



**Instituto Superior de Tecnologias Avançadas**

**Licenciatura Informática**

## **Aplicação de Biblioteca Eletrónica Projeto Omega**

**Realizado por João Carlos Miranda Roque Diogo – Nº 1801**

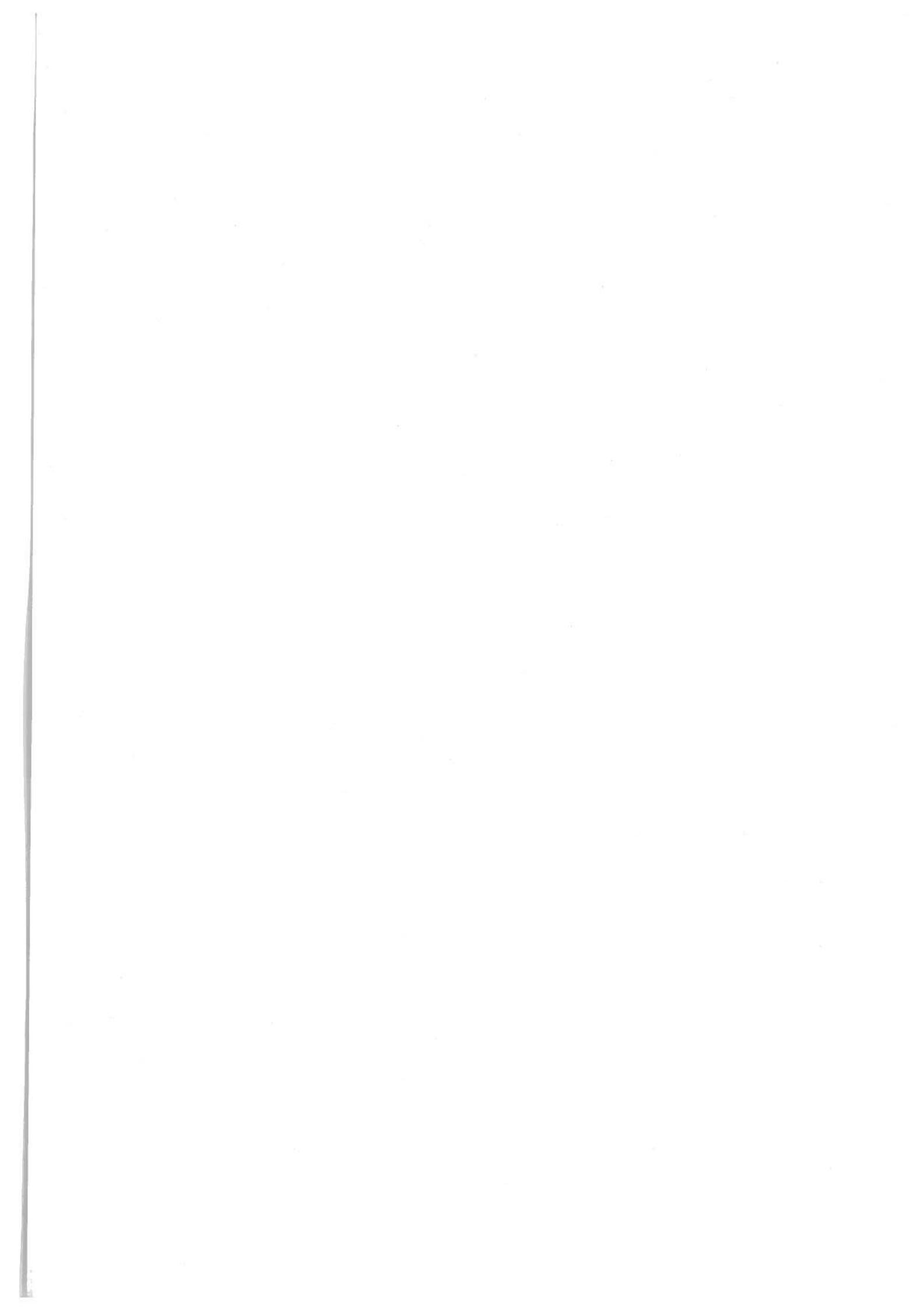
Orientador: Professor Doutor Pedro Ramos Brandão

Orientador Metodológico: Professor Doutor Pedro Ramos Brandão

**Lisboa**

**2014/2015**

**Instituto Superior de Tecnologias Avançadas**



Licenciatura em Informática

Projeto Global

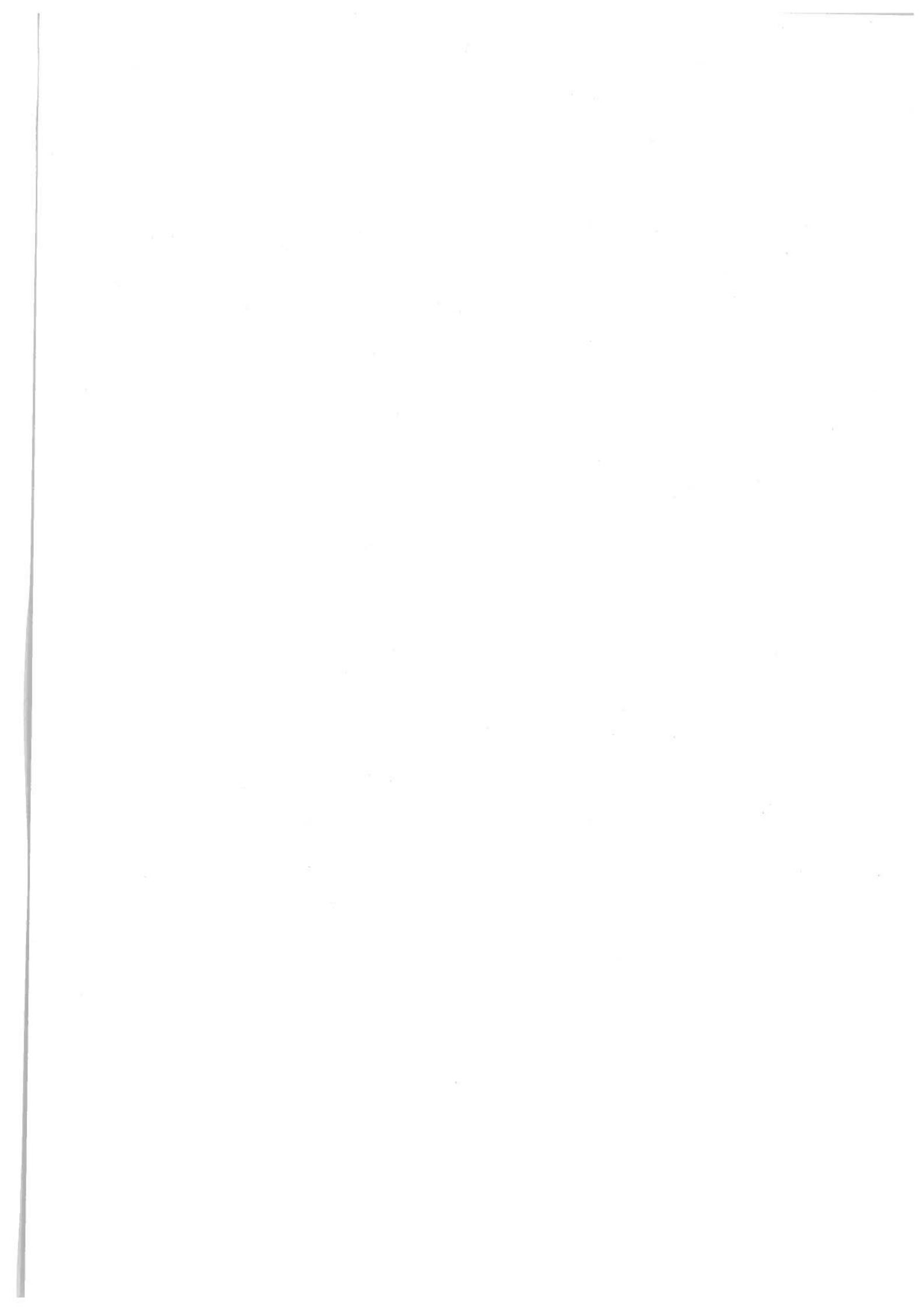
Professor Doutor Pedro Ramos Brandão

**Aplicação de Biblioteca Eletrónica**  
**Projeto Omega**

João Carlos Miranda Roque Diogo

Novembro

2015



## **Resumo**

Na atual sociedade de conhecimento e informação, a necessidade de aceder a conteúdos, informação ou mesmo entretenimento de uma forma rápida, como que "sem fios agarrados", tornou-se um requisito mandatório. Para tal, essa informação é disponibilizada por um conjunto de Sistemas de Informação, que tem vindo a ganhar relevância, sendo atualmente encarados como *Assets* fulcrais na estratégia comercial e de mercado das várias entidades que delas se aproveitam

Fazem parte desses Sistemas de Informação, as Tecnologias de Informação que são constituídas por aplicações que facultam os serviços prestados por uma entidade ou organização. Estes serviços são disponibilizados de forma eficiente, tendo como principal objetivo o enriquecimento das necessidades de um utilizador para lhe prestar um serviço, que, para além de fazer face as suas necessidades, supere as suas expectativas. (Figueiredo Rui, 2013)

Para uma organização ou entidade disponibilizar os seus serviços de forma mais eficaz, é imperativo que detenha uma plataforma de provisionamento de conteúdos com a qual possa disponibilizar os serviços que detenha criando assim um canal de comunicação com o utilizador onde este pode usufruir dos mesmos de forma intuitiva.

**Palavras-chave:** *e-book, AD.*

## **Abstract**

In this society of knowledge and information, the need to access content, information or even entertainment quickly, as if with "no strings attached" became a main drive. To this end, the information it's made available by a group of Information Systems, which gained major importance and is currently regarded as a key asset in the commercial strategy and market of the various entities that take advantage of them.

