Máquinas Virtuais

Para instalar as máquinas virtuais não é necessário nenhum *hardware* específico. O *hardware*, que suporta o funcionamento das máquinas virtuais, é do equipamento onde o *VMware WorkStation* está a ser executado.

Vamos começar por criar os três servidores que são o “*DC01*”, o “*ROUTING*” e o “*SQL*”, seguindo os seguintes passos:

1. Executar o programa *VMware* e seleccionar a opção “*Create a New Virtual Machine*” -> “*Next*”
2. Seleccionar a opção “*Typical (recommended)*” -> “*Next*”.
3. Seleccionar a opção “*I will install the operating system later.*” -> “*Next*”.
4. Escolher as opções “*Microsoft Windows*” e “*Windows Server 2012*” -> “*Next*”.
5. Escolher o nome da Máquina Virtual, ex.: “*DC01*” e a localização, ex.: “*C/PG/ Nome_da_VM*” -> “*Next*”.
6. Escolher o tamanho máximo do disco da VM, ex.: “*60 GB*”, e seleccionar a opção “*Store virtual disk as a single file*” -> “*Next*”.
7. Rever as configurações -> “*Finish*”.

Repetir estes passos para os outros 2 servidores tendo o cuidado de mudar o nome e a localização dos mesmos. Em seguida, criar o cliente, o processo é semelhante:

1. Executar o programa *VMware*, e seleccionar a opção “*Create a New Virtual Machine*” -> “*Next*”
2. Seleccionar a opção “*Typical (recommended)*” -> “*Next*”.
3. Seleccionar a opção “*I will install the operating system later.*” -> “*Next*”.
4. Escolher as opções “*Microsoft Windows*” e “*Windows 8*” -> “*Next*”.
5. Escolher o nome da Máquina Virtual, ex.: “*CLIENT*”, e a localização, ex.: “*C/PG/ Nome_da_VM*” -> “*Next*”.

6. Escolher o tamanho máximo do disco da VM, ex.: “60 GB”, e selecionar a opção “*Store virtual disk as a single file*” -> “*Next*”.
7. Rever as configurações -> “*Finish*”.

Com todas as máquinas virtuais instaladas, pressionar o botão direito do rato e escolher a opção “*Settings*”, o que permite fazer as configurações específicas do *hardware*: definir o número de processadores, a quantidade de memória RAM, tipo das placas de rede que vamos utilizar e no drive de DVD selecionar a localização do ficheiro .iso do Sistema Operativo que vamos instalar.

Para verificar a estrutura e configuração das máquinas virtuais, utilizar a Tabela 1 Estrutura das VM que está organizada da seguinte forma: Nome da VM, Sistema operativo, Tamanho do armazenamento, Memória RAM, Número de processadores, Função e Tipo de placa de rede.

Tabela 1 Estrutura das VM

<i>DC01</i>	<i>Windows Server 2012 R2</i>	60 GB	2 GB	2	<i>Domain Controller</i>	1 <i>Host-only</i>
<i>ROUTING</i>	<i>Windows Server 2012 R2</i>	60 GB	2 GB	2	<i>Routing Server</i>	1 <i>Host-only</i> e 1 <i>NAT</i>
<i>SQL</i>	<i>Windows Server 2012 R2</i>	60 GB	2 GB	2	<i>SQL Server</i>	1 <i>Host-only</i>
<i>CLIENT</i>	<i>Windows 8</i>	80 GB	4 GB	2	<i>Client PC</i>	1 <i>Host-only</i>

Fonte do autor

Configuração do Domain Controller (DC01)

Na rede virtual é necessário criar o domínio “ISTEC.LOCAL”, que vai unir todas as máquinas virtuais.

1. Iniciar o *VMWare Workstation* (como administrador).
2. Iniciar a máquina virtual “DC01”, clicar com o botão direito do rato e escolher a opção “*Power Start Up Guest*”. Fazer a instalação de modo padrão do *Windows Server 2012 R2*, definir o teclado em Português (Portugal) e usar como palavra-passe (P@ssw0rd) para o administrador local.
3. Após efetuar a instalação do Sistema Operativo, alterar as seguintes informações:
 - a) Nome da Máquina: “DC01”;
 - b) Configuração de Ipv4: 192.168.0.1;
 - c) Ativar Serviços na Firewall: “*File and Printer Sharing*”.
4. Executar o *Windows PowerShell* (como administrador) e inserir os seguintes comandos:
 - a) Add-WindowsFeature AD-Domain-Services -IncludeManagementTools;
 - b) Install-ADDSForest -DomainName “ISTEC.LOCAL”.
5. Definir como palavra-passe (P@ssw0rd) para o domínio “ISTEC.LOCAL” e introduzir a mesma quando o for pedida.
6. Por fim, escolher a opção “*Yes to All*” e aguardar que o processo de configuração termine e o serviço DNS fica instalado no decorrer deste processo.

Servidor Routing

Servidor que tem a função de comunicar com o exterior. Utilizando o serviço “*Routing and Remote Access Service*” (RRAS) permite, que todos os servidores efetuem a ligação ao exterior.

1. Iniciar o *VMWare Workstation* (como administrador).
2. Iniciar a máquina virtual “*ROUTING*”, clicar com o botão direito do rato e escolher a opção “*Power Start Up Guest*”. Fazer a instalação de modo padrão do *Windows Server 2012 R2*, definir o teclado em Português (Portugal) e usar como palavra-passe (P@ssw0rd) para o administrador local.
3. Após efetuar a instalação do Sistema Operativo, alterar as seguintes informações:
 - a) Nome da Máquina: “*ROUTING*”;
 - b) Configuração de Ipv4: 192.168.0.2;
 - c) Ativar Serviços na Firewall: “*File and Printer Sharing*”.
4. Reiniciar a máquina virtual para aplicar as configurações.
5. Efetuar a adesão da máquina virtual ao domínio seguindo os passos: Propriedades do sistema -> Aba Nome do Computador-> Alterações de Nome do Computador/Domínio -> Domínio -> escrever “*ISTEC.LOCAL*” -> OK -> Reiniciar.
6. Quando a máquina virtual reiniciar, no ecrã de início de sessão introduzir o nome de utilizador (ISTEC\administrator) e a palavra-passe (P@ssw0rd).
7. Aguardar que a janela *Server Manager* surja, aceder a *Manage* e escolher a opção “*Add Roles and Features*”.
8. Na página *Before You Begin*, selecionar a opção “*Role-based or feature-based installation*” -> *Server Roles*, selecionar a opção “*Remote Access*” -> *Features* -> *Remote Access*, selecionar a opção “*Routing*” -> *Web Server Role (IIS)* -> *Role Services* -> *Confirmation* -> *Install*.
9. Aguardar que a *feature* seja instalada.
10. Abrir a janela *Server Manager*, aceder a *Tools*, e escolher a opção “*Routing and Remote Access*”.

11. Na consola de “*Routing and Remote Access*”, clicar com o botão direito do rato no servidor (*ROUTING*), e escolher a opção “*Configure and Enable Routing and Remote Access*” -> *Network Address Translation (NAT)* -> Selecionar a placa de rede com o nome (NAT) -> *Finish*.
12. Aguardar que o processo de configuração termine.

Servidor SQL

A função do servidor *SQL* é de gerir e guardar dados na base de dados relacional que o programa *WorkMatic* usa.

1. Iniciar o *VMWare Workstation* (como administrador).
2. Iniciar a máquina virtual “*SQL*”, clicar com o botão direito do rato e escolher a opção “*Power Start Up Guest*”. Fazer a instalação de modo padrão do *Windows Server 2012 R2*, definir o teclado em Português (Portugal) e usar como palavra-passe (P@ssw0rd) para o administrador local.
3. Após efetuar, a instalação do Sistema Operativo, alterar as seguintes informações:
 - a) Nome da Máquina: “*SQL*”;
 - b) Configuração de Ipv4: 192.168.0.3;
 - c) Ativar Serviços na Firewall: “*File and Printer Sharing*”.
4. Reiniciar a máquina virtual para aplicar as configurações.
5. Efetuar a adesão da máquina virtual ao domínio, seguindo os passos: Propriedades do sistema -> Aba Nome do Computador-> Alterações de Nome do Computador/Domínio -> Domínio -> escrever “*ISTEC.LOCAL*” -> OK -> Reiniciar.
6. Quando a máquina virtual reiniciar, no ecrã de início de sessão introduzir o nome de utilizador (*ISTEC\administrator*) e a palavra-passe (P@ssw0rd).
7. Executar o *Windows Update* e fazer as atualizações necessárias.
8. Instalar o *SQL Server 2017. Installation*, selecionar a opção “*New SQL Server stand-alone installation or add features to an existing installation*” -> Aceitar os termos e condições -> *Product Updates* -> *Feature Selection* -> *Instance Configuration*, escrever “*SQL*” na opção “*Named instance*” -> *Server*

Configuration -> Database Engine Configuration, seleccionar a opção “*Mixed Mode*” definir como palavra-passe (P@ssw0rd) -> *Complete*.

9. Reiniciar a máquina virtual para aplicar as configurações.

Com o *SQL Server 2017* instalado, vamos ao servidor *DC01*. Na janela *Server Manager* aceder a *Tools* e escolher a opção “*Active Directory Users and Computers*”, para criar um Grupo de Segurança chamado “*APPUsers*”, que vai ser membro do Grupo de Segurança “*Domain Users*”. Para um utilizador ter acesso ao programa *WorkMatic* tem de ser membro do Grupo de Segurança “*APPUsers*”.

Iniciar o *SQL Server 2017 Management Studio* no servidor *SQL*, entrar com a conta de Administrador do domínio, clicar com o botão direito do rato, na opção “*Login*” que está na pasta “*Security*” e seleccionar a opção “*New Login...*”. Na janela “*Login - New*” clicar no botão “*Search*” que vai abrir a janela “*Select User or Group*” na caixa de texto escrever “*APPUsers*” clicar OK.

Depois de fazer a configuração de segurança, vamos criar a base de dados usando o ficheiro “*WorkMatic.sql*” que tem o código necessário.

Máquina virtual CLIENT

1. Iniciar o *VMWare Workstation* (como administrador).
2. Iniciar a máquina virtual “*CLIENT*”, clicar com botão direito do rato e escolher a opção “*Power Start Up Guest*”. Fazer a instalação de modo padrão do *Windows 8*, definir o teclado em Português (Portugal) e usar como palavra-passe (P@ssw0rd) para o administrador local.
3. Após efetuar, a instalação do Sistema Operativo, alterar as seguintes informações:
 - a) Nome da Máquina: “*CLIENT*”;
 - b) Configuração de Ipv4: 192.168.0.4;
 - c) Ativar Serviços na Firewall: “*File and Printer Sharing*”.
4. Reiniciar a máquina virtual para aplicar as configurações.

5. Efetuar a adesão da máquina virtual ao domínio, seguindo os passos: Propriedades do sistema -> Aba Nome do Computador-> Alterações de Nome do Computador/Domínio -> Domínio -> escrever "ISTEC.LOCAL" -> OK -> Reiniciar.
6. Quando a máquina virtual reiniciar, no ecrã de início de sessão introduzir o nome de utilizador (ISTEC\administrator) e a palavra-passe (P@ssw0rd).
7. Instalar o *Visual Studio 2017*. Aceitar os termos e condições -> *Installing*, seleccionar as opções ".net desktop development" e "Data storage and processing" -> *Products* -> *Finish*.

Com todos as máquinas virtuais corretamente criadas e configuradas é só fazer "*Snapshot*" em cada uma delas, para garantir o *Backup*.

Capítulo IV

Desenvolvimento de *WorkMatic*

O programa *WorkMatic* é desenvolvido no âmbito do projeto em WPF/C#. Todo o código fonte está disponível, no fim do projeto em Anexos.

O *WorkMatic* tem como objetivo facilitar a gestão de clientes e encomendas para pequenas empresas de distribuição de garrafas de gás. Foi projetado para funcionar numa rede virtual e num domínio Microsoft, que foram criados para este projeto. Aproveitando, assim, as vantagens de segurança da *Active Directory* na autenticação dos utilizadores no *WorkMatic*. Usa uma base de dados SQL para gerir e armazenar os dados criados pelo utilizador e, ainda, permite a exportação de dados para documentos PDF.

O *WorkMatic* tem funcionalidades diferentes, dependendo do utilizador autenticado. Existem 3 tipos de utilizadores no *WorkMatic*: o administrador, o utilizador e o não utilizador.

Administrador:

- Adicionar, corrigir e apagar os produtos (Tipo e preço);
- Adicionar, corrigir ou apagar as fichas de clientes;
- Exportar as fichas de cliente para documentos PDF;
- Adicionar e corrigir as encomendas dos clientes;
- Exportar os recibos de encomendas para documentos PDF.

Utilizador:

- Adicionar, corrigir e apagar as fichas de clientes;
- Exportar as fichas de cliente para documentos PDF;
- Adicionar e corrigir as encomendas dos clientes;
- Exportar os recibos de encomendas para documentos PDF.

Não utilizador:

- Não tem acesso a nenhuma funcionalidade do *WorkMatic*

Ficheiros "MainWindow.xaml" e "MainWindow.xaml.cs"

Os ficheiros "MainWindow.xaml" e "MainWindow.xaml.cs" são responsáveis pela janela principal do programa.

O ficheiro "MainWindow.xaml" contém o código que determina a interface gráfica da janela principal, que tem um "TabControl" com três "TabItem". Em cada um são configurados o nome e o fundo.

O ficheiro "MainWindow.xaml.cs" contém o código por detrás da interface gráfica responsável pela autenticação dos utilizadores e da interação com os "TabItem".

Ficheiros "FormWindows.xaml" e "FormWindows.xaml.cs"

Os ficheiros "FormWindows.xaml" e "FormWindows.xaml.cs" são responsáveis pelo "TabItem" com o nome "formHolder".

O ficheiro "FormWindows.xaml" contém o código que determina a interface gráfica do "TabItem", que tem duas "Grid" e uma "DataGrid". Na "DataGrid", com o nome "gridfMatic", que pertence à primeira "Grid" são criados os elementos que vão apresentar graficamente alguns campos da tabela "CLIENTS". Na segunda "Grid" são criados os elementos que vamos usar para interagir com a tabela "CLIENTS".

O ficheiro "FormWindows.xaml.cs" contém o código por detrás da interface gráfica responsável pela conexão com a tabela "CLIENTS" e da interação com todos os elementos da "TabItem".

Ficheiros "OrderWindows.xaml" e "OrderWindows.xaml.cs"

Os ficheiros "OrderWindows.xaml" e "OrderWindows.xaml.cs" são responsáveis pelo "TabItem" com o nome "orderHolder".

O ficheiro "OrderWindows.xaml" contém o código que determina a interface gráfica do "TabItem", que tem duas "Grid" e duas "DataGrid". Na "DataGrid", com o nome

“gridoMatic”, que pertence à primeira “Grid” são criados os elementos que vão apresentar graficamente alguns campos da tabela “CLIENTS”. Na “DataGrid”, com o nome “gridoMatic1”, que também pertence à primeira “Grid” são criados os elementos que vão apresentar graficamente alguns campos da tabela “ORDERS”. Na segunda “Grid” são criados os elementos que vamos usar para interagir com as tabelas “ORDERS”, “CLIENTS” e “PRODUCTS”.