Part of the Information Systems, the Information Technologies that are constituted by applications that provide the services to the entity or organization. These services are provided efficiently, with the primary aim of the enrichment needs of a user to provide his service, which, in addition to address their needs, exceed their expectations.

For an organization or entity providing their more effectively services, it is imperative that holds a content provisioning platform with which to provide the services holding thus creating a communication channel with the user where it can enjoy the same in an intuitive form.

**Keywords:** *e-book, AD.*

## **Abreviaturas**

**AD - Active Directory**

**IaaS - Infrastructure as a Service**

**PaaS - Platform as a Service**

**SaaS - Software as a Service**

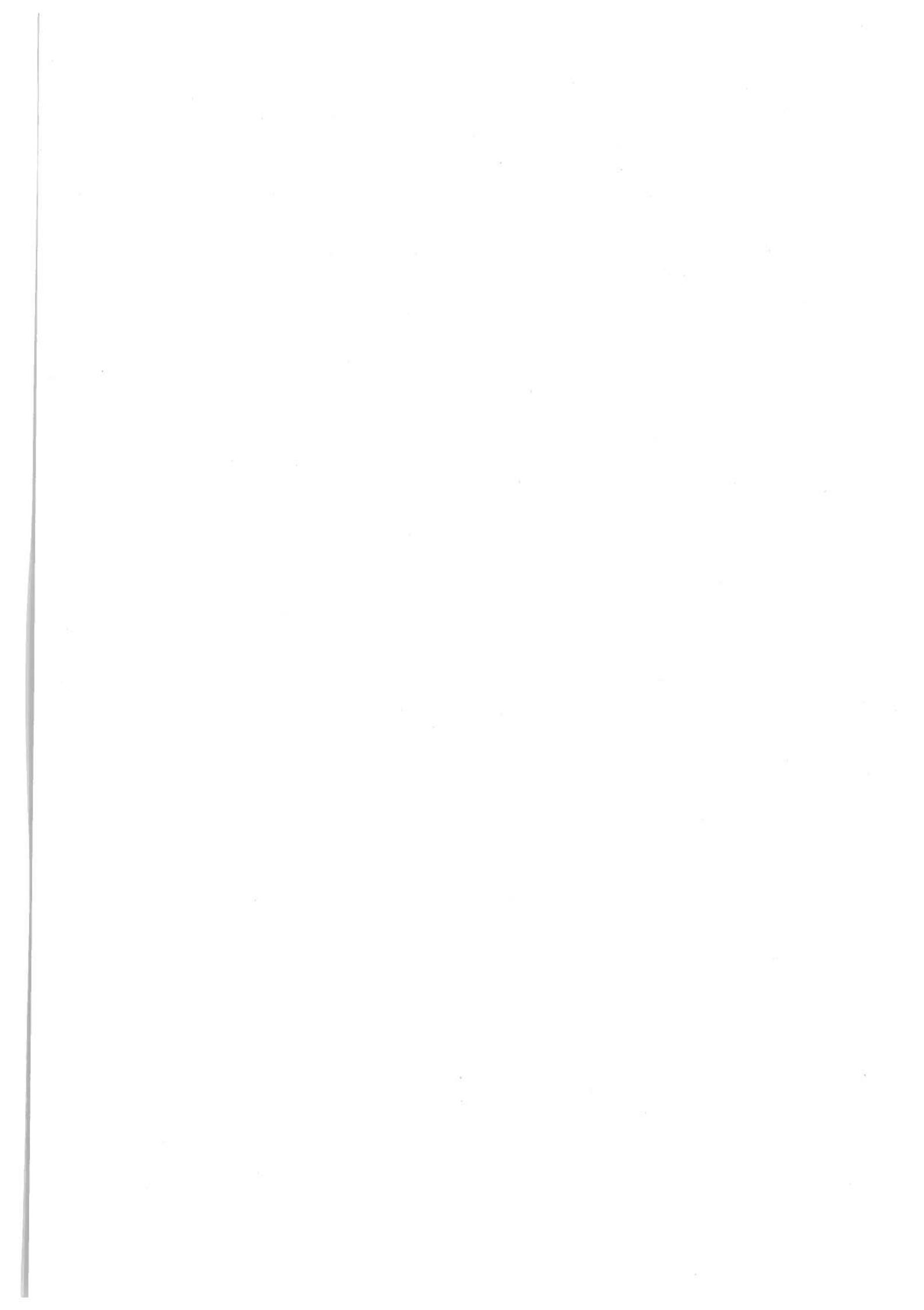
**SI - Sistemas de informacao**

**SQL – Structured Query Language**

**T-SQL – Transact Structured Query Language**

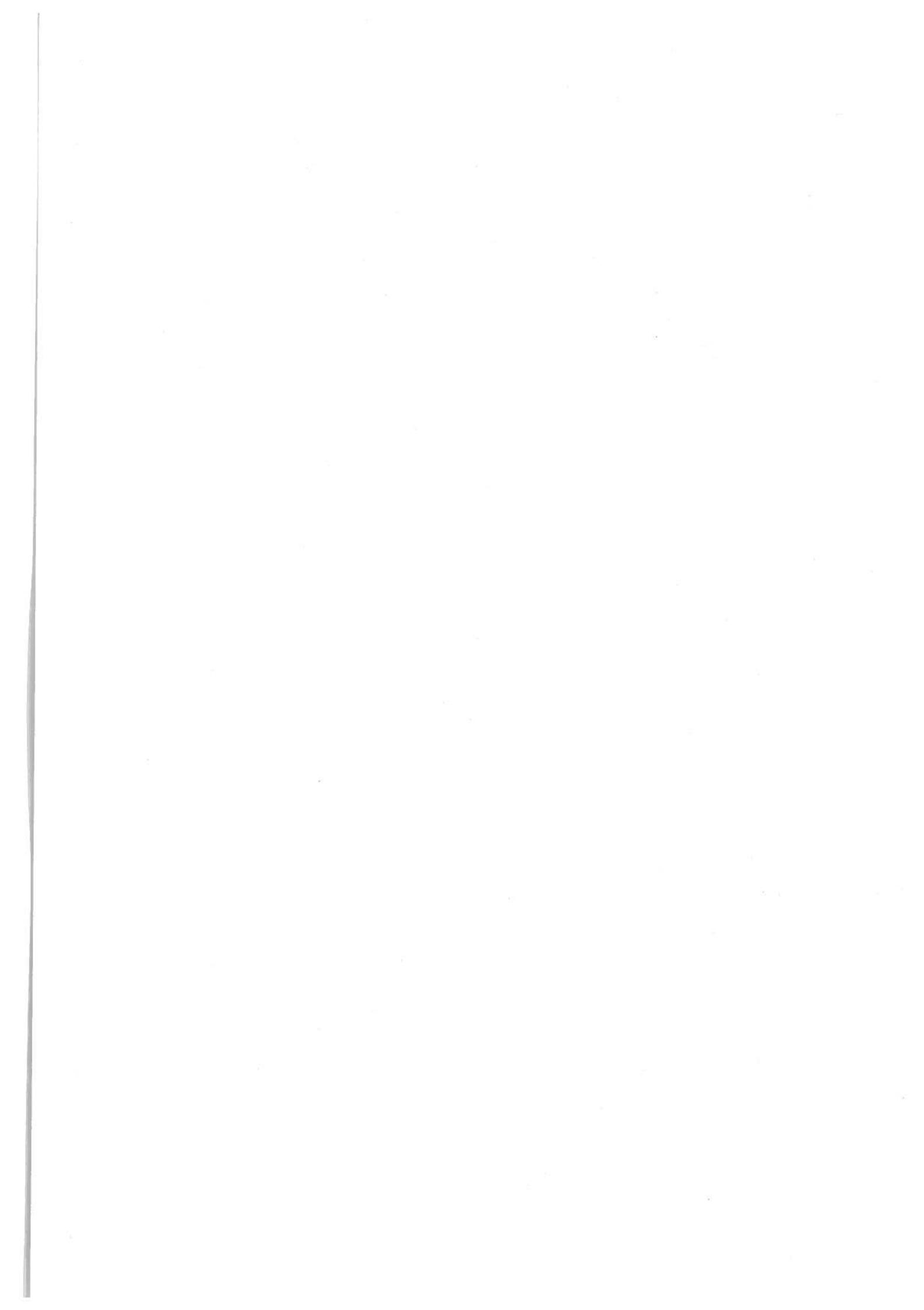
**TI – Tecnologia de Informação**

**URL – Uniform Resource Locator**



# ÍNDICE

Introdução.....	13
Estado de Arte .....	15
1- SQL Server .....	15
2- SMO – SQL Management Objects.....	16
3 - .NET Framework.....	18
4 - Virtualização .....	19
5 - Cloud Computing .....	21
6 - Tipos de virtualização .....	22
Contextualização .....	25
Desenvolvimento .....	27
1- Architecture and Requisite Gathering .....	28
2- Desenvolvimento empírico da aplicação .....	30
Testes de usabilidade .....	69
Conclusão .....	71
Referências Bibliográficas.....	73