O ficheiro “OrderWindows.xaml.cs” contém o código por detrás da interface gráfica responsável pela conexão com as tabelas “ORDERS”, “CLIENTS” e “PRODUCTS”, e da interação com todos os elementos da “TabItem”

Ficheiros “ProductWindow.xaml” e “ProductWindow.xaml.cs”

Os ficheiros “ProductWindow.xaml” e “ProductWindow.xaml.cs” são responsáveis pelo “TabItem” com o nome “productHolder”.

O ficheiro “ProductWindow.xaml” contém o código que determina a interface gráfica do “TabItem”, que tem duas “Grid” e uma “DataGrid”. Na “DataGrid”, com o nome “gridpMatic”, que pertence à primeira “Grid” são criados os elementos que vão apresentar graficamente os campos da tabela “PRODUCTS”. Na segunda “Grid” são criados os elementos que vamos usar para interagir com a tabela “PRODUCTS”.

O ficheiro “ProductWindow.xaml.cs” contém o código por detrás da interface gráfica responsável pela conexão com a tabela “PRODUCTS”, e da interação com todos os elementos da “TabItem”

Ficheiro “WorkMatic.edmx”

O ficheiro “WorkMatic.edmx” contém o código responsável pela conexão do programa com a base de dados. Este ficheiro é criado durante o processo de ativação da ferramenta (ADO.NET Entity Framework).

Classe "ExportPdf.cs"

A classe "ExportPdf.cs" é implementada quando um utilizador pressiona o botão "Export". Esta verifica, o conteúdo dos campos predefinidos e insere-os no documento PDF, pela ordem seleccionada.

Classe "MoneyConverter.cs"

A classe "MoneyConverter.cs" converte os valores "decimal" em valores "string", arredonda o valor à segunda casa decimal depois da vírgula e junta o símbolo "€".

Conclusão

Com a conclusão do projeto foi possível comprovar as vantagens da virtualização, ao instalar quatro sistemas operativos (três *Windows Server 2012 R2* e um *Windows 8.1*) num computador. Permitindo ainda reduzir o nível do *hardware* e do espaço físico necessário para a infraestrutura.

A utilização do SQL comprovou as suas vantagens, ao facilitar a criação da base de dados, a manipulação e o acesso aos dados existentes na base de dados.

O domínio Microsoft foi uma parte importante deste projeto por permitir a interligação e comunicação segura entre as máquinas virtuais existentes.

Este projeto foi desenvolvido num contexto académico usando todas as tecnologias dessa forma. Conclui-se que é possível aplicar estas tecnologias no uso empresarial, usando a estrutura existente neste projeto. É ainda possível, futuramente, desenvolver módulos que acrescentem novas funcionalidades ao programa.

Referências Bibliográficas

- Brandão, P. R. (24 de fevereiro de 2018). *PEDRO-BRANDÃO.pdf*. Obtido de <http://www.kriativ-tech.com:830.pdf>
<http://www.kriativ-tech.com/wp-content/uploads/2018/02/PEDRO-BRAND%C3%83O.pdf>
- Damas, L. (1999). SQL - Structured Query Language. Em L. Damas, *SQL - Structured Query Language* (pp. 2,3,4). Lisboa: FCA - Editora de Informática.
- Ferreira, A. M. (2015). Introdução ao Cloud Computing - IaaS, PaaS, Tecnologia, Conceito e Modelo de Negócios. Em A. M. Ferreira, *Introdução ao Cloud Computing* (pp. 15-22, 81, 82). Lisboa: FCA - Editora de Informática.
- Mell, P., & Grance, T. (01 de setembro de 2011). The NIST Definition of Cloud Computing. *Recommendations of the National Institute of Standards and Technology*, p. 2.
- Microsoft Corporation. (20 de julho de 2015). <https://docs.microsoft.com>. Obtido de [introduction-to-linq: https://docs.microsoft.com/pt-br/dotnet/csharp/programming-guide/concepts/linq/introduction-to-linq](https://docs.microsoft.com/pt-br/dotnet/csharp/programming-guide/concepts/linq/introduction-to-linq)
- Microsoft Corporation. (10 de abril de 2018). <https://docs.microsoft.com>. Obtido de [framework: https://docs.microsoft.com/pt-br/dotnet/framework/get-started/](https://docs.microsoft.com/pt-br/dotnet/framework/get-started/)
- Microsoft Corporation. (26 de janeiro de 2018). <https://docs.microsoft.com>. Obtido de [wpf: https://docs.microsoft.com/pt-br/dotnet/framework/wpf/getting-started/](https://docs.microsoft.com/pt-br/dotnet/framework/wpf/getting-started/)
- Microsoft Corporation. (26 de outubro de 2018). <https://docs.microsoft.com>. Obtido de [visual-studio-ide?view=vs-2017: https://docs.microsoft.com/pt-br/visualstudio/ide/visual-studio-ide?view=vs-2017](https://docs.microsoft.com/pt-br/visualstudio/ide/visual-studio-ide?view=vs-2017)
- Red Hat, Inc. (01 de 01 de 2019). <https://www.redhat.com/pt-br>. Obtido de [what-is-virtualization: https://www.redhat.com/pt-br/topics/virtualization/what-is-virtualization](https://www.redhat.com/pt-br/topics/virtualization/what-is-virtualization)
- Rosa, A. (2013). Windows Server 2012 - Curso Completo. Em A. Rosa, *Windows Server 2012 - Curso Completo* (pp. 77,78). Lisboa: FCA - Editora de Informática, Lda.
- Stanek, W. R. (2014). Windows Server 2012 - Guia de Bolso. Em W. R. Stanek, *Windows Server 2012 - Guia de Bolso* (pp. 3,15,16). Bookman; 1 edition.
- Tanamachi, V. R., de Oliveira, W. P., & Galvani, A. (1 de outubro de 2013). ANÁLISE DE DESEMPENHO ENTRE SOFTWARES. *Revista Eletrônica e-F@tec*, pp. 2,3. Obtido de http://www.fatecgarca.edu.br:8080/revista/Volume3/artigos_vol3/Artigo_12.pdf
- VMware, Inc. (01 de janeiro de 2018). *workstation-pro.html*. Obtido de www.vmware.com:8080/products/workstation-pro.html
<https://www.vmware.com/products/workstation-pro.html>.

Anexos

Anexo I - Código do ficheiro WorkMatic.sql

```
CRIAÇÃO DA BASE DE DADOS  
CREATE DATABASE WorkMatic;
```

```
SELEÇÃO DA BASE DE DADOS  
USE WorkMatic;
```

```
CRIAÇÃO DA TABELA USER  
CREATE TABLE USERS(  
ID_USER INT IDENTITY (1,1) PRIMARY KEY,  
NAME_USER VARCHAR(255) NOT NULL);
```

```
CRIAÇÃO DA TABELA CLIENT  
CREATE TABLE CLIENTS(  
ID_CLIENT INT IDENTITY (1,1) PRIMARY KEY,  
NAME_CLIENT VARCHAR(255) NOT NULL,  
ADDRESS_CLIENT VARCHAR (MAX) NOT NULL,  
COD_POST1 INT CHECK (COD_POST1 BETWEEN 1000 AND 9980),  
COD_POST2 INT CHECK (COD_POST2 BETWEEN 000 AND 999),  
LOCALITY VARCHAR (50),  
NIF INT CHECK (NIF BETWEEN 100000000 AND 999999999) NOT NULL,  
PHONE_1 INT CHECK (PHONE_1 BETWEEN 900000000 AND 999999999),  
PHONE_2 INT CHECK (PHONE_2 BETWEEN 200000000 AND 299999999),  
E_MAIL VARCHAR (50),  
IMAGE_CLIENT VARCHAR (255),  
DATE_CLIENT DATE NOT NULL DEFAULT (GETDATE()),  
CHECK_1 VARCHAR (3) NOT NULL CHECK (CHECK_1 IN('SIM', 'NÃO')));
```

```
CRIAÇÃO DA TABELA ADMIN  
CREATE TABLE ADMINS(  
ID_ADMIN INT IDENTITY (1,1) PRIMARY KEY,  
ID_USER INT FOREIGN KEY REFERENCES USERS(ID_USER),  
CAN_READ_USER BIT NOT NULL,  
FULL_ACCESS BIT NOT NULL);
```

```
CRIAÇÃO DA TABELA PRODUCTS  
CREATE TABLE PRODUCTS(  
ID_PRODUCT INT IDENTITY (1,1) PRIMARY KEY,  
TYPE_PRODUCT VARCHAR (MAX) NOT NULL,  
PRICE_PRODUCT SMALLMONEY NOT NULL);
```

```
CRIAÇÃO DA TABELA ORDERS  
CREATE TABLE ORDERS(  
ID_ORDERS INT IDENTITY (1,1) PRIMARY KEY,  
ID_USER INT FOREIGN KEY REFERENCES USERS(ID_USER),  
ID_CLIENT INT FOREIGN KEY REFERENCES CLIENTS(ID_CLIENT),  
ID_PRODUCTS INT FOREIGN KEY REFERENCES PRODUCTS(ID_PRODUCT),  
QUANTITY TINYINT NOT NULL,  
DATE_ORDERS DATE NOT NULL,  
TIME_ORDERS TIME NOT NULL,  
TOTAL_PRICE SMALLMONEY);
```

Anexo II - Código do ficheiro ClientId.sql

Função que faz um inner join entre a tabla Orders e a tabela Products e apresenta o resultado a variável @ClientId.

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE FUNCTION [dbo].[CLIENT_ORDERS] (@ClientId int)
RETURNS TABLE
AS
RETURN
    SELECT * FROM dbo.ORDERS INNER JOIN dbo.PRODUCTS ON dbo.ORDERS.ID_PRODUCTS =
    dbo.PRODUCTS.ID_PRODUCT
    WHERE dbo.ORDERS.ID_CLIENT = @ClientId;
GO
```

Anexo III - Código do ficheiro MainWindow.xaml

Código do grafico da janela principal.

```
<Window x:Class="WorkMatic.MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:local="clr-namespace:WorkMatic"
    mc:Ignorable="d"
    Title="WorkMatic">
    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition Height="1*" />
        </Grid.RowDefinitions>
        <Grid.ColumnDefinitions>
            <ColumnDefinition />
        </Grid.ColumnDefinitions>
        <TabControl x:Name="tabControl" Grid.Column="0" Grid.Row="0"
            Grid.ColumnSpan="2" SelectionChanged="TabControl_SelectionChanged">
            <TabItem Header="Encomendas">
                <local:OrderWindow Background="#FFE5E5E5" x:Name="orderHolder" />
            </TabItem>
            <TabItem Header="Formulario">
                <local:FormWindow Background="#FFE5E5E5" x:Name="formHolder" />
            </TabItem>
            <TabItem Header="Produtos" x:Name="ProductTab">
                <local:ProductWindow Background="#FFE5E5E5" x:Name="productHolder" />
            </TabItem>
        </TabControl>
    </Grid>
</Window>
```

Anexo IV - Código do ficheiro MainWindow.xaml.cs

Código funcional da janela principal.

```
using System;
```



```

using System.DirectoryServices;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;
using System.DirectoryServices.AccountManagement;
using System.Threading;

namespace WorkMatic
{
    // Criação da classe "MainWindow" está é a classe responsável pela criação dos
    elementos gráficos.
    public partial class MainWindow : Window
    {
        GroupPrincipal appGroup;
        GroupPrincipal adminGroup;

        public MainWindow()
        {
            InitializeComponent();

            // Código responsável pela conexão com a active directory.
            AppDomain.CurrentDomain.SetPrincipalPolicy(System.Security.Principal.PrincipalPolicy.W
indowsPrincipal);
            var ctx = new PrincipalContext(ContextType.Domain);

            appGroup = GroupPrincipal.FindByIdentity(ctx, "APPUsers");
            adminGroup = GroupPrincipal.FindByIdentity(ctx, "Domain Admins");

            if (!UserCanUseApp())
            {
                MessageBox.Show("Não tem permissão para abrir a aplicação!");
                Application.Current.Shutdown();
            }

            productHolder.IsEnabled = IsUserAdmin();
            productHolder.Visibility = (IsUserAdmin() ? Visibility.Visible :
Visibility.Hidden);
            ProductTab.IsEnabled = IsUserAdmin();
            ProductTab.Visibility = (IsUserAdmin() ? Visibility.Visible :
Visibility.Hidden);
        }

        // Método atualiza os dados em cada separador
        private void TabControl_SelectionChanged(object sender,
SelectionChangedEventArgs e)
        {
            if(e.Source is TabControl)
            {
                if(e.AddedItems.Count > 0)

```

```

        {
            var tab = (TabItem)e.AddedItems[0];
            if(tab.Content is OrderWindow)
            {
                ((OrderWindow)tab.Content).loadoMatic();
            } else if(tab.Content is FormWindow)
            {
                ((FormWindow)tab.Content).loadfMatic();
            }
            else if (tab.Content is ProductWindow && IsUserAdmin())
            {
                ((ProductWindow)tab.Content).loadpMatic();
            }
        }
    }
}
// Métodos para validar administradores
public bool IsUserAdmin()
{
    var user = UserPrincipal.Current;
    return user.IsMemberOf(adminGroup);
}
// Métodos para validar utilizadores
public bool UserCanUseApp()
{
    var user = UserPrincipal.Current;
    return user.IsMemberOf(appGroup);
}
}
}

```

Anexo V - Código do ficheiro OrderWindow.xaml

Código do grafico do "TabItem" "orderHolder".