# ÍNDICE DE FIGURAS E GRÁFICOS

Figure 1 – Databases.....	15
Figure 2 – SMO .....	17
Figure 3 - .NETFRAMEWORK.....	18
Figure 4 - .NET FRAMEWORK COMPONENTS .....	19
Figure 5 – VIRTUALIZAÇÃO .....	20
Figure 6 –VIRTUALIZATION COMPONENTS .....	21
Figure 7 – CLOUD COMPUTING .....	22
Figure 8 – TYPES OF VIRTUALIZATION .....	23
Figure 9 –Sample Cloud Decision Framework.....	24
Figure 10 -Diagrama da arquitectura Logica .....	29
Figure 11 - library.....	31
Figure 12 - Variaveis da aplicacao .....	31
Figure 13 - Focus.....	32
Figure 14 - Maxlenght list .....	32
Figure 15 - RESET CONTRO LIST.....	32
Figure 16 - ENABLE/DISABLE CONTROL LIST.....	33
Figure 17 - Error Handling .....	33
Figure 18 - ToolTip .....	33
Figure 19 - Image Sizing .....	34
Figure 20 - Character counter .....	34
Figure 21 - Panel.....	35
Figure 22 - Panel control .....	35
Figure 23 – E-book parameters.....	36
Figure 24 - Grid init.....	36
Figure 25 – Grid row Data.....	37
Figure 26 - grid command .....	38
Figure 27 - Image array .....	38
Figure 28 - Book Find .....	39
Figure 29 - Alphabar Event .....	39
Figure 30 - Create button.....	40
Figure 31 - Alphabar.....	40
Figure 32 - Alphabar trigger.....	41
Figure 33 - App Close .....	41
Figure 34 - Add book.....	41
Figure 35 - Cancel book .....	42
Figure 36 - Form Load.....	42
Figure 37 - Autor event .....	43
Figure 38 - Control focus event .....	43
Figure 39 - cover load.....	44
Figure 40 - file check .....	44
Figure 41 - check content .....	45
Figure 42 - Book save.....	45
Figure 43 - book class.....	46
Figure 44 - Dataitem class .....	47
Figure 45 - book delete .....	48
Figure 46 - book get by id.....	49
Figure 47 - book get.....	50
Figure 48 - book update or insert.....	51

Figure 49 - adoparameter class .....	52
Figure 50 - adostoredprocedure class .....	53
Figure 51 - parameter load.....	53
Figure 52 - get data .....	54
Figure 53 - create data .....	55
Figure 54 - user class .....	55
Figure 55 - ad authentication .....	56
Figure 56 - user validation .....	57
Figure 57 - table properties.....	58
Figure 58 - procedure del book by id.....	58
Figure 59 - procedure get book by id.....	59
Figure 60 - procedure update book .....	60
Figure 61 - procedure get book by id.....	60
Figure 62 - books table .....	61
Figure 63 - Frontend .....	61
Figure 64 - code behind .....	62
Figure 65 - filtered view .....	62
Figure 66 - book view .....	63
Figure 67 - e-book view.....	64
Figure 68 - book removal .....	64
Figure 69 - remove success message .....	65
Figure 70 - alpha bar.....	66
Figure 71 - add book view .....	66
Figure 72 - filter view .....	67
Figure 73 - search type view.....	67
Figure 74 - app exit.....	68
Figure 75 - etapas de testes ( Kekare Harshada, <a href="http://www.buzzle.com/ 2013">http://www.buzzle.com/ 2013</a> ).....	69
Figure 76 – Tabela de testes .....	70

## INTRODUÇÃO

O presente trabalho é realizado no âmbito da unidade curricular anual de Projeto Global do curso de Informática, lecionado pelo Instituto Superior de Tecnologias Avançadas.

Este trabalho tem um conteúdo de relatório Técnico-Científico, sendo que a área científica apresentada no mesmo tem como base as Ciências da Programação, sendo que a escolha do tema passou por uma orientação por parte do Coordenador do projeto, O Professor Doutor Pedro Brandão.

A realização deste trabalho tem como objetivo o desenvolvimento de uma Aplicação que permita pesquisa e acesso de utilizadores de uma biblioteca a um conjunto de informações previamente disponibilizadas relativamente a conteúdos bibliográficos.

O objetivo específico é a elaboração de uma plataforma informática que permita ao utilizador procurar e aceder, de forma rápida, eficiente e intuitiva ao conteúdo de uma biblioteca, armazenada de uma forma digital num Sistema informático.

Após uma pesquisa "de mercado" entre diversas entidades privadas e públicas, quer na área mais comum de Bibliotecas, quer de sistemas de utilização doméstica para organização de conteúdos digitais, foi possível identificar um abundante universo de abordagens para o nosso objetivo.

A metodologia usada para abordar este trabalho, passou por uma pesquisa bibliográfica sobre os temas a analisar e que foram usados de seguida para implementar o projeto na sua componente teórica, efetuando uma revisão em diversos temas tais como, .Net, Bases de dados, Sistemas de gestão bibliográficos, etc.

Este levantamento de informação será apresentado no Estado de Arte e seguidamente, será apresentada a arquitetura considerada necessária para a implementação do projeto, bem como requisitos aplicacionais e funcionais, tais como usabilidade, permissões e manipulação de dados e desenho do "front-end".

No seguimento, é apresentado o desenvolvimento da aplicação. Neste foram usados o Microsoft Visual Studio para desenvolver o "front-end" e o SQL Server que serve o "back-end" da nossa aplicação, onde se faz a introdução e armazenamento da informação propriamente dita.

Após a conclusão destas etapas, será produzido um conjunto de testes de utilização e regressão na aplicação de forma a validar que os objetivos propostos são atingidos.

O presente projeto foi dividido em três grandes etapas, que iremos descrever de seguida:

- Pesquisa bibliográfica de forma a executar o Estado de Arte.

Esta fase conduz a descriminação de forma detalhada de conceitos utilizados no desenvolvimento da aplicação. Estes conceitos passam pelo conceito de .Net, Virtualização e serviços "Cloud". Este recolher de informação é fundamentado através de múltiplas fontes, sendo que as mais óbvias são artigos Web ou literatura técnica.

- Desenvolvimento da aplicação.

Onde são apresentados todos os passos necessários para a conclusão do projeto e as ferramentas usadas no mesmo.

- Testes de usabilidade e regressão

Executados contra a versão final da aplicação desenvolvida.

Para concluir o trabalho, iremos ainda apresentar um *post-mortem* dos objetivos atingidos bem como as dificuldades encontradas e os ganhos retirados da realização do mesmo.

## ESTADO DE ARTE

Neste capítulo iremos abordar os componentes usados na aplicação dando uma breve explicação dos mesmos.

### 1- SQL SERVER

O SQL Server é um sistema de gestão de base de dados relacional (RDBMS) da Microsoft, que é projetado para um ambiente corporativo. O SQL Server é executado em T- SQL (Transact-SQL), um conjunto de extensões de programação da Sybase e da Microsoft que adicionou vários recursos ao SQL padrão, incluindo o controle de transação, exceção e tratamento de erros e declaração variáveis. O código original do SQL Server foi desenvolvido pela Sybase.

No final de 1980 a Microsoft a Sybase e a Ashton-Tate colaboraram para produzir a primeira versão do produto, o SQL Server 4.2 para OS/2. Posteriormente, ambos passaram a oferecer produtos SQL. A Microsoft o MSSQL e a Sybase renomeou na altura o seu produto para Adaptive Server Enterprise. (Janssen Cory, 2012)



FIGURE 1 – DATABASES

Davies, Pete (Dezembro, 2013) <http://www.anrcg.com>

O SQL Server utiliza um conceito de objetos para representar as bases de dados e todos os seus elementos e configurações. Estes objetos ficam armazenados no catálogo de sistema.

Para aceder a tais informações, existem *views* padrões do SGBD, onde as de catálogo são consideradas o interface mais geral dessas informações. (Molinaro Anthony, 2005)

## 2- SMO – SQL MANAGEMENT OBJECTS

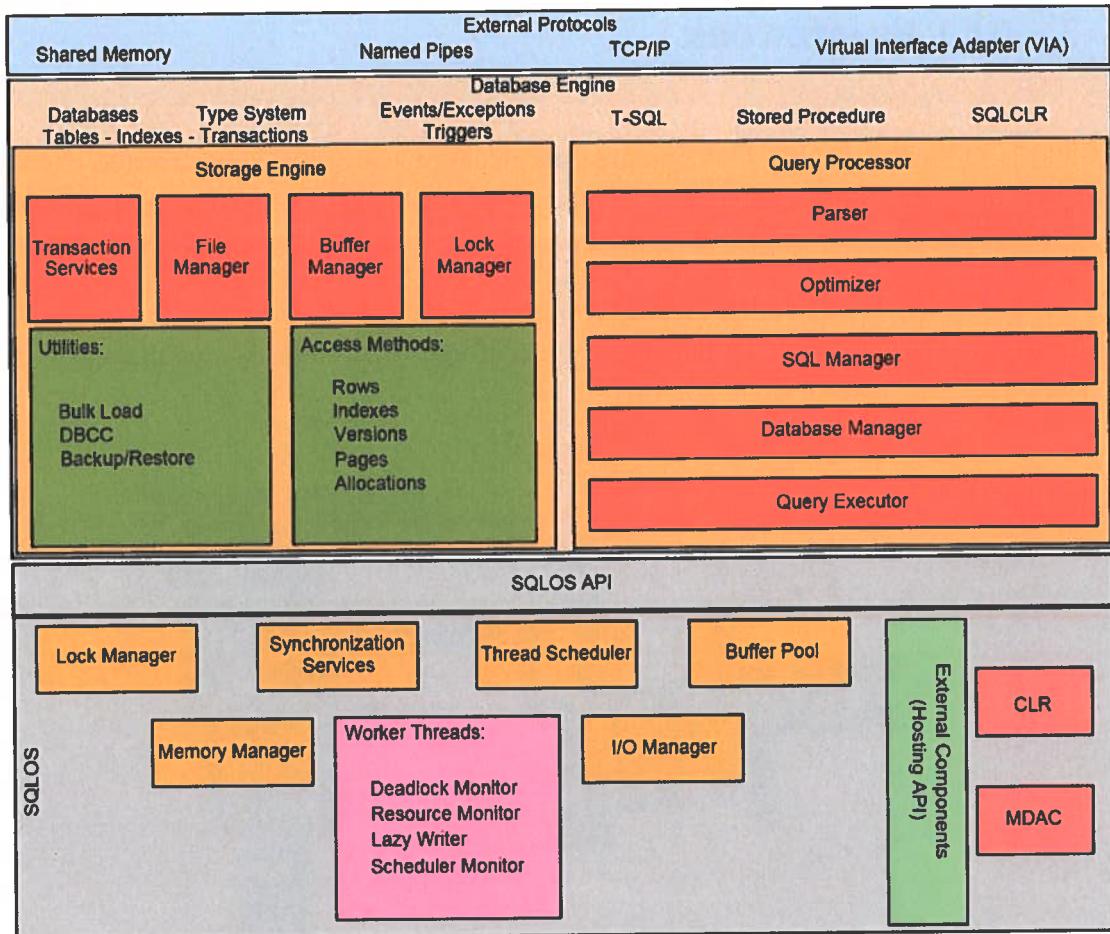
A Microsoft desenvolveu uma biblioteca para a plataforma .Net com funcionalidades para manipulação de bases de dados SQL Server. Essa biblioteca é chamada de SQL Server Management Objects (SMO).