```

<Grid x:Class="WorkMatic.OrderWindow"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
      xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
      xmlns:wpfTool="clr-namespace:Xceed.Wpf.Toolkit;assembly=Xceed.Wpf.Toolkit"
      xmlns:local="clr-namespace:WorkMatic"
      mc:Ignorable="d">
    <Grid.RowDefinitions>
        <RowDefinition />
        <RowDefinition />
    </Grid.RowDefinitions>
    <Grid.ColumnDefinitions>
        <ColumnDefinition Width="auto"></ColumnDefinition>
        <ColumnDefinition Width="3*"></ColumnDefinition>
    </Grid.ColumnDefinitions>
    <DataGrid Name="gridoMatic1" IsReadOnly="true" AutoGenerateColumns="False"
              Grid.Column="0" Grid.Row="0" SelectionChanged="GridoMatic1_SelectionChanged"
              CanUserResizeRows="False" CanUserResizeColumns="False" CanUserReorderColumns="False"
              FontFamily="Verdana" Background="Gray">
        <DataGrid.Columns>
            <DataGridTextColumn Header="Nome" Binding="{Binding NAME_CLIENT}"
                                MinWidth="255" MaxWidth="255"/>

```

```
        <DataGridTextColumn Header="NIF" Binding="{Binding NIF}" MinWidth="100"
MaxWidth="100"/>
        <DataGridTextColumn Header="Morada" Binding="{Binding ADDRESS_CLIENT}"
MinWidth="325" MaxWidth="8000"/>
    </DataGrid.Columns>
</DataGrid>

    <DataGrid Name="gridoMatic2" IsReadOnly="true" AutoGenerateColumns="False"
Grid.Column="0" SelectionMode="Single" Grid.Row="1"
SelectionChanged="GridoMatic2_SelectionChanged" Grid.IsSharedSizeScope="True"
CanUserResizeRows="False" CanUserResizeColumns="False" CanUserReorderColumns="False"
FontFamily="Verdana" Background="Gray">
    <DataGrid.Resources>
        <local:MoneyConverter x:Key="MoneyConverter" />
    </DataGrid.Resources>
    <DataGrid.Columns>
        <DataGridTextColumn Header="Produto" Binding="{Binding TYPE_PRODUCT}"
MinWidth="200" MaxWidth="8000"/>
        <DataGridTextColumn Header="Preço" Binding="{Binding PRICE_PRODUCT,
Converter={StaticResource MoneyConverter}}" MinWidth="100" MaxWidth="100"/>
        <DataGridTextColumn Header="Quantidade" Binding="{Binding QUANTITY}"
MinWidth="80" MaxWidth="80"/>
        <DataGridTextColumn Header="Dia" Binding="{Binding DATE_ORDERS}"
MinWidth="100" MaxWidth="100"/>
        <DataGridTextColumn Header="Hora" Binding="{Binding TIME_ORDERS}"
MinWidth="100" MaxWidth="100"/>
        <DataGridTextColumn Header="Preço Total" Binding="{Binding TOTAL_PRICE,
Converter={StaticResource MoneyConverter}}" MinWidth="100" MaxWidth="100" />
    </DataGrid.Columns>
</DataGrid>

    <Grid Grid.Column="1" Grid.RowSpan="3" Background="{DynamicResource {x:Static
SystemColors.WindowFrameBrushKey}}">

        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="auto"></ColumnDefinition>
            <ColumnDefinition Width="3*"></ColumnDefinition>
        </Grid.ColumnDefinitions>
        <Grid.RowDefinitions>
            <RowDefinition Height="*"></RowDefinition>
            <RowDefinition Height="*"></RowDefinition>
            <RowDefinition Height="*"></RowDefinition>
            <RowDefinition Height="*"></RowDefinition>
            <RowDefinition Height="*"></RowDefinition>
            <RowDefinition Height="*"></RowDefinition>
            <RowDefinition Height="*"></RowDefinition>
            <RowDefinition Height="*"></RowDefinition>
            <RowDefinition Height="*"></RowDefinition>
            <RowDefinition Height="*"></RowDefinition>
            <RowDefinition Height="*"></RowDefinition>
            <RowDefinition Height="*"></RowDefinition>
            <RowDefinition Height="*"></RowDefinition>
            <RowDefinition Height="*"></RowDefinition>
            <RowDefinition Height="*"></RowDefinition>
        </Grid.RowDefinitions>
        <Grid.Resources>
            <Style TargetType="TextBox">
                <Setter Property="Margin" Value="10,5,20,5"/>
                <Setter Property="FontFamily" Value="Verdana"/>
            </Style>
            <Style TargetType="Label">

```

```

                <Setter Property="HorizontalAlignment" Value="Center"/>
                <Setter Property="Margin" Value="5,0"/>
                <Setter Property="FontWeight" Value="Black"/>
                <Setter Property="FontFamily" Value="Verdana"/>
                <Setter Property="Foreground" Value="WhiteSmoke" />
            </Style>
        </Grid.Resources>
        <Label Content="Nome" Grid.Row="0" Grid.Column="0"></Label>
        <TextBox x:Name="txtNAME" Grid.Row="0" Grid.Column="1"
IsEnabled="False"></TextBox>
        <Label Content="Morada" Grid.Row="1" Grid.Column="0" ></Label>
        <TextBox x:Name="txtADDRESS" Grid.Row="1" Grid.Column="1"
IsEnabled="False"></TextBox>
        <Label Content="NIF" Grid.Row="2" Grid.Column="0" ></Label>
        <TextBox x:Name="txtNIF" Grid.Row="2" Grid.Column="1"
IsEnabled="False"></TextBox>
        <Label Content="Quantidade" Grid.Row="3" Grid.Column="0" ></Label>
        <TextBox x:Name="txtQUANTITY" MaxLength="2" PreviewTextInput="NumericOnly"
Grid.Row="3" Grid.Column="1" TextChanged="TxtQUANTITY_TextChanged" ></TextBox>
        <Label Content="Dia" Grid.Row="4" Grid.Column="0" ></Label>
        <DatePicker x:Name="txtDAY" Grid.Row="4" Grid.Column="1"
Margin="10,0,20,0"></DatePicker>
        <Label Content="Hora" Grid.Row="5" Grid.Column="0" ></Label>
        <wpfTool:TimePicker x:Name="txtHOURL" Grid.Row="5" Grid.Column="1"
Margin="10,0,20,0"></wpfTool:TimePicker>
        <Label Content="Produto" Grid.Row="6" Grid.Column="0" ></Label>
        <ComboBox x:Name="txtPRODUCT" Grid.Row="6" Grid.Column="1" Margin="10,0,20,0"
IsSynchronizedWithCurrentItem="True" SelectionChanged="TxtPRODUCT_SelectionChanged"/>
        <Label Content="Preço" Grid.Row="7" Grid.Column="0" ></Label>
        <TextBox x:Name="txtPRICE" Grid.Row="7" Grid.Column="1" IsEnabled="False"/>
        <Label Content="Preço Total" Grid.Row="8" Grid.Column="0" ></Label>
        <TextBox x:Name="txtTOTAL" Grid.Row="8" Grid.Column="1"
IsEnabled="False"></TextBox>
        <Button Name="btoInsert" Content="Insert" Grid.Row="11" Grid.ColumnSpan="2"
Margin="50,5" Click="btInsert_Click" />
        <Button Name="btoUpdate" Content="Update" Grid.Row="12" Grid.ColumnSpan="2"
Margin="50,5" Click="btUpdate_Click" />
        <Button Name="btoExport" Content="Export" Grid.Row="13" Grid.ColumnSpan="2"
Margin="50,5" Click="btExport_Click" />
    </Grid>
</Grid>

```

Anexo VI - Código do ficheiro OrderWindow.xaml.cs

Código funcional do "TabItem" "orderHolder".

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Linq;
using System.Runtime.Remoting.Contexts;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;

```

```

using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;

namespace WorkMatic
{
    // Criação da classe "OrderWindow" está é a classe responsável pelo funcionamento
    de todos os elementos gráficos da aplicação no menu "Compras".
    public partial class OrderWindow : Grid
    {
        ORDER currentSelectedOrder;
        CLIENT currentSelectedClient;

        // Inicialização dos metodos "InitializeComponent()" e "loadoMatic()" quando a
        classe "OrderWindow" inicia.
        public OrderWindow()
        {
            InitializeComponent();
            if (!DesignerProperties.GetIsInDesignMode(this))
            {
                loadoMatic();
            }
        }

        // Responsável por aceder a base de dados, ler os valores e apresentar ao
        programa. Usa "Entities" para realizar esta função.
        public void loadoMatic()
        {
            using (WorkMaticEntities db = new WorkMaticEntities())
            {
                var qomatic1 = (from c in db.CLIENTS select c).ToList();
                gridoMatic1.ItemsSource = qomatic1;

                var qomatic3 = (from c in db.PRODUCTS select c).ToList();

                txtPRODUCT.DisplayMemberPath = "TYPE_PRODUCT";
                txtPRODUCT.ItemsSource = qomatic3;
            }
        }

        private void btInsert_Click(object sender, RoutedEventArgs e)
        {
            // Lê os valores que estão nas "txtBox" e cria um atributo com esse valor
            using (WorkMaticEntities db = new WorkMaticEntities())
            {
                if (currentSelectedClient != null &&
                    !String.IsNullOrEmpty(txtQUANTITY.Text) && txtDAY.SelectedDate != null &&
                    txtHOUR.Value != null && txtPRODUCT.SelectedItem != null &&
                    !String.IsNullOrEmpty(txtTOTAL.Text))
                {
                    // Lê os valores que estão nas "txtBox" e cria um objeto e os seus
                    atributos com os valor das textboxes
                    ORDER allOrder = new ORDER

```

```

        {
            ID_CLIENT = currentSelectedClient.ID_CLIENT,
            ID_PRODUCTS = ((PRODUCT)txtPRODUCT.SelectedItem).ID_PRODUCT,
            QUANTITY = byte.Parse(txtQUANTITY.Text),
            DATE_ORDERS = (DateTime)txtDAY.SelectedDate,
            TIME_ORDERS = new TimeSpan(txtHOUR.Value.Value.Hour,
txtHOUR.Value.Value.Minute, txtHOUR.Value.Value.Second),
            TOTAL_PRICE = decimal.Parse(txtQUANTITY.Text) *
decimal.Parse(txtPRICE.Text)
        };
        db.ORDERS.Add(allOrder);
        db.SaveChanges();
        gridoMatic2.ItemsSource =
db.CLIENT_ORDERS(currentSelectedClient.ID_CLIENT);
    }
    else
    {
        MessageBox.Show("Preencha todos os dados", "Cuidado",
MessageBoxButton.OK, MessageBoxImage.Information);
    }
}

//Botão "update" e usado para o utilizador poder corrigir ou atualizar os
valores que já estão preenchidos na base de dados.
private void btUpdate_Click(object sender, RoutedEventArgs e)
{
    if (currentSelectedClient != null &&
!String.IsNullOrEmpty(txtQUANTITY.Text) && txtDAY.SelectedDate != null &&
txtHOUR.Value != null && txtPRODUCT.SelectedItem != null)
    {
        using (WorkMaticEntities db = new WorkMaticEntities())
        {
            int ID_ORDERS = currentSelectedOrder.ID_ORDERS;
            ORDER orderToUpdate = db.ORDERS.Where(x => x.ID_ORDERS ==
ID_ORDERS).FirstOrDefault();

            orderToUpdate.QUANTITY = byte.Parse(txtQUANTITY.Text);
            orderToUpdate.DATE_ORDERS = (DateTime)txtDAY.SelectedDate;
            orderToUpdate.TIME_ORDERS = new TimeSpan(txtHOUR.Value.Value.Hour,
txtHOUR.Value.Value.Minute, txtHOUR.Value.Value.Second);
            orderToUpdate.TOTAL_PRICE = decimal.Parse(txtQUANTITY.Text) *
decimal.Parse(txtPRICE.Text);

            db.SaveChanges();
            gridoMatic2.ItemsSource = db.CLIENTS.ToList();
            gridoMatic2.ItemsSource =
db.CLIENT_ORDERS(currentSelectedClient.ID_CLIENT);
        }
    }
    else
    {
        MessageBox.Show("Preencha todos os dados", "Cuidado",
MessageBoxButton.OK, MessageBoxImage.Information);
    }
}
}

```

//Botão "export" implementa a classe "ExportPdf", mas deixa o utilizador escolher a pasta de destino usando para isso a janela de pesquisa de ficheiros do Windows.

```
private void btExport_Click(object sender, RoutedEventArgs e)
{
    if (currentSelectedOrder != null)
    {
        using (WorkMaticEntities db = new WorkMaticEntities())
        {
            int ID_ORDER = currentSelectedOrder.ID_ORDERS;
            ORDER orderToExport = db.ORDERD.Where(x => x.ID_ORDERS ==
ID_ORDER).FirstOrDefault();

            Microsoft.Win32.SaveFileDialog dlg = new
Microsoft.Win32.SaveFileDialog();

            dlg.DefaultExt = ".pdf";
            dlg.Filter = "PDF Files (*.pdf)|*.pdf";
            // Apresenta "SaveFileDialog" usando o metodo "ShowDialog"
            Nullable<bool> result = dlg.ShowDialog();

            // Coloca o caminho mais o nome do ficheiro numa "txtBox" criada.
            if (result == true)
            {
                string filename = dlg.FileName;
                ExportPdf.ExportOrder(filename, orderToExport);
            }
        }
    }
    else
    {
        MessageBox.Show("Preencha todos os dados", "Cuidado",
MessageBoxButton.OK, MessageBoxImage.Information);
    }
}

//Serve para obter o cliente selecionado nas tabelas
private void GridoMatic1_SelectionChanged(object sender,
SelectionChangedEventArgs e)
{
    if (e.AddedItems.Count != 0)
    {
        using (WorkMaticEntities db = new WorkMaticEntities())
        {
            int ID_CLIENT = ((CLIENT)e.AddedItems[0]).ID_CLIENT;
            CLIENT clientToUpdate = db.CLIENTS.Where(x => x.ID_CLIENT ==
ID_CLIENT).FirstOrDefault();
            currentSelectedClient = clientToUpdate;

            txtNAME.Text = clientToUpdate.NAME_CLIENT;
            txtADDRESS.Text = clientToUpdate.ADDRESS_CLIENT;
            txtNIF.Text = clientToUpdate.NIF.ToString();

            gridoMatic2.ItemsSource = db.CLIENT_ORDERS(ID_CLIENT);
        }
    }
    else
    {
        currentSelectedClient = null;
    }
}
```

```

        txtNAME.Text = String.Empty;
        txtADDRESS.Text = String.Empty;
        txtNIF.Text = String.Empty;

        gridoMatic2.ItemsSource = null;
    }
}

//Serve para obter o cliente selecionado nas tabelas
private void GridoMatic2_SelectionChanged(object sender,
SelectionChangedEventArgs e)
{
    if (e.AddedItems.Count != 0)
    {
        using (WorkMaticEntities db = new WorkMaticEntities())
        {
            int ID_ORDER = ((CLIENT_ORDERS_Result)e.AddedItems[0]).ID_ORDERS;
            ORDER orderToUpdate = db.ORDERS.Where(x => x.ID_ORDERS ==
ID_ORDER).FirstOrDefault();
            currentSelectedOrder = orderToUpdate;

            txtQUANTITY.Text = orderToUpdate.QUANTITY.ToString();
            txtDAY.SelectedDate = orderToUpdate.DATE_ORDERS;
            txtHOUR.Value = new DateTime(orderToUpdate.TIME_ORDERS.Ticks);
            setTotalPrice((decimal) orderToUpdate.TOTAL_PRICE);

            txtPRODUCT.SelectedItem =
((List<PRODUCT>)txtPRODUCT.ItemsSource).FirstOrDefault(x => x.ID_PRODUCT ==
orderToUpdate.ID_PRODUCTS);
        }
    }
    else
    {
        currentSelectedOrder = null;
        txtQUANTITY.Text = String.Empty;
        txtDAY.SelectedDate = DateTime.Now;
        txtHOUR.Value = DateTime.Now;
        setTotalPrice(0);
    }
}
//Sempre que o utilizador muda na combobox o nome do produto o preço é
atualizado
private void TxtPRODUCT_SelectionChanged(object sender,
SelectionChangedEventArgs e)
{
    if(e.AddedItems.Count > 0)
    {
        setItemPrice(((PRODUCT)e.AddedItems[0]).PRICE_PRODUCT);

        if (!String.IsNullOrEmpty(txtQUANTITY.Text))
        {
            setTotalPrice(getItemPrice() * decimal.Parse(txtQUANTITY.Text));
        }
    }
    else
    {
        setItemPrice(0);
        setTotalPrice(0);
    }
}
}

```



```

// Não permite usar letras numa textbox
private void NumericOnly(object sender, TextCompositionEventArgs e)
{
    Regex regex = new Regex("[^0-9]+");
    e.Handled = regex.IsMatch(e.Text);
}

//Sempre que o utilizador escreve um valor na "txtQUANTITY" o preço total é
atualizado
private void TxtQUANTITY_TextChanged(object sender, TextChangedEventArgs e)
{
    if (!String.IsNullOrEmpty(txtQUANTITY.Text))
    {
        setTotalPrice(getItemPrice() * decimal.Parse(txtQUANTITY.Text));
    }
    else setTotalPrice(0);
}

// Método set para a textBox "txtTOTAL" que converter os valores decimal numa
string com duas casas decimais e o símbolo do "€"
private void setTotalPrice(decimal total)
{
    txtTOTAL.Text = Math.Round(total, 2).ToString() + "€";
}

// Método set para a textBox "txtPRICE" que converter os valores decimal numa
string com duas casas decimais e o símbolo do "€"
private void setItemPrice(decimal price)
{
    txtPRICE.Text = Math.Round(price, 2).ToString() + "€";
}

// Método get para a textBox "txtTOTAL" que converter os valores decimal numa
string com duas casas decimais e o símbolo do "€"
private decimal getTotalPrice()
{
    return decimal.Parse(txtTOTAL.Text.Replace("€", ""));
}

// Método get para a textBox "txtPRICE" que converter os valores decimal numa
string com duas casas decimais e o símbolo do "€"
private decimal getItemPrice()
{
    return decimal.Parse(txtPRICE.Text.Replace("€", ""));
}
}
}

```

Anexo VII - Código do ficheiro FormWindow.xaml

Código do grafico do "TabItem" "formHolder".

```

<Grid x:Class="WorkMatic.FormWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    mc:Ignorable="d">
    <Grid.ColumnDefinitions>
        <ColumnDefinition Width="auto" ></ColumnDefinition>

```

```

        <ColumnDefinition Width="3*" ></ColumnDefinition>
    </Grid.ColumnDefinitions>
    <DataGrid Name="gridfMatic1" IsReadOnly="true" AutoGenerateColumns="False"
Grid.Column="0" Grid.Row="0" SelectionChanged="GridsF_SelectionChanged"
SelectionMode="Single" Grid.IsSharedSizeScope="True" CanUserResizeRows="False"
CanUserResizeColumns="False" CanUserReorderColumns="False" CanUserSortColumns="False"
FontFamily="Verdana" Background="Gray">
    <DataGrid.Columns>
        <DataGridTextColumn Header="Nome" Binding="{Binding NAME_CLIENT}"
MinWidth="100" MaxWidth="255"/>
        <DataGridTextColumn Header="Morada" Binding="{Binding ADDRESS_CLIENT}"
MinWidth="150" MaxWidth="8000"/>
        <DataGridTextColumn Header="Localidade" Binding="{Binding LOCALITY}"
MinWidth="100" MaxWidth="100"/>
        <DataGridTextColumn Header="NIF" Binding="{Binding NIF}" MinWidth="100"
MaxWidth="100"/>
        <DataGridTextColumn Header="EMAIL" Binding="{Binding E_MAIL}"
MinWidth="100" MaxWidth="255"/>
        <DataGridTextColumn Header="Data" Binding="{Binding DATE_CLIENT}"
MinWidth="75" MaxWidth="75"/>
    </DataGrid.Columns>
</DataGrid>

    <Grid Grid.Column="1" Grid.RowSpan="2" Background="{DynamicResource {x:Static
SystemColors.WindowFrameBrushKey}}">
    <Grid.ColumnDefinitions>
        <ColumnDefinition Width="2*"></ColumnDefinition>
        <ColumnDefinition Width="5*"></ColumnDefinition>
    </Grid.ColumnDefinitions>
    <Grid.RowDefinitions>
        <RowDefinition Height="*"></RowDefinition>
        <RowDefinition Height="*"></RowDefinition>
        <RowDefinition Height="*"></RowDefinition>
        <RowDefinition Height="*"></RowDefinition>
        <RowDefinition Height="*"></RowDefinition>
        <RowDefinition Height="*"></RowDefinition>
        <RowDefinition Height="*"></RowDefinition>
        <RowDefinition Height="*"></RowDefinition>
        <RowDefinition Height="*"></RowDefinition>
        <RowDefinition Height="*"></RowDefinition>
        <RowDefinition Height="*"></RowDefinition>
        <RowDefinition Height="*"></RowDefinition>
        <RowDefinition Height="*"></RowDefinition>
        <RowDefinition Height="*"></RowDefinition>
        <RowDefinition Height="*"></RowDefinition>
    </Grid.RowDefinitions>
    <Grid.Resources>
        <Style TargetType="TextBox">
            <Setter Property="Margin" Value="10,5,20,5"/>
            <Setter Property="FontFamily" Value="Verdana"/>
        </Style>
        <Style TargetType="Label">
            <Setter Property="HorizontalAlignment" Value="Center"/>
            <Setter Property="Margin" Value="5,0"/>
            <Setter Property="FontWeight" Value="Black"/>
            <Setter Property="FontFamily" Value="Verdana"/>
            <Setter Property="Foreground" Value="WhiteSmoke" />
        </Style>
    </Grid.Resources>
    <Label Content="Nome" Grid.Row="0" Grid.Column="0"></Label>

```

```

        <TextBox x:Name="txtNAME" MaxLength="250" Grid.Row="0"
Grid.Column="1"></TextBox>
        <Label Content="Morada" Grid.Row="1" Grid.Column="0"></Label>
        <TextBox x:Name="txtADDRESS" MaxLength="8000" Grid.Row="1"
Grid.Column="1"></TextBox>
        <Label Content="Código Postal" Grid.Row="2" Grid.Column="0"></Label>
        <TextBox x:Name="txtCOD_POST1" Grid.Row="2" Grid.Column="1"
PreviewTextInput="NumericOnly" MaxLength="4" MaxWidth="80" MinWidth="80"
Margin="10,5,130,5" HorizontalAlignment="Left" />
        <TextBox x:Name="txtCOD_POST2" Grid.Row="2" Grid.Column="1" PreviewTextInput
="NumericOnly" MaxLength="3" MinWidth="40" MaxWidth="40" Margin="105,5,5,5"
HorizontalAlignment="Left" />
        <Label Content="Localidade" Grid.Row="3" Grid.Column="0"></Label>
        <TextBox x:Name="txtLOCALITY" MaxLength="50" Grid.Row="3"
Grid.Column="1"></TextBox>
        <Label Content="NIF" Grid.Row="4" Grid.Column="0"></Label>
        <TextBox x:Name="txtNIF" Grid.Row="4" Grid.Column="1" MaxLength="9"
PreviewTextInput="NumericOnly"></TextBox>
        <Label Content="Telemóvel" Grid.Row="5" Grid.Column="0"></Label>
        <TextBox x:Name="txtPHONE_1" Grid.Row="5" Grid.Column="1" MaxLength="9"
PreviewTextInput="NumericOnly"></TextBox>
        <Label Content="Telefone" Grid.Row="6" Grid.Column="0"></Label>
        <TextBox x:Name="txtPHONE_2" Grid.Row="6" Grid.Column="1" MaxLength="9"
PreviewTextInput="NumericOnly"></TextBox>
        <Label Content="EMAIL" Grid.Row="7" Grid.Column="0"></Label>
        <TextBox x:Name="txtE_MAIL" MaxLength="50" Grid.Row="7"
Grid.Column="1"></TextBox>
        <Label Content="Imagem" Grid.Row="8" Grid.Column="0"></Label>
        <TextBox x:Name="txtIMAGE" MaxLength="250" Grid.Row="8" Grid.Column="1"
Margin="10,5,150,5"></TextBox>
        <Button Name="btImage" Content="Browse" Grid.Row="8" Grid.Column="1"
Margin="180,3,20,3" Click="btImage_Click" ></Button>
        <Label Content="Aviso do RGPD" Grid.Row="9" Grid.Column="0"></Label>
        <ComboBox x:Name="txtCHECK_1" Grid.Row="9" Grid.Column="1"
Margin="10,5,150,3">
            <ComboBoxItem IsSelected="True">SIM</ComboBoxItem>
            <ComboBoxItem>NÃO</ComboBoxItem>
        </ComboBox>
        <Label Content="Data" Grid.Row="10" Grid.Column="0"></Label>
        <DatePicker x:Name="txtDATE" Grid.Row="10" Grid.Column="1"
Margin="10,0,20,0"></DatePicker>
        <Button Name="btInsert" Content="Insert" Grid.Row="11" Grid.ColumnSpan="2"
Margin="50,5" Click="btInsert_Click"></Button>
        <Button Name="btUpdate" Content="Update" Grid.Row="12" Grid.ColumnSpan="2"
Margin="50,5" Click="btUpdate_Click"></Button>
        <Button Name="btDelete" Content="Delete" Grid.Row="13" Grid.ColumnSpan="2"
Margin="50,5" Click="btDelete_Click"></Button>
        <Button Name="btExport" Content="Export" Grid.Row="14" Grid.ColumnSpan="2"
Margin="50,5" Click="btExport_Click"></Button>

    </Grid>

</Grid>

```

Anexo VIII - Código do ficheiro FormWindow.xaml.cs

Código funcional do "TabItem" "formHolder".

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Linq; //Biblioteca para linguagem linq
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace WorkMatic
{
    // Criação da classe "FormWindow" está é a classe responsável pelo funcionamento
    // de todos os elementos gráficos da aplicação no menu "Formulário".
    public partial class FormWindow : Grid
    {
        CLIENT currentSelectedClient;

        // Inicialização dos metodos "InitializeComponent()" e "loadfMatic()" quando a
        // classe "FormWindow" inicia.
        public FormWindow()
        {
            InitializeComponent();
            if (!DesignerProperties.GetIsInDesignMode(this))
            {
                loadfMatic();
            }
        }
        // Responsável por aceder a base de dados, ler os valores e apresentar ao
        // programa. Usa "Entities" para realizar esta função.
        public void loadfMatic()
        {
            using (WorkMaticEntities db = new WorkMaticEntities())
            {
                var qfmatic1 = (from c in db.CLIENTS select c).ToList();
                gridfMatic1.ItemsSource = qfmatic1;
            }
        }
        // Botão "insert" e usado para inserir os valores que o utilizador preenche no
        // programa e gravá-los na base de dados.
        private void btInsert_Click(object sender, RoutedEventArgs e)
        {
            if (!String.IsNullOrEmpty(txtNAME.Text) &&
                !String.IsNullOrEmpty(txtADDRESS.Text) && !String.IsNullOrEmpty(txtCOD_POST1.Text) &&
                !String.IsNullOrEmpty(txtCOD_POST2.Text) && !String.IsNullOrEmpty(txtLOCALITY.Text) &&
                !String.IsNullOrEmpty(txtNIF.Text) && !String.IsNullOrEmpty(txtPHONE_1.Text) &&
                !String.IsNullOrEmpty(txtPHONE_2.Text) && !String.IsNullOrEmpty(txtE_MAIL.Text) &&
                !String.IsNullOrEmpty(txtIMAGE.Text))
            {
                using (WorkMaticEntities db = new WorkMaticEntities())
                {

```

// Lê os valores que estão nas "textBox" e cria um objeto e os seus atributos com os valor das textboxes

```
CLIENT allClients = new CLIENT
{
    NAME_CLIENT = txtNAME.Text,
    ADDRESS_CLIENT = txtADDRESS.Text,
    COD_POST1 = int.Parse(txtCOD_POST1.Text),
    COD_POST2 = int.Parse(txtCOD_POST2.Text),
    LOCALITY = txtLOCALITY.Text,
    NIF = int.Parse(txtNIF.Text),
    PHONE_1 = int.Parse(txtPHONE_1.Text),
    PHONE_2 = int.Parse(txtPHONE_2.Text),
    E_MAIL = txtE_MAIL.Text,
    IMAGE_CLIENT = txtIMAGE.Text,
    CHECK_1 = txtCHECK_1.Text,
    DATE_CLIENT = (DateTime)txtDATE.SelectedDate
};
db.CLIENTS.Add(allClients);
db.SaveChanges();
gridfMatic1.ItemsSource = db.CLIENTS.ToList();
}
else
{
    MessageBox.Show("Preencha todos os dados", "Cuidado",
    MessageBoxButton.OK, MessageBoxImage.Information);
}
```

// Botão "Image" serve para o utilizar poder guardar imagens no formulário. Para isso usa a janela de pesquisa de ficheiros do Windows.

```
private void btImage_Click(object sender, RoutedEventArgs e)
{
    // Criação do "OpenFileDialog" que é responsável por apresentar a janela
    de pesquisa de ficheiro do Windowss.
    Microsoft.Win32.OpenFileDialog dlg = new Microsoft.Win32.OpenFileDialog();

    // Limita os tipos de ficheiros que o utilizador pode escolher e vem pré
    selecionado o tipo ".png".
    dlg.DefaultExt = ".png";
    dlg.Filter = "PNG Files (*.png)|*.png|JPEG Files (*.jpeg)|*.jpeg|JPG Files
    (*.jpg)|*.jpg|GIF Files (*.gif)|*.gif";

    // Apresenta "OpenFileDialog" usando o metodo "ShowDialog"
    Nullable<bool> result = dlg.ShowDialog();

    // Coloca o caminho mais o nome do ficheiro numa "textBox" criada.
    if (result == true)
    {
        string filename = dlg.FileName;
        txtIMAGE.Text = filename;
    }
}
```

//Botão "update" e usado para o utilizador poder corrigir ou atualizar os valores que já estão preenchidos na base de dados.