O principal objetivo dessa biblioteca é obter e alterar as informações de estrutura de base de dados através das linguagens de programação tais como o Visual Basic .Net e C# .Net. Com isso, é possível desenvolver *software* que manipule os esquemas de bases de dados dinamicamente, sem que o programador precise criar *scripts* em SQL.

Essa biblioteca foi desenvolvida tendo em vista um melhor desempenho quanto à alocação de memória do servidor em tempo de execução. Para tal, os objetos são primeiramente instanciados de forma parcial, obtendo uma quantidade mínima de informações do servidor da base de dados. Somente quando uma determinada propriedade não contemplada na instância parcial do objeto é requisitada ou quando uma chamada de método depender dessa propriedade, o objeto é totalmente instanciado. (Rankins Ray, 2013)

O SMO possui duas categorias de classes:

- Classes de instância, as quais representam os objetos da base de dados que fazem o acesso às informações necessárias para as funcionalidades de sincronização de bases de dados.
- Classes utilitárias, que executam funcionalidades específicas na base de dados.



**FIGURE 2 – SMO**  
Vasch, Soum (dezembro 2007) <http://commons.wikimedia.org>

Os objetos SMO são relacionados de forma hierárquica, onde pode existir um pai para vários filhos, agrupados em coleções de objetos; e um pai para um único filho, este representado por um objeto apenas. Um exemplo seria uma tabela possuir uma ou várias colunas.

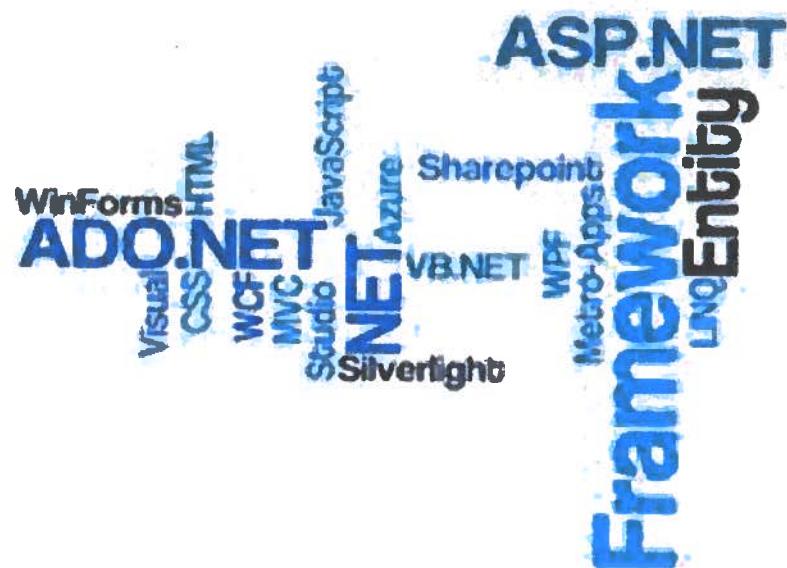
As classes destinadas ao encapsulamento dos meta-dados fazem parte de um conjunto de classes chamado Microsoft.SqlServer.Management.Smo, o qual está contido no arquivo Microsoft.SqlServer.Smo.dll. Cada tipo de objeto da base de dados possui classes para a sua representação. A criação de objetos torna-se assim mais simples quando se utiliza SMO.

Para criar uma tabela na base de dados, por exemplo, basta instanciar a classe “Table” associando- se um nome a ela. Após a criação do objeto referente à tabela, utiliza-se a classe “Column” para criar as suas respetivas colunas. (Rankins Ray, 2013)

### 3 - .NET FRAMEWORK

O Microsoft .NET Framework é uma plataforma que fornece ferramentas e tecnologias que o utilizador precisa para construir aplicações distribuídas, assim como *web services* e aplicações *web*.

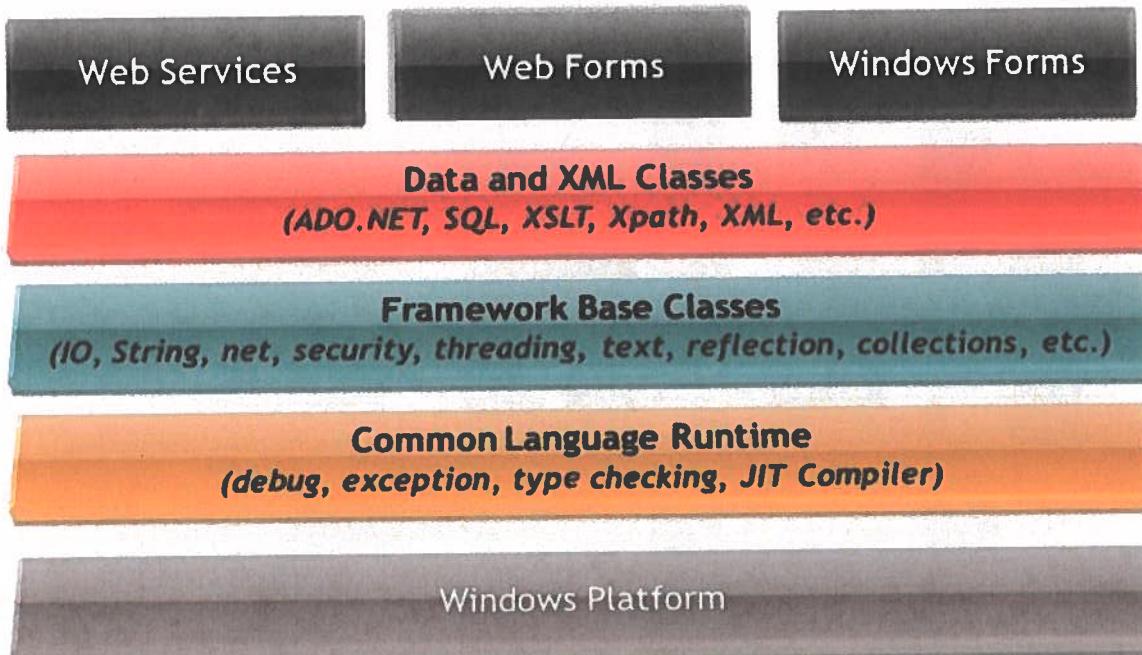
O .NET Framework fornece a ferramenta de “runtime” e de compilação necessária para construir e executar qualquer linguagem em conformidade com a Common Language Specification (CLS). (Janssen Cory, 2012)



**FIGURE 3 - .NETFRAMEWORK**  
Kumar, Vivek (Junho, 2013) <http://www.cpd-india.com>

Os dois principais componentes do Framework .Net são:

- Common Language Runtime (CLR)
- .Net Framework Class Library (FCL).



**FIGURE 4 - .NET FRAMEWORK COMPONENTS**  
 Arnold, Roy (Março, 2008) <https://middlewares.files.wordpress.com>

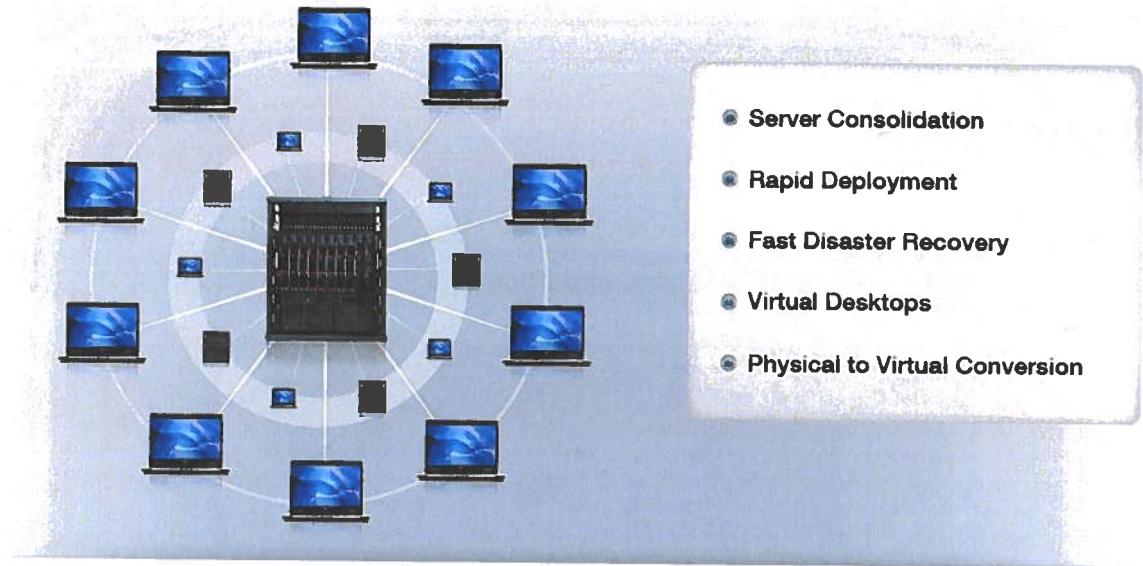
O Microsoft .NET é um componente de *software* que faz parte do sistema operativo Microsoft Windows. O mesmo tem uma vasta biblioteca de soluções pré-construída para os requisitos de programas comuns, e gere a execução de programas escritos especificamente para a “framework”. O .NET Framework é peça base de oferta Microsoft, e destina-se a ser usado pela maioria das novas aplicações criadas para a plataforma Windows. (Chappell David, 2006)

## 4 - VIRTUALIZAÇÃO

A virtualização refere-se à criação de um recurso virtual tal como um servidor, área de trabalho, sistema operativo, arquivo, armazenamento ou rede. (Janssen Cory, 2012)

O principal objetivo da virtualização é gerir cargas de trabalho, transformando radicalmente a computação tradicional de forma a torná-la mais escalável. A virtualização tem sido uma parte do panorama de TI há várias décadas, e hoje pode ser aplicada a uma ampla gama de camadas de sistema, tais como: (Daly Jimmy, 2013)

- Ao nível de sistema operativo,
- Ao nível de *hardware*
- De servidores.