```
private void btUpdate_Click(object sender, RoutedEventArgs e)
{
    if (currentSelectedClient != null)
    {
```

```

        using (WorkMaticEntities db = new WorkMaticEntities())
        {
            int ID_CLIENTE = currentSelectedClient.ID_CLIENT;
            CLIENT clientToUpdate = db.CLIENTS.Where(x => x.ID_CLIENT ==
ID_CLIENTE).FirstOrDefault();

            clientToUpdate.NAME_CLIENT = txtNAME.Text;
            clientToUpdate.ADDRESS_CLIENT = txtADDRESS.Text;
            clientToUpdate.COD_POST1 = int.Parse(txtCOD_POST1.Text);
            clientToUpdate.COD_POST2 = int.Parse(txtCOD_POST2.Text);
            clientToUpdate.LOCALITY = txtLOCALITY.Text;
            clientToUpdate.NIF = int.Parse(txtNIF.Text);
            clientToUpdate.PHONE_1 = int.Parse(txtPHONE_1.Text);
            clientToUpdate.PHONE_2 = int.Parse(txtPHONE_2.Text);
            clientToUpdate.E_MAIL = txtE_MAIL.Text;
            clientToUpdate.IMAGE_CLIENT = txtIMAGE.Text;
            clientToUpdate.CHECK_1 = txtCHECK_1.Text;
            clientToUpdate.DATE_CLIENT = (DateTime)txtDATE.SelectedDate;

            db.SaveChanges();
            gridfMatic1.ItemsSource = db.CLIENTS.ToList();
        }
    }
    else
    {
        MessageBox.Show("Preencha todos os dados", "Cuidado",
        MessageBoxButton.OK, MessageBoxImage.Information);
    }
}
//Serve para obter o cliente selecionado nas tabelas
private void GridsF_SelectionChanged(object sender, SelectionChangedEventArgs
e)
{
    if (e.AddedItems.Count != 0)
    {
        using (WorkMaticEntities db = new WorkMaticEntities())
        {
            int ID_CLIENTE = ((CLIENT)e.AddedItems[0]).ID_CLIENT;
            CLIENT clientToUpdate = db.CLIENTS.Where(x => x.ID_CLIENT ==
ID_CLIENTE).FirstOrDefault();
            currentSelectedClient = clientToUpdate;

            txtNAME.Text = clientToUpdate.NAME_CLIENT;
            txtADDRESS.Text = clientToUpdate.ADDRESS_CLIENT;
            txtCOD_POST1.Text = clientToUpdate.COD_POST1.ToString();
            txtCOD_POST2.Text = clientToUpdate.COD_POST2.ToString();
            txtLOCALITY.Text = clientToUpdate.LOCALITY;
            txtNIF.Text = clientToUpdate.NIF.ToString();
            txtPHONE_1.Text = clientToUpdate.PHONE_1.ToString();
            txtPHONE_2.Text = clientToUpdate.PHONE_2.ToString();
            txtE_MAIL.Text = clientToUpdate.E_MAIL;
            txtIMAGE.Text = clientToUpdate.IMAGE_CLIENT;
            txtCHECK_1.Text = clientToUpdate.CHECK_1;
            txtDATE.SelectedDate = clientToUpdate.DATE_CLIENT;
        }
    }
    else
    {
        currentSelectedClient = null;
        txtNAME.Text = String.Empty;
    }
}

```

```

        txtADDRESS.Text = String.Empty;
        txtCOD_POST1.Text = String.Empty;
        txtCOD_POST2.Text = String.Empty;
        txtLOCALITY.Text = String.Empty;
        txtNIF.Text = String.Empty;
        txtPHONE_1.Text = String.Empty;
        txtPHONE_2.Text = String.Empty;
        txtE_MAIL.Text = String.Empty;
        txtIMAGE.Text = String.Empty;
        txtCHECK_1.Text = String.Empty;
        txtDATE.SelectedDate = DateTime.Now;
    }
}
//Botão "delete" e usado para o utilizador poder apagar os valores que já
estão preenchidos na base de dados.
private void btDelete_Click(object sender, RoutedEventArgs e)
{
    if (currentSelectedClient != null)
    {
        using (WorkMaticEntities db = new WorkMaticEntities())
        {
            int ID_CLIENTE = currentSelectedClient.ID_CLIENTE;
            CLIENT clientToUpdate = db.CLIENTS.Where(x => x.ID_CLIENTE ==
ID_CLIENTE).FirstOrDefault();

            db.CLIENTS.Remove(clientToUpdate);

            db.SaveChanges();
            gridfMatic1.ItemsSource = db.CLIENTS.ToList();
        }
    }
    else
    {
        MessageBox.Show("Preencha todos os dados", "Cuidado",
MessageBoxButton.OK, MessageBoxImage.Information);
    }
}
//Botão "export" implementa a classe "ExportPdf", mas deixa o utilizador
escolher a pasta de destino usando para isso a janela de pesquisa de ficheiros do
Windows.
private void btExport_Click(object sender, RoutedEventArgs e)
{
    if (currentSelectedClient != null)
    {
        using (WorkMaticEntities db = new WorkMaticEntities())
        {
            int ID_CLIENTE = currentSelectedClient.ID_CLIENTE;
            CLIENT clientToExport = db.CLIENTS.Where(x => x.ID_CLIENTE ==
ID_CLIENTE).FirstOrDefault();

            Microsoft.Win32.SaveFileDialog dlg = new
Microsoft.Win32.SaveFileDialog();

            dlg.DefaultExt = ".pdf";
            dlg.Filter = "PDF Files (*.pdf)|*.pdf";
            // Apresenta "SaveFileDialog" usando o metodo "ShowDialog"
            Nullable<bool> result = dlg.ShowDialog();

```

```

        // Coloca o caminho mais o nome do ficheiro numa "textBox" criada.
        if (result == true)
        {
            string filename = dlg.FileName;
            ExportPdf.ExportClient(filename, clientToExport);
        }
    }
}
else
{
    MessageBox.Show("preencha todos os dados", "Cuidado",
    MessageBoxButton.OK, MessageBoxImage.Information);
}

}
// Não permite usar letras numa textbox
private void NumericOnly(object sender, TextCompositionEventArgs e)
{
    Regex regex = new Regex("[^0-9]+");
    e.Handled = regex.IsMatch(e.Text);
}
}
}
}

```

Anexo IX - Código do ficheiro ProductWindow.xaml.

Código do grafico do "TabItem" "productHolder".

```

<Grid x:Class="WorkMatic.ProductWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:local="clr-namespace:WorkMatic"
    mc:Ignorable="d">
    <Grid.RowDefinitions>
        <RowDefinition />
    </Grid.RowDefinitions>
    <Grid.ColumnDefinitions>
        <ColumnDefinition Width="auto"></ColumnDefinition>
        <ColumnDefinition Width="3*"></ColumnDefinition>
    </Grid.ColumnDefinitions>
    <DataGrid Name="gridpMatic1" IsReadOnly="true"
    SelectionChanged="GridsP_SelectionChanged" AutoGenerateColumns="False" Grid.Column="0"
    Grid.Row="0" SelectionMode="Single" Grid.IsSharedSizeScope="True"
    CanUserResizeRows="False" CanUserResizeColumns="False" CanUserReorderColumns="False"
    CanUserSortColumns="False" FontFamily="Verdana" Background="Gray">
        <DataGrid.Resources>
            <local:MoneyConverter x:Key="MoneyConverter" />
        </DataGrid.Resources>
        <DataGrid.Columns>
            <DataGridTextColumn Header="Produto" Binding="{Binding TYPE_PRODUCT}"
    MinWidth="325" MaxWidth="8000"/>
            <DataGridTextColumn Header="Preço" Binding="{Binding PRICE_PRODUCT,
    Converter={StaticResource MoneyConverter}}" MinWidth="100" MaxWidth="100"/>
        </DataGrid.Columns>
    </DataGrid>

```



```

<Grid Grid.Column="1" Grid.RowSpan="2" Background="{DynamicResource {x:Static
SystemColors.WindowFrameBrushKey}}">
  <Grid.ColumnDefinitions>
    <ColumnDefinition Width="1*"></ColumnDefinition>
    <ColumnDefinition Width="3*"></ColumnDefinition>
  </Grid.ColumnDefinitions>
  <Grid.RowDefinitions>
    <RowDefinition Height="*"></RowDefinition>
    <RowDefinition Height="*"></RowDefinition>
    <RowDefinition Height="*"></RowDefinition>
    <RowDefinition Height="*"></RowDefinition>
    <RowDefinition Height="*"></RowDefinition>
    <RowDefinition Height="*"></RowDefinition>
    <RowDefinition Height="*"></RowDefinition>
    <RowDefinition Height="*"></RowDefinition>
    <RowDefinition Height="*"></RowDefinition>
    <RowDefinition Height="*"></RowDefinition>
    <RowDefinition Height="*"></RowDefinition>
    <RowDefinition Height="*"></RowDefinition>
    <RowDefinition Height="*"></RowDefinition>
    <RowDefinition Height="*"></RowDefinition>
  </Grid.RowDefinitions>
  <Grid.Resources>
    <Style TargetType="TextBox">
      <Setter Property="Margin" Value="10,5,20,5"/>
      <Setter Property="FontFamily" Value="Verdana"/>
    </Style>
    <Style TargetType="Label">
      <Setter Property="HorizontalAlignment" Value="Center"/>
      <Setter Property="Margin" Value="5,0"/>
      <Setter Property="FontWeight" Value="Black"/>
      <Setter Property="FontFamily" Value="Verdana"/>
      <Setter Property="Foreground" Value="WhiteSmoke" />
    </Style>
  </Grid.Resources>
  <Label Content="Tipo" Grid.Row="0" Grid.Column="0"></Label>
  <TextBox x:Name="txtTYPE" MaxLength="50" Grid.Row="0"
Grid.Column="1"></TextBox>
  <Label Content="Preço" Grid.Row="1" Grid.Column="0"></Label>
  <TextBox x:Name="txtPRICE" MaxLength="10" Grid.Row="1"
Grid.Column="1"></TextBox>
  <Button Name="btpInsert" Content="Insert" Grid.Row="11" Grid.ColumnSpan="2"
Margin="50,5" Click="btInsert_Click"/>
  <Button Name="btpUpdate" Content="Update" Grid.Row="12" Grid.ColumnSpan="2"
Margin="50,5" Click="btUpdate_Click"></Button>
  <Button Name="btpDelete" Content="Delete" Grid.Row="13" Grid.ColumnSpan="2"
Margin="50,5" Click="btDelete_Click"></Button>

</Grid>

</Grid>

```

Anexo X - Código do ficheiro ProductWindow.xaml.cs.

Código funcional do "TabItem" "productHolder".

```

using System;
using System.Collections.Generic;

```

```

using System.ComponentModel;
using System.Data.Entity.Infrastructure;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;

namespace WorkMatic
{
    // Criação da classe "ProductWindow" está é a classe responsável pelo
    funcionamento de todos os elementos gráficos da aplicação no menu "Produtos".
    public partial class ProductWindow : Grid
    {
        PRODUCT currentSelectedProduct;

        // Inicialização dos metodos "InitializeComponent()" e "loadpMatic()" quando a
        classe "ProductWindow" inicia.
        public ProductWindow()
        {
            InitializeComponent();
            if (!DesignerProperties.GetIsInDesignMode(this))
            {
                loadpMatic();
            }
        }

        // Responsável por aceder a base de dados, ler os valores e apresentar ao
        programa. Usa "Entities" para realizar esta função.
        public void loadpMatic()
        {
            using (WorkMaticEntities db = new WorkMaticEntities())
            {
                var qomatic1 = (from c in db.PRODUCTS select c).ToList();
                gridpMatic1.ItemsSource = qomatic1;
            }
        }

        // Botão "insert" e usado para inserir os valores que o administrador preenche
        no programa e gravá-los na base de dados.
        private void btInsert_Click(object sender, RoutedEventArgs e)
        {
            if (!String.IsNullOrEmpty(txtTYPE.Text) &&
                !String.IsNullOrEmpty(txtPRICE.Text))
            {
                using (WorkMaticEntities db = new WorkMaticEntities())
                {
                    // Lê os valores que estão nas "txtBox" e cria um objeto e os seus
                    atributos com os valor das textboxes
                    PRODUCT allProducts = new PRODUCT
                    {
                        TYPE_PRODUCT = txtTYPE.Text,
                        PRICE_PRODUCT = decimal.Parse(txtPRICE.Text)
                    }
                }
            }
        }
    }
}

```

```

        };
        db.PRODUCTS.Add(allProducts);
        db.SaveChanges();
        gridpMatic1.ItemsSource = db.PRODUCTS.ToList();
    }
}
else
{
    MessageBox.Show("Preencha todos os dados", "Cuidado",
    MessageBoxButton.OK, MessageBoxImage.Information);
}
}

//Botão "update" e usado para o administrados poder corrigir ou atualizar os
valores que já estão preenchidos na base de dados.
private void btUpdate_Click(object sender, RoutedEventArgs e)
{
    if (currentSelectedProduct != null)
    {
        using (WorkMaticEntities db = new WorkMaticEntities())
        {
            int ID_PRODUCT = currentSelectedProduct.ID_PRODUCT;
            PRODUCT productToUpdate = db.PRODUCTS.Where(x => x.ID_PRODUCT ==
ID_PRODUCT).FirstOrDefault();

            {
                productToUpdate.TYPE_PRODUCT = txtTYPE.Text;
                productToUpdate.PRICE_PRODUCT = decimal.Parse(txtPRICE.Text);
            };

            db.SaveChanges();
            gridpMatic1.ItemsSource = db.PRODUCTS.ToList();
        }
    }
    else
    {
        MessageBox.Show("Preencha todos os dados", "Cuidado",
        MessageBoxButton.OK, MessageBoxImage.Information);
    }
}

//Serve para obter o produto selecionado nas tabelas
private void GridsP_SelectionChanged(object sender, SelectionChangedEventArgs
e)
{
    if (e.AddedItems.Count != 0)
    {
        using (WorkMaticEntities db = new WorkMaticEntities())
        {
            int ID_PRODUCT = ((PRODUCT)e.AddedItems[0]).ID_PRODUCT;
            PRODUCT productToUpdate = db.PRODUCTS.Where(x => x.ID_PRODUCT ==
ID_PRODUCT).FirstOrDefault();
            currentSelectedProduct = productToUpdate;

            txtTYPE.Text = productToUpdate.TYPE_PRODUCT;
            setItemPrice((decimal) productToUpdate.PRICE_PRODUCT);
        }
    }
}

```

```

        else
        {
            currentSelectedProduct = null;
            txtTYPE.Text = String.Empty;
            txtPRICE.Text = String.Empty;
        }
    }

    //Botão "delete" e usado para o administrador poder apagar os valores que já
    estão preenchidos na base de dados.
    private void btDelete_Click(object sender, RoutedEventArgs e)
    {
        if (currentSelectedProduct != null)
        {
            using (WorkMaticEntities db = new WorkMaticEntities())
            {
                int ID_PRODUCT = currentSelectedProduct.ID_PRODUCT;
                PRODUCT productToUpdate = db.PRODUCTS.Where(x => x.ID_PRODUCT ==
                ID_PRODUCT).FirstOrDefault();

                db.PRODUCTS.Remove(productToUpdate);

                db.SaveChanges();
                gridpMatic1.ItemsSource = db.CLIENTS.ToList();
            }
        }
        else
        {
            MessageBox.Show("Os campos tem de estar todos preechidos", "Cuidado",
            MessageBoxButton.OK, MessageBoxImage.Information);
        }
    }
    private void setItemPrice(decimal price)
    {
        txtPRICE.Text = Math.Round(price, 2).ToString() + "€";
    }
}
}

```

Anexo XI - Código da classe ExportPdf.cs.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.IO; //Biblioteca para entrada e saida de ficheiros
using iTextSharp; //Biblioteca do iTextSharp e extensões
using iTextSharp.text; //Extensão (text)
using iTextSharp.text.pdf; //Extensão (pdf)
using System.Windows;

namespace WorkMatic
{
    // Criação da classe "exportPdf" serve para que o utilizador possa exportar os
    valores de o formulário para um documento tipo PDF.
    class ExportPdf
    {
        // Metodo que exporta quando se usa o botão na grid FormWindow
    }
}

```

```

public static void ExportClient(string caminho, CLIENT CLIENT)
{
    if (!string.IsNullOrEmpty(caminho))
    {
        Document doc = new Document(PageSize.A4);
        doc.SetMargins(40, 40, 40, 80);
        doc.AddCreationDate();

        PdfWriter writer = PdfWriter.GetInstance(doc, new FileStream(caminho,
        FileMode.Create));
        Image image = Image.GetInstance(new Uri(CLIENT.IMAGE_CLIENT));
        image.ScaleToFit(new Rectangle(256, 512));

        doc.Open();

        string data = "";

        try
        {
            Paragraph title = new Paragraph(data, new Font(Font.NORMAL, 16));
            title.Alignment = Element.ALIGN_CENTER;
            title.Alignment = Element.TITLE;
            title.Add("Formulario");
            title.Add(Environment.NewLine);
            title.Add(Environment.NewLine);
            title.Add(Environment.NewLine);
            title.Add(Environment.NewLine);

            Paragraph paragraph = new Paragraph(data, new Font(Font.NORMAL,
14));
            paragraph.Alignment = Element.ALIGN_JUSTIFIED;
            paragraph.Add("Nome : " + CLIENT.NAME_CLIENT);
            paragraph.Add(Environment.NewLine);
            paragraph.Add(Environment.NewLine);
            paragraph.Add("Morada : " + CLIENT.ADDRESS_CLIENT);
            paragraph.Add(Environment.NewLine);
            paragraph.Add(Environment.NewLine);
            paragraph.Add("Codigo Postal : " + CLIENT.COD_POST1.ToString() + "
- " + CLIENT.COD_POST2.ToString());
            paragraph.Add(Environment.NewLine);
            paragraph.Add(Environment.NewLine);
            paragraph.Add("Localidade : " + CLIENT.LOCALITY);
            paragraph.Add(Environment.NewLine);
            paragraph.Add(Environment.NewLine);
            paragraph.Add("NIF : " + CLIENT.NIF.ToString());
            paragraph.Add(Environment.NewLine);
            paragraph.Add(Environment.NewLine);
            paragraph.Add("Telemóvel : " + CLIENT.PHONE_1.ToString());
            paragraph.Add(Environment.NewLine);
            paragraph.Add(Environment.NewLine);
            paragraph.Add("Telefone : " + CLIENT.PHONE_2.ToString());
            paragraph.Add(Environment.NewLine);
            paragraph.Add(Environment.NewLine);
            paragraph.Add("EMAIL : " + CLIENT.E_MAIL);
            paragraph.Add(Environment.NewLine);
            paragraph.Add(Environment.NewLine);
            paragraph.Add("RGPD : " + CLIENT.CHECK_1);
            paragraph.Add(Environment.NewLine);
            paragraph.Add(Environment.NewLine);
            paragraph.Add("Data : " + CLIENT.DATE_CLIENT.ToString());

```

```

        paragraph.Add(Environment.NewLine);
        paragraph.Add(Environment.NewLine);

        doc.Add(title);
        doc.Add(paragraph);
        doc.Add(image);
        doc.AddAuthor(Environment.UserName);

        MessageBox.Show("O documento foi gravado com sucesso", "Gravado",
        MessageBoxButton.OK, MessageBoxImage.Information);
        doc.Close();

    }

    catch (Exception ex)
    {
        MessageBox.Show("Erro foi " + ex.Message, "Aviso",
        MessageBoxButton.OK, MessageBoxImage.Information);
    }
}

// Metodo que exporta quando se usa o botão na grid OrderWindow
public static void ExportOrder(string caminho, ORDER ORDER)
{
    if (!string.IsNullOrEmpty(caminho))
    {
        Document doc = new Document(PageSize.A4);
        doc.SetMargins(40, 40, 40, 80);
        doc.AddCreationDate();

        PdfWriter writer = PdfWriter.GetInstance(doc, new FileStream(caminho,
        FileMode.Create));

        doc.Open();

        string data = "";

        try
        {
            Paragraph title = new Paragraph(data, new Font(Font.NORMAL, 16));
            title.Alignment = Element.ALIGN_CENTER;
            title.Alignment = Element.TITLE;
            title.Add("Recibo");
            title.Add(Environment.NewLine);
            title.Add(Environment.NewLine);
            title.Add(Environment.NewLine);
            title.Add(Environment.NewLine);

            Paragraph paragraph = new Paragraph(data, new Font(Font.NORMAL,
14));

            paragraph.Alignment = Element.ALIGN_JUSTIFIED;
            paragraph.Add("Nome : " + ORDER.CLIENT.NAME_CLIENT);
            paragraph.Add(Environment.NewLine);
            paragraph.Add(Environment.NewLine);
            paragraph.Add("Morada : " + ORDER.CLIENT.ADDRESS_CLIENT);
            paragraph.Add(Environment.NewLine);
            paragraph.Add(Environment.NewLine);
            paragraph.Add("NIF : " + ORDER.CLIENT.NIF.ToString());
            paragraph.Add(Environment.NewLine);
            paragraph.Add(Environment.NewLine);

```

```

        paragraph.Add("Telemóvel : " + ORDER.CLIENT.PHONE_1.ToString());
        paragraph.Add(Environment.NewLine);
        paragraph.Add(Environment.NewLine);
        paragraph.Add("Telefone : " + ORDER.CLIENT.PHONE_2.ToString());
        paragraph.Add(Environment.NewLine);
        paragraph.Add(Environment.NewLine);
        paragraph.Add("EMAIL : " + ORDER.CLIENT.E_MAIL);
        paragraph.Add(Environment.NewLine);
        paragraph.Add(Environment.NewLine);
        paragraph.Add("Produto : " +
ORDER.PRODUCT.TYPE_PRODUCT.ToString());
        paragraph.Add(Environment.NewLine);
        paragraph.Add(Environment.NewLine);
        paragraph.Add("Preço : " +
Math.Round((decimal)ORDER.PRODUCT.PRICE_PRODUCT,2).ToString() + " €");
        paragraph.Add(Environment.NewLine);
        paragraph.Add(Environment.NewLine);
        paragraph.Add("Quantidade : " + ORDER.QUANTITY.ToString());
        paragraph.Add(Environment.NewLine);
        paragraph.Add(Environment.NewLine);
        paragraph.Add("Data : " + ORDER.DATE_ORDERS.ToString());
        paragraph.Add(Environment.NewLine);
        paragraph.Add(Environment.NewLine);
        paragraph.Add("Hora : " + ORDER.TIME_ORDERS.ToString());
        paragraph.Add(Environment.NewLine);
        paragraph.Add(Environment.NewLine);

        Paragraph price = new Paragraph(data, new Font(Font.NORMAL, 16));
        price.Alignment = Element.ALIGN_RIGHT;
        price.Alignment = Element.TITLE;
        price.Add("Preço : " +
Math.Round((decimal)ORDER.TOTAL_PRICE,2).ToString() + " €");
        price.Add(Environment.NewLine);

        doc.Add(title);
        doc.Add(paragraph);
        doc.Add(price);
        doc.AddAuthor(Environment.UserName);

        MessageBox.Show("O documento foi gravado com sucesso", "Gravado",
MessageBoxButton.OK, MessageBoxImage.Information);
        doc.Close();

    }

    catch (Exception ex)
    {
        MessageBox.Show("Erro foi " + ex.Message, "Aviso",
MessageBoxButton.OK, MessageBoxImage.Information);
    }
}
}
}

```

Anexo XII - Código da classe MoneyConverter.cs.

```
using System;
using System.Collections.Generic;
using System.Globalization;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Data;

namespace WorkMatic
{
    // Criação da classe "MoneyConverter" serve para converter os valores decimal numa
    // string com duas casas decimais e o símbolo do "€"
    class MoneyConverter : IValueConverter
    {
        public object Convert(object value, Type targetType, object parameter,
CultureInfo culture)
        {
            if(value is decimal)
            {
                var valueToConvert = (decimal)value;

                return Math.Round(valueToConvert, 2).ToString() + "€";
            }

            return string.Empty;
        }

        public object ConvertBack(object value, Type targetType, object parameter,
CultureInfo culture)
        {
            throw new NotImplementedException();
        }
    }
}
```

Anexo XIII - Código do ficheiro WorkMatic.edmx.

```
<?xml version="1.0" encoding="utf-8"?>
<edmx:Edmx Version="3.0" xmlns:edmx="http://schemas.microsoft.com/ado/2009/11/edmx">
  <!-- EF Runtime content -->
  <edmx:Runtime>
    <!-- SSDL content -->
    <edmx:StorageModels>
      <Schema Namespace="WorkMaticModel.Store" Provider="System.Data.SqlClient"
ProviderManifestToken="2012" Alias="Self"
xmlns:store="http://schemas.microsoft.com/ado/2007/12/edm/EntityStoreSchemaGenerator"
xmlns:customannotation="http://schemas.microsoft.com/ado/2013/11/edm/customannotation"
xmlns="http://schemas.microsoft.com/ado/2009/11/edm/ssdl">
        <EntityType Name="ADMINS">
          <Key>
            <PropertyRef Name="ID_ADMIN" />
          </Key>
        </EntityType>
      </Schema>
    </edmx:StorageModels>
  </edmx:Runtime>
</edmx:Edmx>
```



```

        <Property Name="ID_ADMIN" Type="int" StoreGeneratedPattern="Identity"
Nullable="false" />
        <Property Name="ID_USER" Type="int" />
        <Property Name="CAN_READ_USER" Type="bit" Nullable="false" />
        <Property Name="FULL_ACCESS" Type="bit" Nullable="false" />
    </EntityType>
    <EntityType Name="CLIENTS">
        <Key>
            <PropertyRef Name="ID_CLIENT" />
        </Key>
        <Property Name="ID_CLIENT" Type="int" StoreGeneratedPattern="Identity"
Nullable="false" />
        <Property Name="NAME_CLIENT" Type="varchar" MaxLength="255" Nullable="false"
/>
        <Property Name="ADDRESS_CLIENT" Type="varchar(max)" Nullable="false" />
        <Property Name="COD_POST1" Type="int" />
        <Property Name="COD_POST2" Type="int" />
        <Property Name="LOCALITY" Type="varchar" MaxLength="50" />
        <Property Name="NIF" Type="int" Nullable="false" />
        <Property Name="PHONE_1" Type="int" />
        <Property Name="PHONE_2" Type="int" />
        <Property Name="E_MAIL" Type="varchar" MaxLength="50" />
        <Property Name="IMAGE_CLIENT" Type="varchar" MaxLength="255" />
        <Property Name="DATE_CLIENT" Type="date" Nullable="false" />
        <Property Name="CHECK_1" Type="varchar" MaxLength="3" Nullable="false" />
    </EntityType>
    <EntityType Name="ORDERS">
        <Key>
            <PropertyRef Name="ID_ORDERS" />
        </Key>
        <Property Name="ID_ORDERS" Type="int" StoreGeneratedPattern="Identity"
Nullable="false" />
        <Property Name="ID_USER" Type="int" />
        <Property Name="ID_CLIENT" Type="int" />
        <Property Name="ID_PRODUCTS" Type="int" />
        <Property Name="QUANTITY" Type="tinyint" Nullable="false" />
        <Property Name="DATE_ORDERS" Type="date" Nullable="false" />
        <Property Name="TIME_ORDERS" Type="time" Precision="7" Nullable="false" />
        <Property Name="TOTAL_PRICE" Type="smallmoney" />
    </EntityType>
    <EntityType Name="PRODUCTS">
        <Key>
            <PropertyRef Name="ID_PRODUCT" />
        </Key>
        <Property Name="ID_PRODUCT" Type="int" StoreGeneratedPattern="Identity"
Nullable="false" />
        <Property Name="TYPE_PRODUCT" Type="varchar(max)" Nullable="false" />
        <Property Name="PRICE_PRODUCT" Type="smallmoney" Nullable="false" />
    </EntityType>
    <EntityType Name="sysdiagrams">
        <Key>
            <PropertyRef Name="diagram_id" />
        </Key>
        <Property Name="name" Type="nvarchar" MaxLength="128" Nullable="false" />
        <Property Name="principal_id" Type="int" Nullable="false" />
        <Property Name="diagram_id" Type="int" StoreGeneratedPattern="Identity"
Nullable="false" />
        <Property Name="version" Type="int" />
        <Property Name="definition" Type="varbinary(max)" />
    </EntityType>
    <EntityType Name="USERS">

```

```

    <Key>
      <PropertyRef Name="ID_USER" />
    </Key>
    <Property Name="ID_USER" Type="int" StoreGeneratedPattern="Identity"
Nullable="false" />
    <Property Name="NAME_USER" Type="varchar" MaxLength="255" Nullable="false"
/>
  </EntityType>
  <Association Name="FK__ADMINS__ID_USER__5441852A">
    <End Role="USERS" Type="Self.USERS" Multiplicity="0..1" />
    <End Role="ADMINS" Type="Self.ADMINS" Multiplicity="*" />
    <ReferentialConstraint>
      <Principal Role="USERS">
        <PropertyRef Name="ID_USER" />
      </Principal>
      <Dependent Role="ADMINS">
        <PropertyRef Name="ID_USER" />
      </Dependent>
    </ReferentialConstraint>
  </Association>
  <Association Name="FK__ORDERS__ID_CLIEN__59FA5E80">
    <End Role="CLIENTS" Type="Self.CLIENTS" Multiplicity="0..1">
      <onDelete Action="Cascade" />
    </End>
    <End Role="ORDERS" Type="Self.ORDERS" Multiplicity="*" />
    <ReferentialConstraint>
      <Principal Role="CLIENTS">
        <PropertyRef Name="ID_CLIENT" />
      </Principal>
      <Dependent Role="ORDERS">
        <PropertyRef Name="ID_CLIENT" />
      </Dependent>
    </ReferentialConstraint>
  </Association>
  <Association Name="FK__ORDERS__ID_PRODU__5AEE82B9">
    <End Role="PRODUCTS" Type="Self.PRODUCTS" Multiplicity="0..1" />
    <End Role="ORDERS" Type="Self.ORDERS" Multiplicity="*" />
    <ReferentialConstraint>
      <Principal Role="PRODUCTS">
        <PropertyRef Name="ID_PRODUCT" />
      </Principal>
      <Dependent Role="ORDERS">
        <PropertyRef Name="ID_PRODUCTS" />
      </Dependent>
    </ReferentialConstraint>
  </Association>
  <Association Name="FK__ORDERS__ID_USER__59063A47">
    <End Role="USERS" Type="Self.USERS" Multiplicity="0..1" />
    <End Role="ORDERS" Type="Self.ORDERS" Multiplicity="*" />
    <ReferentialConstraint>
      <Principal Role="USERS">
        <PropertyRef Name="ID_USER" />
      </Principal>
      <Dependent Role="ORDERS">
        <PropertyRef Name="ID_USER" />
      </Dependent>
    </ReferentialConstraint>
  </Association>
  <Function Name="CLIENT_ORDERS" Aggregate="false" BuiltIn="false"
NiladicFunction="false" IsComposable="true"
ParameterTypeSemantics="AllowImplicitConversion" Schema="dbo">