**FIGURE 5 – VIRTUALIZAÇÃO**  
Oneil, Collin <http://www.tthree.org>

A forma mais comum de virtualização é a virtualização ao nível do sistema operativo. Neste tipo de virtualização, é possível executar vários sistemas operativos num único servidor físico. A tecnologia de virtualização envolve separar o *hardware* (a sua parte física) e o *software* (parte aplicacional), emulando uma camada de *hardware*.

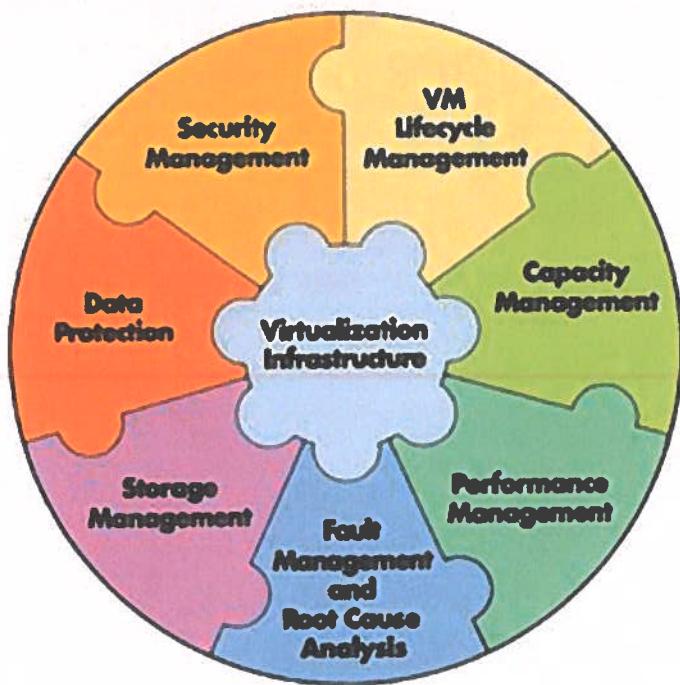
Quando um sistema operativo diferente está a funcionar no topo do sistema operativo primário por meio de virtualização, é referida como uma máquina virtual. Uma máquina virtual é nada mais de que um objeto de dados num servidor físico que pode ser movido e copiado para outro servidor, assim como um arquivo de dados normal.

Os computadores num ambiente virtualizado usam dois tipos de estruturas de ficheiros: uma definição do *hardware* e outra de definição de disco rígido. O *software* de virtualização, ou o “hyper visor”, oferece tecnologia de cache que pode ser usado para armazenar em “cache” as alterações de *hardware* virtual ou no disco rígido virtual para escrever num momento posterior. Esta tecnologia permite que um utilizador descarte as alterações feitas ao sistema operativo, permitindo-lhe arrancar a partir de um estado previamente conhecido. (Portnoy Matthew, 2012)

A virtualização pode ser categorizada em diferentes camadas: *desktop*, servidor, arquivo, armazenamento e rede.

Cada camada de virtualização tem seu próprio conjunto de vantagens e complexidades. A tecnologia oferece muitos benefícios, incluindo a implementação de baixo custo, utilização completa de recursos, redução de custos operacionais e de economia de energia.

No entanto, a implementação de tecnologias de virtualização requer um planeamento cuidadoso uma vez que as máquinas virtuais utilizam os mesmos recursos ao serem executadas, o que pode conduzir a diminuição de desempenho. (Kusnetzky Dan, 2011)



**FIGURE 6 –VIRTUALIZATION COMPONENTS**  
Floersch, Doug (Dezembro, 2013) <http://www.tgs-kc.com>

## 5 - CLOUD COMPUTING

Uma nuvem refere-se a um ambiente de “TI” distinto que é desenvolvido para efeitos de aprovisionamento remoto, escalável e facilmente mensurável a nível de recursos “TI”. O termo originou-se como uma metáfora para a “Internet”, que é, em essência, uma rede de redes que fornecem acesso remoto a um conjunto de recursos de “TI” descentralizados.

Antes da computação em nuvem se tornar seu próprio segmento formalizado da indústria de “TI”, o símbolo de uma nuvem era habitualmente usado para representar a “Internet” em uma variedade de especificações e documentação regular de arquiteturas baseadas na Web.

Este mesmo símbolo é agora usado para representar especificamente o contorno de um ambiente de “Cloud Computing”. (Vangie Beal, 2014)



**FIGURE 7 – CLOUD COMPUTING**  
Nag, Rahul (Agosto 2013) <http://thetechstuffs.com>

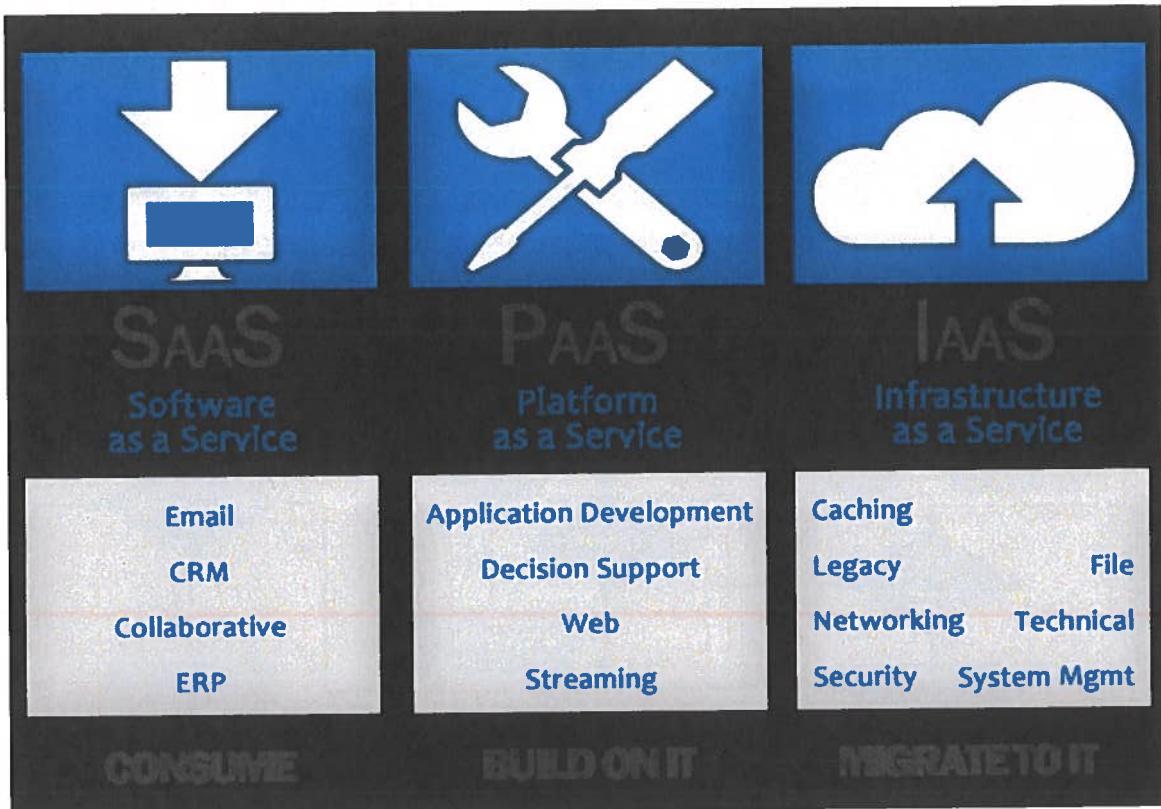
Grande parte da “Internet” é dedicada ao acesso dos recursos com base em conteúdo de “TI” publicado através da *World Wide Web*. Recursos de “TI” fornecidos por ambientes de nuvem, por outro lado, são dedicados a fornecer capacidade de processamento *back-end* e sua disponibilização a utilizadores para esses mesmos recursos.

Outra distinção fundamental é que ela não é necessariamente baseada na *Web*, mesmo sendo habitualmente baseada em protocolos e tecnologias da Internet.

Os protocolos são padrões e métodos que permitem que os computadores para comunicar uns com outros de uma maneira pré-definida e estruturada. Uma nuvem pode ser baseada no uso de quaisquer protocolos que permitem o acesso remoto aos recursos de “TI”. (Woodford Chris, 2014)

## **6 - TIPOS DE VIRTUALIZAÇÃO**

Para além das definições habituais de “Public Cloud” e “Private Cloud”, devemos também referir os três grandes tipos de nuvem aplicacional e de serviço.

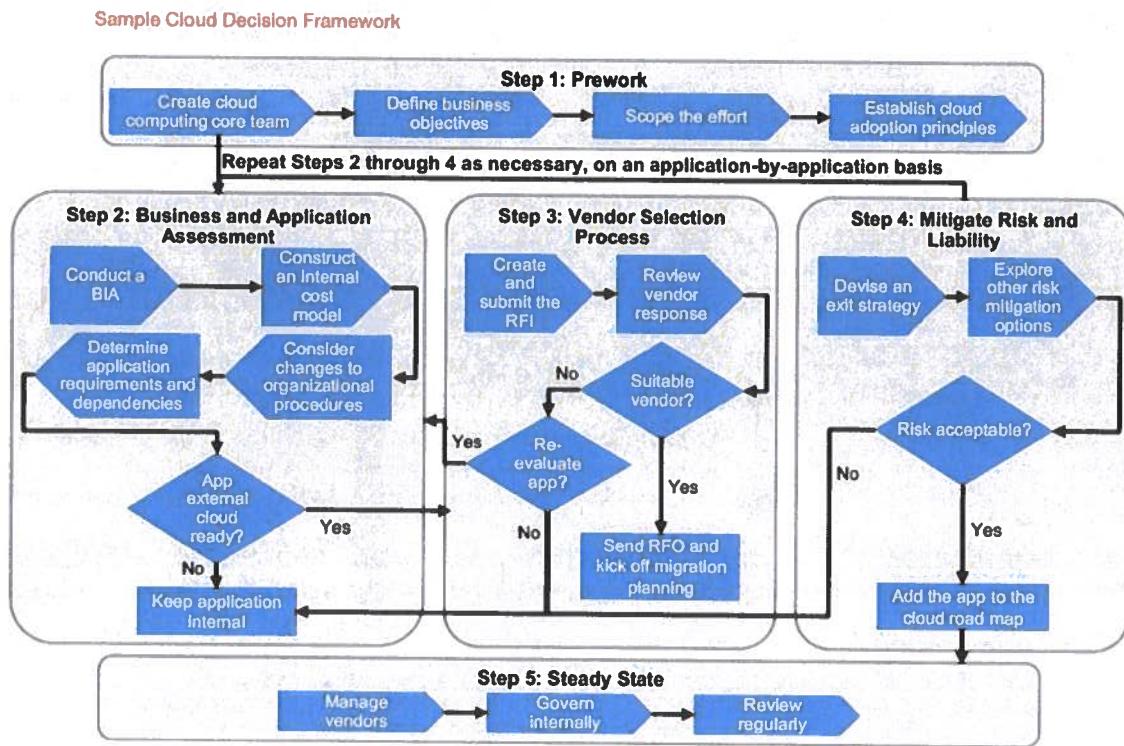


**FIGURE 8 – TYPES OF VIRTUALIZATION**

Preciado, Trigueros (Março, 2013) <http://cloudprivado.org>

São eles:

- Infrastructure as a Service (IaaS) significa que se adquire o acesso a recursos em bruto através da Net, recursos tais como Servidores ou “Storage”. Como são adquiridos como base na necessidade imediata são também conhecidos como “Utility Computing”. O exemplo mais comum são os serviços de “Web Hosting” onde se paga uma subscrição mensal ou por megabyte/gigabyte para uma empresa servir o *file hosting* para o *site*.
- Software as a Service (SaaS) significa que estamos a adquirir uma aplicação completa a correr num servidor remoto que não nos pertence. Sistemas de *email* “Web-Based” ou o Google Documents são dos exemplos mais comuns para este tipo de nuvem.
- Platform as a Service (PaaS) significa que são desenvolvidos pelo cliente, aplicações, através do uso de um conjunto de ferramentas “Web-Based” que por sua vez vão correr em cima de *hardware* e *software* disponibilizados por outra companhia. Um bom exemplo deste tipo de implementação são as plataformas de e-commerce que por sua vez correm nas “frameworks” de algo como o Force.com (Salesforce) ou mesmo o Google App Engine. (Kavis Michael J., 2014), (Rhoton John, 2009)



**FIGURE 9 –SAMPLE CLOUD DECISION FRAMEWORK**  
Garther (Outubro 2012) <http://www.forbes.com>

A imagem acima mostra um possível *workflow* a ser usado no processo de decisão de utilização de “Cloud Computing”.

Nela podemos verificar os principais passos que devem ser efetuados no momento da decisão, sendo eles: (Bahga Arshdeep ,Madisetti Vijay , 2013)

- Custo da solução
- Analisar os processos internos que devem ser alterados
- Requisitos aplicacionais e dependências entre as mesmas
- Análise de propostas de fornecedores
- Efetuar plano de migração
- Efetuar análise de risco

## CONTEXTUALIZAÇÃO

O tema deste trabalho foi-nos apresentado como potencial escolha por parte do Orientador do Projeto Global, o Professor Doutor Pedro Brandão. Após um período de introspeção decidimos avançar com o tema apresentado.

Seguiu-se um período de recolha de exemplos onde nos possamos basear e identificar ideias para a aplicação. Esta pesquisa passou pela utilização de diversos *softwares* de armazenamento de e-books para uso privado, tal como o Book Collector, quer *opensource* quer por versões trial de soluções comerciais. Nesta fase de pesquisa também visualizamos as plataformas implementadas em algumas universidades públicas para catalogação e disponibilização de conteúdos quer o mesmo tipo de plataformas em bibliotecas públicas.

Com a proliferação de informação disponível a cada momento e o aumento drástico de novas publicações, tanto as bibliotecas tradicionais quer as bibliotecas privadas de consumidores mais exigentes testemunharam um desafio enorme, esse desafio resume-se por uma questão simples, como organizar e apresentar a informação de forma simples e eficiente. Dentro deste desafio surgiram múltiplas formas de abordar e resolver a mesma, algumas baseadas nos anteriores sistemas de organização bibliotecário (tal como o Sistema decimal de Dewy) quer com sistemas completamente informatizados baseados em plataformas de serviços como o SharePoint (usadas para bibliotecas internas de grandes empresas na disponibilização e organização dos seus documentos). Após visualização e teste das diversas abordagens concluímos que estávamos prontos a levantar um conjunto realista de requisitos que poderiam ser implementados no decurso deste projeto e que iriam de encontro ao que nos foi proposto.

A decisão que tomamos baseou-se numa plataforma que permita a simples indexação de resultados de livros, que permitisse a disponibilização de dados básicos tais como título, editora, escritor etc., pretendíamos assim que a nossa aplicação permita o armazenamento do livro em formato *e-book* na base de dados e a sua leitura (nos casos em que estivesse disponível).

Queríamos também permitir que através de um segundo tipo de utilizador, neste caso um administrador, fosse possível a manipulação de forma funcional, de todos os dados numa base de dados persistente.



## **DESENVOLVIMENTO**

Neste capítulo iremos abordar as etapas que constituíram o desenvolvimento, desenho e operacionalização da plataforma.

Entre as múltiplas etapas podemos destacar o levantamento de requisitos necessários para a elaboração do projeto ou o desenvolvimento propriamente dito do mesmo. São também apresentados diversos conceitos teóricos usados durante a execução do projeto

Após a apresentação deste tema específico por parte do Orientador do Projeto Global decidimos então aceitar a proposta e encetar de imediato o processo de recolha de informação para o Estado de Arte.

Após uma pesquisa por várias aplicações existentes no Mercado, quer por aplicações *opensource*, quer por aplicações de cariz comercial, passando por aplicações mais sofisticadas de organização documental, começou o processo de aquisição de requisitos e conceitos gerais do que se pretendia implementar.

Para a execução do projeto foram seguidos algumas fases importantes para a sua implementação, sendo que a baixo descrevemos os principais blocos seguidos:

- Architecture and Requisite Gathering
- Design and Development
- Positive and negative path testing.

Inicialmente foi realizada uma análise rápida por múltiplas aplicações de gestão documental e de *home library* de modo a perceber as principais funcionalidades existentes no Mercado.

De seguida e já munidos dos requisitos apresentados por parte do Orientador do Projeto, a saber, o uso de C#, utilização de uma base de dados SQL Server de modo a armazenar toda a informação e a disponibilização da dita aplicação num conceito de ambiente virtualizado pertencente a um domínio onde o utilizador estaria registado, foi iniciado o desenvolvimento propriamente dito.

A criação desta aplicação tem como objetivo final a implementação de uma aplicação que permita a um utilizador pesquisar um título de livro, autor, ou mesmo uma passagem descriptiva, de forma a obter um título de uma obra literária, no caso dessa obra o possuir, apresentar ao utilizador a dita obra em formato eletrónico.

## 1- ARCHITECTURE AND REQUISITE GATHERING

Para este projeto o conceito por trás da sua implementação passa por a intenção de uma integração entre uma biblioteca virtual e a sua congénere física. Para tal seria facultado um *Front-end* gráfico (Aplicação) para um utilizador que ao autenticar-se no Domínio dessa mesma biblioteca (física) poderia procurar de forma rápida qualquer informação disponível na sua base de dados e em alguns casos aceder de imediato ao conteúdo através do formato *e-book*.<sup>1</sup>

Para tal ser implementado de forma bem-sucedida teríamos que alavancar diversas componentes abordadas no decurso da Licenciatura, tais como Bases de Dados, Programação em C#, AD e Virtualização.

O projeto assenta em cima de uma infraestrutura virtual que iremos de seguida descrever.