```

```

<Parameter Name="ClientId" Type="int" Mode="In" />
<ReturnType>
  <CollectionType>
    <RowType>
      <Property Name="ID_ORDERS" Type="int" Nullable="false" />
      <Property Name="ID_USER" Type="int" />
      <Property Name="ID_CLIENT" Type="int" />
      <Property Name="ID_PRODUCTS" Type="int" />
      <Property Name="QUANTITY" Type="tinyint" Nullable="false" />
      <Property Name="DATE_ORDERS" Type="date" Nullable="false" />
      <Property Name="TIME_ORDERS" Type="time" Precision="7"
Nullable="false" />
      <Property Name="TOTAL_PRICE" Type="smallmoney" />
      <Property Name="ID_PRODUCT" Type="int" Nullable="false" />
      <Property Name="TYPE_PRODUCT" Type="varchar(max)" Nullable="false" />
      <Property Name="PRICE_PRODUCT" Type="smallmoney" Nullable="false" />
    </RowType>
  </CollectionType>
</ReturnType>
</Function>
<Function Name="fn_diagramobjects" Aggregate="false" BuiltIn="false"
NiladicFunction="false" IsComposable="true"
ParameterTypeSemantics="AllowImplicitConversion" Schema="dbo" ReturnType="int" />
<Function Name="sp_alterdiagram" Aggregate="false" BuiltIn="false"
NiladicFunction="false" IsComposable="false"
ParameterTypeSemantics="AllowImplicitConversion" Schema="dbo">
  <Parameter Name="diagramname" Type="nvarchar" Mode="In" />
  <Parameter Name="owner_id" Type="int" Mode="In" />
  <Parameter Name="version" Type="int" Mode="In" />
  <Parameter Name="definition" Type="varbinary(max)" Mode="In" />
</Function>
<Function Name="sp_creatediagram" Aggregate="false" BuiltIn="false"
NiladicFunction="false" IsComposable="false"
ParameterTypeSemantics="AllowImplicitConversion" Schema="dbo">
  <Parameter Name="diagramname" Type="nvarchar" Mode="In" />
  <Parameter Name="owner_id" Type="int" Mode="In" />
  <Parameter Name="version" Type="int" Mode="In" />
  <Parameter Name="definition" Type="varbinary(max)" Mode="In" />
</Function>
<Function Name="sp_dropdiagram" Aggregate="false" BuiltIn="false"
NiladicFunction="false" IsComposable="false"
ParameterTypeSemantics="AllowImplicitConversion" Schema="dbo">
  <Parameter Name="diagramname" Type="nvarchar" Mode="In" />
  <Parameter Name="owner_id" Type="int" Mode="In" />
</Function>
<Function Name="sp_helpdiagramdefinition" Aggregate="false" BuiltIn="false"
NiladicFunction="false" IsComposable="false"
ParameterTypeSemantics="AllowImplicitConversion" Schema="dbo">
  <Parameter Name="diagramname" Type="nvarchar" Mode="In" />
  <Parameter Name="owner_id" Type="int" Mode="In" />
</Function>
<Function Name="sp_helpdiagrams" Aggregate="false" BuiltIn="false"
NiladicFunction="false" IsComposable="false"
ParameterTypeSemantics="AllowImplicitConversion" Schema="dbo">
  <Parameter Name="diagramname" Type="nvarchar" Mode="In" />
  <Parameter Name="owner_id" Type="int" Mode="In" />
</Function>
<Function Name="sp_renamediagram" Aggregate="false" BuiltIn="false"
NiladicFunction="false" IsComposable="false"
ParameterTypeSemantics="AllowImplicitConversion" Schema="dbo">
  <Parameter Name="diagramname" Type="nvarchar" Mode="In" />

```

```

        <Parameter Name="owner_id" Type="int" Mode="In" />
        <Parameter Name="new_diagramname" Type="nvarchar" Mode="In" />
    </Function>
    <Function Name="sp_upgraddiagrams" Aggregate="false" BuiltIn="false"
    NiladicFunction="false" IsComposable="false"
    ParameterTypeSemantics="AllowImplicitConversion" Schema="dbo" />
    <EntityContainer Name="WorkMaticModelStoreContainer">
        <EntitySet Name="ADMINS" EntityType="Self.ADMINS" Schema="dbo"
store:Type="Tables" />
        <EntitySet Name="CLIENTS" EntityType="Self.CLIENTS" Schema="dbo"
store:Type="Tables" />
        <EntitySet Name="ORDERS" EntityType="Self.ORDERS" Schema="dbo"
store:Type="Tables" />
        <EntitySet Name="PRODUCTS" EntityType="Self.PRODUCTS" Schema="dbo"
store:Type="Tables" />
        <EntitySet Name="sysdiagrams" EntityType="Self.sysdiagrams" Schema="dbo"
store:Type="Tables" />
        <EntitySet Name="USERS" EntityType="Self.USERS" Schema="dbo"
store:Type="Tables" />
        <AssociationSet Name="FK_ADMINS_ID_USER_5441852A"
Association="Self.FK_ADMINS_ID_USER_5441852A">
            <End Role="USERS" EntitySet="USERS" />
            <End Role="ADMINS" EntitySet="ADMINS" />
        </AssociationSet>
        <AssociationSet Name="FK_ORDERS_ID_CLIEN_59FA5E80"
Association="Self.FK_ORDERS_ID_CLIEN_59FA5E80">
            <End Role="CLIENTS" EntitySet="CLIENTS" />
            <End Role="ORDERS" EntitySet="ORDERS" />
        </AssociationSet>
        <AssociationSet Name="FK_ORDERS_ID_PRODU_5AEE82B9"
Association="Self.FK_ORDERS_ID_PRODU_5AEE82B9">
            <End Role="PRODUCTS" EntitySet="PRODUCTS" />
            <End Role="ORDERS" EntitySet="ORDERS" />
        </AssociationSet>
        <AssociationSet Name="FK_ORDERS_ID_USER_59063A47"
Association="Self.FK_ORDERS_ID_USER_59063A47">
            <End Role="USERS" EntitySet="USERS" />
            <End Role="ORDERS" EntitySet="ORDERS" />
        </AssociationSet>
    </EntityContainer>
</Schema></edmx:StorageModels>
<!-- CSDL content -->
<edmx:ConceptualModels>
    <Schema Namespace="WorkMaticModel" Alias="Self"
annotation:UseStrongSpatialTypes="false"
xmlns:annotation="http://schemas.microsoft.com/ado/2009/02/edm/annotation"
xmlns:customannotation="http://schemas.microsoft.com/ado/2013/11/edm/customannotation"
xmlns="http://schemas.microsoft.com/ado/2009/11/edm">
        <EntityType Name="ADMIN">
            <Key>
                <PropertyRef Name="ID_ADMIN" />
            </Key>
            <Property Name="ID_ADMIN" Type="Int32" Nullable="false"
annotation:StoreGeneratedPattern="Identity" />
            <Property Name="ID_USER" Type="Int32" />
            <Property Name="CAN_READ_USER" Type="Boolean" Nullable="false" />
            <Property Name="FULL_ACCESS" Type="Boolean" Nullable="false" />
            <NavigationProperty Name="USER"
Relationship="Self.FK_ADMINS_ID_USER_5441852A" FromRole="ADMINS" ToRole="USERS" />
        </EntityType>
        <EntityType Name="CLIENT">

```

```

    <Key>
      <PropertyRef Name="ID_CLIENT" />
    </Key>
    <Property Name="ID_CLIENT" Type="Int32" Nullable="false"
annotation:StoreGeneratedPattern="Identity" />
    <Property Name="NAME_CLIENT" Type="String" MaxLength="255"
FixedLength="false" Unicode="false" Nullable="false" />
    <Property Name="ADDRESS_CLIENT" Type="String" MaxLength="Max"
FixedLength="false" Unicode="false" Nullable="false" />
    <Property Name="COD_POST1" Type="Int32" />
    <Property Name="COD_POST2" Type="Int32" />
    <Property Name="LOCALITY" Type="String" MaxLength="50" FixedLength="false"
Unicode="false" />
    <Property Name="NIF" Type="Int32" Nullable="false" />
    <Property Name="PHONE_1" Type="Int32" />
    <Property Name="PHONE_2" Type="Int32" />
    <Property Name="E_MAIL" Type="String" MaxLength="50" FixedLength="false"
Unicode="false" />
    <Property Name="IMAGE_CLIENT" Type="String" MaxLength="255"
FixedLength="false" Unicode="false" />
    <Property Name="DATE_CLIENT" Type="DateTime" Nullable="false" Precision="0"
/>
    <Property Name="CHECK_1" Type="String" MaxLength="3" FixedLength="false"
Unicode="false" Nullable="false" />
    <NavigationProperty Name="ORDERS"
Relationship="Self.FK_ORDERS_ID_CLIEN_59FA5E80" FromRole="CLIENTS" ToRole="ORDERS"
/>
  </EntityType>
  <EntityType Name="ORDER">
    <Key>
      <PropertyRef Name="ID_ORDERS" />
    </Key>
    <Property Name="ID_ORDERS" Type="Int32" Nullable="false"
annotation:StoreGeneratedPattern="Identity" />
    <Property Name="ID_USER" Type="Int32" />
    <Property Name="ID_CLIENT" Type="Int32" />
    <Property Name="ID_PRODUCTS" Type="Int32" />
    <Property Name="QUANTITY" Type="Byte" Nullable="false" />
    <Property Name="DATE_ORDERS" Type="DateTime" Nullable="false" Precision="0"
/>
    <Property Name="TIME_ORDERS" Type="Time" Nullable="false" Precision="7" />
    <NavigationProperty Name="CLIENT"
Relationship="Self.FK_ORDERS_ID_CLIEN_59FA5E80" FromRole="ORDERS" ToRole="CLIENTS"
/>
    <NavigationProperty Name="PRODUCT"
Relationship="Self.FK_ORDERS_ID_PRODU_5AEE82B9" FromRole="ORDERS" ToRole="PRODUCTS"
/>
    <NavigationProperty Name="USER"
Relationship="Self.FK_ORDERS_ID_USER_59063A47" FromRole="ORDERS" ToRole="USERS" />
    <Property Name="TOTAL_PRICE" Type="Decimal" Precision="10" Scale="4" />
  </EntityType>
  <EntityType Name="PRODUCT">
    <Key>
      <PropertyRef Name="ID_PRODUCT" />
    </Key>
    <Property Name="ID_PRODUCT" Type="Int32" Nullable="false"
annotation:StoreGeneratedPattern="Identity" />
    <Property Name="TYPE_PRODUCT" Type="String" MaxLength="Max"
FixedLength="false" Unicode="false" Nullable="false" />
    <Property Name="PRICE_PRODUCT" Type="Decimal" Precision="10" Scale="4"
Nullable="false" />

```

```

    <NavigationProperty Name="ORDERS"
Relationship="Self.FK__ORDERS__ID_PRODU__5AEE82B9" FromRole="PRODUCTS" ToRole="ORDERS"
/>
</EntityType>
<EntityType Name="USER">
  <Key>
    <PropertyRef Name="ID_USER" />
  </Key>
  <Property Name="ID_USER" Type="Int32" Nullable="false"
annotation:StoreGeneratedPattern="Identity" />
  <Property Name="NAME_USER" Type="String" MaxLength="255" FixedLength="false"
Unicode="false" Nullable="false" />
  <NavigationProperty Name="ADMINS"
Relationship="Self.FK__ADMINS__ID_USER__5441852A" FromRole="USERS" ToRole="ADMINS" />
  <NavigationProperty Name="ORDERS"
Relationship="Self.FK__ORDERS__ID_USER__59063A47" FromRole="USERS" ToRole="ORDERS" />
</EntityType>
<Association Name="FK__ADMINS__ID_USER__5441852A">
  <End Role="USERS" Type="Self.USER" Multiplicity="0..1" />
  <End Role="ADMINS" Type="Self.ADMIN" Multiplicity="*" />
  <ReferentialConstraint>
    <Principal Role="USERS">
      <PropertyRef Name="ID_USER" />
    </Principal>
    <Dependent Role="ADMINS">
      <PropertyRef Name="ID_USER" />
    </Dependent>
  </ReferentialConstraint>
</Association>
<Association Name="FK__ORDERS__ID_CLIEN__59FA5E80">
  <End Role="CLIENTS" Type="Self.CLIENT" Multiplicity="0..1" />
  <End Role="ORDERS" Type="Self.ORDER" Multiplicity="*" />
  <ReferentialConstraint>
    <Principal Role="CLIENTS">
      <PropertyRef Name="ID_CLIENT" />
    </Principal>
    <Dependent Role="ORDERS">
      <PropertyRef Name="ID_CLIENT" />
    </Dependent>
  </ReferentialConstraint>
</Association>
<Association Name="FK__ORDERS__ID_PRODU__5AEE82B9">
  <End Role="PRODUCTS" Type="Self.PRODUCT" Multiplicity="0..1" />
  <End Role="ORDERS" Type="Self.ORDER" Multiplicity="*" />
  <ReferentialConstraint>
    <Principal Role="PRODUCTS">
      <PropertyRef Name="ID_PRODUCT" />
    </Principal>
    <Dependent Role="ORDERS">
      <PropertyRef Name="ID_PRODUCTS" />
    </Dependent>
  </ReferentialConstraint>
</Association>
<Association Name="FK__ORDERS__ID_USER__59063A47">
  <End Role="USERS" Type="Self.USER" Multiplicity="0..1" />
  <End Role="ORDERS" Type="Self.ORDER" Multiplicity="*" />
  <ReferentialConstraint>
    <Principal Role="USERS">
      <PropertyRef Name="ID_USER" />
    </Principal>
    <Dependent Role="ORDERS">

```

```

        <PropertyRef Name="ID_USER" />
    </Dependent>
</ReferentialConstraint>
</Association>
<EntityContainer Name="WorkMaticEntities"
annotation:LazyLoadingEnabled="true">
    <EntitySet Name="ADMINS" EntityType="Self.ADMIN" />
    <EntitySet Name="CLIENTS" EntityType="Self.CLIENT" />
    <EntitySet Name="ORDERS" EntityType="Self.ORDER" />
    <EntitySet Name="PRODUCTS" EntityType="Self.PRODUCT" />
    <EntitySet Name="USERS" EntityType="Self.USER" />
    <AssociationSet Name="FK_ADMINS_ID_USER_5441852A"
Association="Self.FK_ADMINS_ID_USER_5441852A">
        <End Role="USERS" EntitySet="USERS" />
        <End Role="ADMINS" EntitySet="ADMINS" />
    </AssociationSet>
    <AssociationSet Name="FK_ORDERS_ID_CLIEN_59FA5E80"
Association="Self.FK_ORDERS_ID_CLIEN_59FA5E80">
        <End Role="CLIENTS" EntitySet="CLIENTS" />
        <End Role="ORDERS" EntitySet="ORDERS" />
    </AssociationSet>
    <AssociationSet Name="FK_ORDERS_ID_PRODU_5AEE82B9"
Association="Self.FK_ORDERS_ID_PRODU_5AEE82B9">
        <End Role="PRODUCTS" EntitySet="PRODUCTS" />
        <End Role="ORDERS" EntitySet="ORDERS" />
    </AssociationSet>
    <AssociationSet Name="FK_ORDERS_ID_USER_59063A47"
Association="Self.FK_ORDERS_ID_USER_59063A47">
        <End Role="USERS" EntitySet="USERS" />
        <End Role="ORDERS" EntitySet="ORDERS" />
    </AssociationSet>
    <FunctionImport Name="GETCLIENT_ORDERS" IsComposable="true"
ReturnType="Collection(WorkMaticModel.GETCLIENT_ORDERS_Result)">
        <Parameter Name="ClientId" Mode="In" Type="Int32" />
    </FunctionImport>
    <EntitySet Name="sysdiagrams" EntityType="WorkMaticModel.sysdiagram" />
    <FunctionImport Name="sp_alterdiagram">
        <Parameter Name="diagramname" Mode="In" Type="String" />
        <Parameter Name="owner_id" Mode="In" Type="Int32" />
        <Parameter Name="version" Mode="In" Type="Int32" />
        <Parameter Name="definition" Mode="In" Type="Binary" />
    </FunctionImport>
    <FunctionImport Name="sp_creatediagram">
        <Parameter Name="diagramname" Mode="In" Type="String" />
        <Parameter Name="owner_id" Mode="In" Type="Int32" />
        <Parameter Name="version" Mode="In" Type="Int32" />
        <Parameter Name="definition" Mode="In" Type="Binary" />
    </FunctionImport>
    <FunctionImport Name="sp_dropdiagram">
        <Parameter Name="diagramname" Mode="In" Type="String" />
        <Parameter Name="owner_id" Mode="In" Type="Int32" />
    </FunctionImport>
    <FunctionImport Name="sp_helpdiagramdefinition"
ReturnType="Collection(WorkMaticModel.sp_helpdiagramdefinition_Result)">
        <Parameter Name="diagramname" Mode="In" Type="String" />
        <Parameter Name="owner_id" Mode="In" Type="Int32" />
    </FunctionImport>
    <FunctionImport Name="sp_helpdiagrams"
ReturnType="Collection(WorkMaticModel.sp_helpdiagrams_Result)">
        <Parameter Name="diagramname" Mode="In" Type="String" />
        <Parameter Name="owner_id" Mode="In" Type="Int32" />