- Um Domain Controller, Windows 2012R2, contendo os grupos necessários para a autenticação quer dos utilizadores comuns como do utilizador com privilégios de Administração.<sup>2</sup>
- Dois servidores Hyper-V, Windows 2012R2, contendo as máquinas virtuais necessárias para servirem de cliente para a aplicação propriamente dita e contendo também a Base de Dados, SQL Server 2014.<sup>3,4</sup>
- Clientes (múltiplos) de Windows 8, servindo de *host* para a aplicação.<sup>5</sup>

---

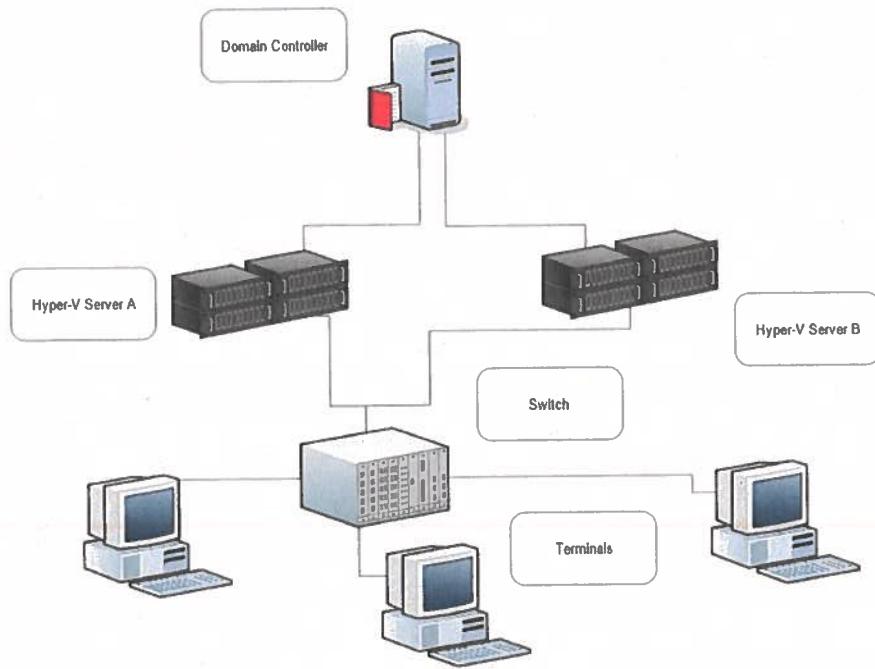
<sup>1</sup> [https://technet.microsoft.com/en-us/library/cc782657\(v=ws.10\).aspx](https://technet.microsoft.com/en-us/library/cc782657(v=ws.10).aspx)

<sup>2</sup> <https://technet.microsoft.com/en-us/library/hh801901.aspx>

<sup>3</sup> <http://blogs.technet.com/b/dataplatforminsider/archive/2013/06/05/sql-server-2014-a-closer-look.aspx>

<sup>4</sup> <https://technet.microsoft.com/en-us/windowsserver/dd448604.aspx>

<sup>5</sup> <https://technet.microsoft.com/en-us/library/hh832030.aspx>



**FIGURE 10 -DIAGRAMA DA ARQUITECTURA LOGICA**

Sendo assim um utilizador da biblioteca chegaria a um conjunto de terminais disponíveis (PC físico por exemplo), procede ao seu login com uma conta pré criada e distribuída pela Administração da mesma Biblioteca, e ao *logar-se* realiza-se a validação se o *user* contém perfil de administração ou não. Não contendo tal perfil a vista, que se apresenta uma vez lançando a aplicação, resume-se a consulta sem qualquer possibilidade de manipulação (ou completo acesso no caso do utilizador conter o perfil de Administrador).

Uma vez aberta a aplicação, o utilizador tem então a possibilidade de procurar quer por título, quer por autor ou editora, ou por uma "String"<sup>6</sup> que seja contida na descrição do mesmo Livro. No caso de obter sucesso o livro será de imediato apresentado com todos os dados previamente carregados para o mesmo, no caso de conter uma cópia digital, tal será também apresentada.

Durante a procura de qualquer tipo de informação, a aplicação abre uma ligação ao nosso servidor de Base de dados onde procura (ou no caso de edição por parte de um Administrador, executa a mesma) a referida informação na tabela de conteúdo previamente preenchida.

<sup>6</sup> <http://www.acsu.buffalo.edu/~fineberg/mfc158/week10lecture.htm>

Após obtida tal informação pelo utilizador, o utilizador pode de seguida ler o livro em formato digital caso ele exista ou escolher sair da aplicação terminando a mesma através do ícone de saída no topo esquerdo.

Para efeitos de demonstração deste projeto deve-se relembrar que todos os servidores lógicos, Domínio contendo os serviços de AD, servidores Hyper-V, servidor contendo a Base de Dados, todos se encontram virtualizados correndo sobre a Plataforma VMware Workstation<sup>7</sup>.

Numa implementação real, tais serviços seriam realizados em servidores físicos isolados e, em ultima instância, os clientes terminais poderiam ser apenas clientes Thin client<sup>8</sup> de forma a aceder ao Sistema Operativo virtualizado disponível pelos Servidores Hyper-V.

## 2- DESENVOLVIMENTO EMPÍRICO DA APLICAÇÃO

Todas as imagens contidas no presente subcapítulo foram retiradas do código elaborado no decurso do projeto.

As imagens contem parte do *code-behind* como também imagens demonstrando o *layout* gráfico da aplicação.

Como seria de esperar todas as imagens se encontram comentadas com uma breve descrição e anotação do seu significado.

---

<sup>7</sup> [https://pubs.vmware.com/Release\\_Notes/en/workstation/12pro/workstation-1201-release-notes.html](https://pubs.vmware.com/Release_Notes/en/workstation/12pro/workstation-1201-release-notes.html)

<sup>8</sup> <http://searchnetworking.techtarget.com/definition/thin-client>

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Drawing;
using System.Data;
using System.Data.SqlClient;
using System.IO;
using System.Configuration;
using System.Web;
using System.Collections;
using System.Security.Principal;
using System.DirectoryServices.ActiveDirectory;
using System.DirectoryServices.AccountManagement;
using System.ComponentModel;
using System.Drawing.Imaging;
using System.Windows.Forms;

using Telerik.WinForms;
using Telerik.WinForms.UI;
using Telerik.WinForms.Themes;

```

FIGURE 11 - LIBRARY

Estas são as bibliotecas usadas pela aplicação.

Além das bibliotecas genéricas importadas automaticamente pelo projeto de Visual Studio temos também bibliotecas adicionais importadas dos controlos de Telerik.

Estes controlos são de sobra importância pois permitem-nos costumizar de forma extremamente ágil quer os temas / *layout* do "front-end" quer aceder a customização a nível dos botões de seleção e caixas de seleção. Sendo um projeto da Telerik, vêm de raiz com bibliotecas próprias.

Os controlos da "framewok" de DevCraft<sup>9</sup> da Telerik, são de imensa importância pois permitem-nos ser muito mais versáteis que os controlos<sup>10</sup> habituais do Visual Studio.

```

string errorMsg = string.Empty;
User App_user = null;
List<string> lstCol_Admin = new List<string> { "id", "name", "description", "author", "image", "pdf", "delete" };
List<string> lstCol_User = new List<string> { "id", "name", "description", "author", "image", "pdf" };
Int32 row_edit = -1, row_pdf = -1, new_book = -1;

```

FIGURE 12 - VARIAVEIS DA APLICACAO

<sup>9</sup> <http://www.telerik.com/devcraft devcraft>

<sup>10</sup> <https://msdn.microsoft.com/en-us/library/windows/desktop/dn742484%28v=vs.85%29.aspx>

O código acima mostra a declaração<sup>11</sup> de variáveis da aplicação.

```
private void goto_Focus(List<object> lst, object position)
{
    int idx = lst.FindIndex(item => item == position);
    idx = (idx == lst.Count()) ? idx : ++idx;

    if (lst[idx].GetType() == typeof(RadTextBox)) ((RadTextBox)lst[idx]).Focus();
    else if (lst[idx].GetType() == typeof(TextBox)) ((TextBox)lst[idx]).Focus();
    else if (lst[idx].GetType() == typeof(RadBrowseEditor)) ((RadBrowseEditor)lst[idx]).Focus();
}
```

FIGURE 13 - FOCUS

Esta função permite colocar o *focus*<sup>12</sup> no objeto pretendido.

```
private void set_MaxLength(List<object> toParam, Int32 paramSize)
{
    foreach (object ctrl in toParam)
    {
        if (ctrl.GetType() == typeof(RadTextBox)) ((RadTextBox)ctrl).MaxLength = paramSize;
        else if (ctrl.GetType() == typeof(TextBox)) ((TextBox)ctrl).MaxLength = paramSize;
    }
}
```

FIGURE 14 - MAXLENGTH LIST

Esta função permite parametrizar a propriedade “MaxLength” do objeto com o tamanho máximo pretendido.

```
private static void resetObject(List<object> toReset)
{
    foreach(object ctrl in toReset)
    {
        if (ctrl.GetType() == typeof(RadTextBox)) ((RadTextBox)ctrl).Text = string.Empty;
        else if (ctrl.GetType() == typeof(TextBox)) ((TextBox)ctrl).Text = string.Empty;
        else if (ctrl.GetType() == typeof(RadCheckBox)) ((RadCheckBox)ctrl).Checked = false;
        else if (ctrl.GetType() == typeof(RadDropDownList)) ((RadDropDownList)ctrl).SelectedIndex = -1;
        else if (ctrl.GetType() == typeof(RadBrowseEditor)) ((RadBrowseEditor)ctrl).Value = string.Empty;
        else if (ctrl.GetType() == typeof(RadLabelElement)) ((RadLabelElement)ctrl).Text = string.Empty;
        else if (ctrl.GetType() == typeof(PictureBox)) ((PictureBox)ctrl).Image = Properties.Resources.sample;
    }
}
```

FIGURE 15 - RESET CONTROL LIST

<sup>11</sup> <https://msdn.microsoft.com/en-us/library/office/gg264241.aspx>

<sup>12</sup> <http://markhobley.yi.org/glossary/mousefocus.html#sloppyfocus>

Esta função é responsável por limpar todos os objetos passados por valor.

A função em causa é chamada por exemplo no *use case*<sup>13</sup> de criação de um novo e-book através da conta de administrador.