```

```

</FunctionImport>
<FunctionImport Name="sp_renamediagram">
  <Parameter Name="diagramname" Mode="In" Type="String" />
  <Parameter Name="owner_id" Mode="In" Type="Int32" />
  <Parameter Name="new_diagramname" Mode="In" Type="String" />
</FunctionImport>
<FunctionImport Name="sp_upgraddiagrams" />
<FunctionImport Name="CLIENT_ORDERS" IsComposable="true"
ReturnTypes="Collection(WorkMaticModel.CLIENT_ORDERS_Result)">
  <Parameter Name="ClientId" Mode="In" Type="Int32" />
</FunctionImport>
</EntityContainer>
<ComplexType Name="GETCLIENT_ORDERS_Result">
  <Property Type="Int32" Name="ID_ORDERS" Nullable="false" />
  <Property Type="Int32" Name="ID_USER" Nullable="true" />
  <Property Type="Int32" Name="ID_CLIENT" Nullable="true" />
  <Property Type="Int32" Name="ID_PRODUCTS" Nullable="true" />
  <Property Type="Byte" Name="QUANTITY" Nullable="false" />
  <Property Type="DateTime" Name="DATE_ORDERS" Nullable="false" Precision="10"
/>
  <Property Type="Time" Name="TIME_ORDERS" Nullable="false" />
  <Property Type="Int32" Name="ID_PRODUCT" Nullable="false" />
  <Property Type="String" Name="TYPE_PRODUCT" Nullable="false" />
  <Property Type="Decimal" Name="PRICE_PRODUCT" Nullable="false"
Precision="10" Scale="4" />
</ComplexType>
<EntityType Name="sysdiagram">
  <Key>
    <PropertyRef Name="diagram_id" />
  </Key>
  <Property Name="name" Type="String" Nullable="false" MaxLength="128"
FixedLength="false" Unicode="true" />
  <Property Name="principal_id" Type="Int32" Nullable="false" />
  <Property Name="diagram_id" Type="Int32" Nullable="false"
annotation:StoreGeneratedPattern="Identity" />
  <Property Name="version" Type="Int32" />
  <Property Name="definition" Type="Binary" MaxLength="Max"
FixedLength="false" />
</EntityType>
<ComplexType Name="sp_helpdiagramdefinition_Result">
  <Property Type="Int32" Name="version" Nullable="true" />
  <Property Type="Binary" Name="definition" Nullable="true" />
</ComplexType>
<ComplexType Name="sp_helpdiagrams_Result">
  <Property Type="String" Name="Database" Nullable="true" MaxLength="128" />
  <Property Type="String" Name="Name" Nullable="false" MaxLength="128" />
  <Property Type="Int32" Name="ID" Nullable="false" />
  <Property Type="String" Name="Owner" Nullable="true" MaxLength="128" />
  <Property Type="Int32" Name="OwnerID" Nullable="false" />
</ComplexType>
<ComplexType Name="CLIENT_ORDERS_Result">
  <Property Type="Int32" Name="ID_ORDERS" Nullable="false" />
  <Property Type="Int32" Name="ID_USER" Nullable="true" />
  <Property Type="Int32" Name="ID_CLIENT" Nullable="true" />
  <Property Type="Int32" Name="ID_PRODUCTS" Nullable="true" />
  <Property Type="Byte" Name="QUANTITY" Nullable="false" />
  <Property Type="DateTime" Name="DATE_ORDERS" Nullable="false" Precision="10"
/>
  <Property Type="Time" Name="TIME_ORDERS" Nullable="false" />
  <Property Type="Decimal" Name="TOTAL_PRICE" Nullable="true" Precision="10"
Scale="4" />

```



```

        <Property Type="Int32" Name="ID_PRODUCT" Nullable="false" />
        <Property Type="String" Name="TYPE_PRODUCT" Nullable="false" />
        <Property Type="Decimal" Name="PRICE_PRODUCT" Nullable="false"
Precision="10" Scale="4" />
    </ComplexType>
</Schema>
</edmx:ConceptualModels>
<!-- C-S mapping content -->
<edmx:Mappings>
    <Mapping Space="C-S"
xmlns="http://schemas.microsoft.com/ado/2009/11/mapping/cs">
        <EntityContainerMapping StorageEntityContainer="WorkMaticModelStoreContainer"
CdmEntityContainer="WorkMaticEntities">
            <EntitySetMapping Name="ADMINS">
                <EntityTypeMapping TypeName="WorkMaticModel.ADMIN">
                    <MappingFragment StoreEntitySet="ADMINS">
                        <ScalarProperty Name="ID_ADMIN" ColumnName="ID_ADMIN" />
                        <ScalarProperty Name="ID_USER" ColumnName="ID_USER" />
                        <ScalarProperty Name="CAN_READ_USER" ColumnName="CAN_READ_USER" />
                        <ScalarProperty Name="FULL_ACCESS" ColumnName="FULL_ACCESS" />
                    </MappingFragment>
                </EntityTypeMapping>
            </EntitySetMapping>
            <EntitySetMapping Name="CLIENTS">
                <EntityTypeMapping TypeName="WorkMaticModel.CLIENT">
                    <MappingFragment StoreEntitySet="CLIENTS">
                        <ScalarProperty Name="ID_CLIENT" ColumnName="ID_CLIENT" />
                        <ScalarProperty Name="NAME_CLIENT" ColumnName="NAME_CLIENT" />
                        <ScalarProperty Name="ADDRESS_CLIENT" ColumnName="ADDRESS_CLIENT" />
                        <ScalarProperty Name="COD_POST1" ColumnName="COD_POST1" />
                        <ScalarProperty Name="COD_POST2" ColumnName="COD_POST2" />
                        <ScalarProperty Name="LOCALITY" ColumnName="LOCALITY" />
                        <ScalarProperty Name="NIF" ColumnName="NIF" />
                        <ScalarProperty Name="PHONE_1" ColumnName="PHONE_1" />
                        <ScalarProperty Name="PHONE_2" ColumnName="PHONE_2" />
                        <ScalarProperty Name="E_MAIL" ColumnName="E_MAIL" />
                        <ScalarProperty Name="IMAGE_CLIENT" ColumnName="IMAGE_CLIENT" />
                        <ScalarProperty Name="DATE_CLIENT" ColumnName="DATE_CLIENT" />
                        <ScalarProperty Name="CHECK_1" ColumnName="CHECK_1" />
                    </MappingFragment>
                </EntityTypeMapping>
            </EntitySetMapping>
            <EntitySetMapping Name="ORDERS">
                <EntityTypeMapping TypeName="WorkMaticModel.ORDER">
                    <MappingFragment StoreEntitySet="ORDERS">
                        <ScalarProperty Name="TOTAL_PRICE" ColumnName="TOTAL_PRICE" />
                        <ScalarProperty Name="ID_ORDERS" ColumnName="ID_ORDERS" />
                        <ScalarProperty Name="ID_USER" ColumnName="ID_USER" />
                        <ScalarProperty Name="ID_CLIENT" ColumnName="ID_CLIENT" />
                        <ScalarProperty Name="ID_PRODUCTS" ColumnName="ID_PRODUCTS" />
                        <ScalarProperty Name="QUANTITY" ColumnName="QUANTITY" />
                        <ScalarProperty Name="DATE_ORDERS" ColumnName="DATE_ORDERS" />
                        <ScalarProperty Name="TIME_ORDERS" ColumnName="TIME_ORDERS" />
                    </MappingFragment>
                </EntityTypeMapping>
            </EntitySetMapping>
            <EntitySetMapping Name="PRODUCTS">
                <EntityTypeMapping TypeName="WorkMaticModel.PRODUCT">
                    <MappingFragment StoreEntitySet="PRODUCTS">
                        <ScalarProperty Name="ID_PRODUCT" ColumnName="ID_PRODUCT" />
                        <ScalarProperty Name="TYPE_PRODUCT" ColumnName="TYPE_PRODUCT" />
                    </MappingFragment>
                </EntityTypeMapping>
            </EntitySetMapping>
        </EntityContainerMapping>
    </Mapping>
</edmx:Mappings>

```

```

        <ScalarProperty Name="PRICE_PRODUCT" ColumnName="PRICE_PRODUCT" />
    </MappingFragment>
</EntityTypeMapping>
</EntitySetMapping>
</EntitySetMapping>
<EntitySetMapping Name="USERS">
    <EntityTypeMapping TypeName="WorkMaticModel.USER">
        <MappingFragment StoreEntitySet="USERS">
            <ScalarProperty Name="ID_USER" ColumnName="ID_USER" />
            <ScalarProperty Name="NAME_USER" ColumnName="NAME_USER" />
        </MappingFragment>
    </EntityTypeMapping>
</EntitySetMapping>
<EntitySetMapping Name="sysdiagrams">
    <EntityTypeMapping TypeName="WorkMaticModel.sysdiagram">
        <MappingFragment StoreEntitySet="sysdiagrams">
            <ScalarProperty Name="definition" ColumnName="definition" />
            <ScalarProperty Name="version" ColumnName="version" />
            <ScalarProperty Name="diagram_id" ColumnName="diagram_id" />
            <ScalarProperty Name="principal_id" ColumnName="principal_id" />
            <ScalarProperty Name="name" ColumnName="name" />
        </MappingFragment>
    </EntityTypeMapping>
</EntitySetMapping>
<FunctionImportMapping FunctionImportName="sp_alterdiagram"
FunctionName="WorkMaticModel.Store.sp_alterdiagram" />
<FunctionImportMapping FunctionImportName="sp_creatediagram"
FunctionName="WorkMaticModel.Store.sp_creatediagram" />
<FunctionImportMapping FunctionImportName="sp_dropdiagram"
FunctionName="WorkMaticModel.Store.sp_dropdiagram" />
<FunctionImportMapping FunctionImportName="sp_helpdiagramdefinition"
FunctionName="WorkMaticModel.Store.sp_helpdiagramdefinition">
    <ResultMapping>
        <ComplexTypeMapping
TypeName="WorkMaticModel.sp_helpdiagramdefinition_Result">
            <ScalarProperty Name="version" ColumnName="version" />
            <ScalarProperty Name="definition" ColumnName="definition" />
        </ComplexTypeMapping>
    </ResultMapping>
</FunctionImportMapping>
<FunctionImportMapping FunctionImportName="sp_helpdiagrams"
FunctionName="WorkMaticModel.Store.sp_helpdiagrams">
    <ResultMapping>
        <ComplexTypeMapping TypeName="WorkMaticModel.sp_helpdiagrams_Result">
            <ScalarProperty Name="Database" ColumnName="Database" />
            <ScalarProperty Name="Name" ColumnName="Name" />
            <ScalarProperty Name="ID" ColumnName="ID" />
            <ScalarProperty Name="Owner" ColumnName="Owner" />
            <ScalarProperty Name="OwnerID" ColumnName="OwnerID" />
        </ComplexTypeMapping>
    </ResultMapping>
</FunctionImportMapping>
<FunctionImportMapping FunctionImportName="sp_renamediagram"
FunctionName="WorkMaticModel.Store.sp_renamediagram" />
<FunctionImportMapping FunctionImportName="sp_upgraddiagrams"
FunctionName="WorkMaticModel.Store.sp_upgraddiagrams" />
<FunctionImportMapping FunctionImportName="CLIENT_ORDERS"
FunctionName="WorkMaticModel.Store.CLIENT_ORDERS">
    <ResultMapping>
        <ComplexTypeMapping TypeName="WorkMaticModel.CLIENT_ORDERS_Result">
            <ScalarProperty Name="ID_ORDERS" ColumnName="ID_ORDERS" />
            <ScalarProperty Name="ID_USER" ColumnName="ID_USER" />
        </ComplexTypeMapping>
    </ResultMapping>
</FunctionImportMapping>

```

```

        <ScalarProperty Name="ID_CLIENT" ColumnName="ID_CLIENT" />
        <ScalarProperty Name="ID_PRODUCTS" ColumnName="ID_PRODUCTS" />
        <ScalarProperty Name="QUANTITY" ColumnName="QUANTITY" />
        <ScalarProperty Name="DATE_ORDERS" ColumnName="DATE_ORDERS" />
        <ScalarProperty Name="TIME_ORDERS" ColumnName="TIME_ORDERS" />
        <ScalarProperty Name="TOTAL_PRICE" ColumnName="TOTAL_PRICE" />
        <ScalarProperty Name="ID_PRODUCT" ColumnName="ID_PRODUCT" />
        <ScalarProperty Name="TYPE_PRODUCT" ColumnName="TYPE_PRODUCT" />
        <ScalarProperty Name="PRICE_PRODUCT" ColumnName="PRICE_PRODUCT" />
    </ComplexTypeMapping>
</ResultMapping>
</FunctionImportMapping>
</EntityContainerMapping>
</Mapping>
</edmx:Mappings>
</edmx:Runtime>
<!-- EF Designer content (DO NOT EDIT MANUALLY BELOW HERE) -->
<Designer xmlns="http://schemas.microsoft.com/ado/2009/11/edmx">
    <Connection>
        <DesignerInfoPropertySet>
            <DesignerProperty Name="MetadataArtifactProcessing"
Value="EmbedInOutputAssembly" />
        </DesignerInfoPropertySet>
    </Connection>
    <Options>
        <DesignerInfoPropertySet>
            <DesignerProperty Name="ValidateOnBuild" Value="true" />
            <DesignerProperty Name="EnablePluralization" Value="true" />
            <DesignerProperty Name="IncludeForeignKeysInModel" Value="true" />
            <DesignerProperty Name="UseLegacyProvider" Value="false" />
            <DesignerProperty Name="CodeGenerationStrategy" Value="None" />
        </DesignerInfoPropertySet>
    </Options>
    <!-- Diagram content (shape and connector positions) -->
    <Diagrams></Diagrams>
</Designer>
</edmx:Edmx>

```