```
private static void EnabledObject(List<object> toEnable, bool onoff)
{
    foreach (object ctrl in toEnable)
    {
        if (ctrl.GetType() == typeof(RadioButton)) ((RadioButton)ctrl).Enabled = onoff;
        else if (ctrl.GetType() == typeof(RadTextBox)) ((RadTextBox)ctrl).Enabled = onoff;
    }
}
```

FIGURE 16 - ENABLE/DISABLE CONTROL LIST

Esta função é usada para fazer *enabled / disable* aos objetos passados por valor.

A função é usada no caso de desejarmos ver os dados de um e-book.

```
private void writeStatus(RadLabelElement lbl, string text, Color cor, bool reset, bool showWin, MessageBoxButtons msgBtn, RadMessageIcon msgIcon)
{
    if (reset) resetObject(new List<object> { lbl });

    lbl.ForeColor = cor;
    lbl.Text = text;
    if (showWin)
    {
        RadMessageBox SetThemeName(funcoes.c_ThemeName);
        DialogResult dr = RadMessageBox.Show(text, funcoes.c_AppName, msgIcon, MessageBoxButtons.DefaultButton.Button1);
    }
}
```

FIGURE 17 - ERROR HANDLING

Esta função é usada para controlo de "error handling"<sup>14</sup>, permite que a descrição do erro seja visível através de uma caixa de alerta e que fique visível na barra de estados da aplicação.

A função é usada na introdução de um e-book.

```
public static void writeToolTip(object ctrl, string tooltip)
{
    if (ctrl.GetType() == typeof(RadLabelElement)) ((RadLabelElement)ctrl).ToolTipText = tooltip;
    else if (ctrl.GetType() == typeof(RadLabel)) ((RadLabel)ctrl).LabelElement.ToolTipText = tooltip;
    else if (ctrl.GetType() == typeof(RadStatusBarPanelElement)) ((RadStatusBarPanelElement)ctrl).ToolTipText = tooltip;
    else if (ctrl.GetType() == typeof(RadioButton)) ((RadioButton)ctrl).ButtonElement.ToolTipText = tooltip;
}
```

FIGURE 18 - TOOLTIP

<sup>13</sup> <http://alistair.cockburn.us/Use+cases%2c+ten+years+later>

<sup>14</sup> [https://msdn.microsoft.com/en-us/library/windows/desktop/ms679320\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/ms679320(v=vs.85).aspx)

A função writeToolTip<sup>15</sup> permite parametrizar o objeto inicialmente, de forma a passar um valor de texto para a "label"<sup>16</sup> que é visualizada através da ação "mouse hover"<sup>17</sup>.

A função é usada no arranque da aplicação para inicializar os botões da barra de navegação.

```
private byte[] getResizedImage(string path, int width, int height)
{
    Bitmap imgIn = new Bitmap(path);
    double y = imgIn.Height;
    double x = imgIn.Width;

    double factor = 1;
    if (width > 0) factor = width / x;
    else if (height > 0) factor = height / y;

    MemoryStream outStream = new System.IO.MemoryStream();
    Bitmap imgOut = new Bitmap((int)(x * factor), (int)(y * factor));

    imgOut.SetResolution(72, 72);

    Graphics g = Graphics.FromImage(imgOut);
    g.Clear(Color.White);
    g.DrawImage(imgIn, new Rectangle(0, 0, (int)(factor * x), (int)(factor * y)), new Rectangle(0, 0, (int)x, (int)y), GraphicsUnit.Pixel);

    imgOut.Save(outStream, ImageFormat.Png);
    return outStream.ToArray();
}
```

FIGURE 19 - IMAGE SIZING

Na função getResizedImage encontramos o código usado para fazer a redimensionamento do tamanho de uma imagem passada por valor.

Essa função é usada no carregamento na imagem da capa do *e-book*. Evitamos assim que um utilizador insira uma imagem com um tamanho exagerado, causando problemas na apresentação do "GUI"<sup>18</sup>.

```
private void writeText(object senderR, object senderW)
{
    (senderR as RadLabel).Text = string.Format("[ {0} - {1} ] ", (senderW as RadTextBox).TextLength, (senderW as RadTextBox).MaxLength);
}
```

FIGURE 20 - CHARACTER COUNTER

<sup>15</sup> <http://www.telerik.com/products/aspnet-ajax/tooltip.aspx>

<sup>16</sup> <http://www.telerik.com/help/winforms/panels-and-labels-label-overview.html>

<sup>17</sup> <http://www.w3.org/TR/html5/dom.html#the-title-attribute mouse hover>

<sup>18</sup> <http://www.linfo.org/gui.html>

A função writeText permite informar o utilizador, através de uma RadLabel, da quantidade de caracteres que foi ou poderá ser introduzido em uma RadTextBox, passada por valor.

Essa função é usada na introdução de um novo e-book.

```
private void sizePannel(int pannel, Size nSize)
{
    sContainer.SplitPanels[pannel].SizeInfo.SizeMode = Telerik.WinForms.Docking.SplitContainerSizeMode.Absolute;
    sContainer.SplitPanels[pannel].SizeInfo.AbsoluteSize = nSize;
}
```

FIGURE 21 - PANEL

A função sizePannel por sua vez, permite definir um tamanho passado por valor a um painel.

Essa função é usada na introdução de um novo *e-book*.

```
private void collapsePannel(List<int> lstPannel, RadSplitContainer sc, bool value)
{
    if (lstPannel.Count > 0)
    {
        foreach (int i in lstPannel)
        {
            sc.SplitPanels[i].Collapsed = value;
        }
    }
}
```

FIGURE 22 - PANEL CONTROL

Na função de collapsePannel, é responsável pelo tratamento da apresentação visual dos painéis<sup>19,20</sup>.

A função é usada na situação de por exemplo, "mouse click" na imagem de capa abrindo assim o painel de informação do e-book.

<sup>19</sup> <http://publib.boulder.ibm.com/cgi-bin/bookmgr/BOOKS/F29BDG00/GLOSSARY?SHELF=ceesl002&DT=19921204095534>

<sup>20</sup> [https://msdn.microsoft.com/en-us/library/system.windows.forms.panel\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.windows.forms.panel(v=vs.110).aspx)

```

void gridRow_Data(GridViewRowInfo rinfo)
{
    if (rinfo != null)
    {
        TxtId.Text = rinfo.Cells["TxtID"].Value.ToString();
        txtAutor.Text = rinfo.Cells["TxtAutor"].Value.ToString();
        txtNome.Text = rinfo.Cells["TxtName"].Value.ToString();
        txtDescricao.Text = rinfo.Cells["TxtDesc"].Value.ToString();
        imgCapa.Image = (Image)rinfo.Cells["ImageCol"].Value;
    }
}

```

FIGURE 23 – E-BOOK PARAMETERS

A função gridRow\_Data, importa a informação alojada na grid e colocada nos respetivos campos apresentados, tais como a descrição do livro, nome, autor etc...

```

void gridMode(RadGridView grid, List<string> lstColumn)
{
    grid.MasterTemplate.Columns.Clear();
    if (lstColumn.Count > 0)
    {
        foreach (string col in lstColumn)
        {
            if (col.ToLower().Equals("id")) // id
            {
            }
            else if (col.ToLower().Equals("image")) // image
            {
            }
            else if (col.ToLower().Equals("name")) // name
            {
            }
            else if (col.ToLower().Equals("description")) // description
            {
            }
            else if (col.ToLower().Equals("author")) // author
            {
            }
            else if (col.ToLower().Equals("pdf")) // pdf
            {
            }
            else if (col.ToLower().Equals("delete")) // delete
            {
                GridViewCommandColumn colDel = new GridViewCommandColumn();
                colDel.Name = "CmdDel";
                colDel.UseDefaultText = true;
                colDel.HeaderText = string.Empty;
                colDel.MaxWidth = 60;
                colDel.MinWidth = 60;
                colDel.ImageAlignment = ContentAlignment.MiddleCenter;
                colDel.DefaultText = string.Empty;
                colDel.Image = Properties.Resources.small_edit_delete;
                grid.MasterTemplate.Columns.Add(colDel);
            }
        }
        grid.CommandCellClick += new CommandCellClickEventHandler(Gridview_CommandCellClick);
        grid.CellClick += new GridViewCellEventHandler(Gridview_CellClick);

        grid.AllowAddNewRow = false;
        grid.AllowEditRow = false;
        grid.AllowRowReorder = false;
        grid.EnableAlternatingRowColor = true;
        grid.ShowFilteringRow = false;
        grid.ShowGroupPanel = false;
        grid.EnablePaging = true;

        grid.GridViewElement.PagingPanelElement.ShowNumericalButtons = false;

        grid.MasterTemplate.AutoGenerateColumns = false;
        grid.MasterTemplate.AutoSizeColumnsMode = GridViewAutoSizeColumnsMode.Fill;
        grid.TableElement.RowHeight = 100;
    }
}

```

FIGURE 24 - GRID INIT

A função gridMode, permite a inicialização da "grid"<sup>21</sup> consoante o tipo de utilizador em uso.

A função é usada no arranque na aplicação, para configurar as várias propriedades e eventos da "grid", tais como: impedir a introdução de novas linhas, paginação, alteração de cor por linha e do evento de clicar na célula.

```
void GridView_CellClick(object sender, EventArgs e)
{
    resetObject(new List<object> { lblError });
    GridImageCellElement cellElement = sender as GridImageCellElement;

    if (cellElement != null)
    {
        row_pdf = -1;
        collapsePanel(new List<int> { 2 }, sContainer, true);
        if (row_edit.Equals(Int32.Parse(cellElement.RowInfo.Cells["TxtID"].Value.ToString())))
        {
            row_edit = -1;
            resetObject(new List<object> { txtAutor, txtNome, txtDescricao, bedtCapa, bedtPdf, imgCapa });
            collapsePanel(new List<int> { 1 }, sContainer, true);
        }
        else
        {
            row_edit = Int32.Parse(cellElement.RowInfo.Cells["TxtID"].Value.ToString());
            EnabledObject(new List<object> { btnSave, btnCancel, txtAutor, txtNome, txtDescricao, bedtCapa, bedtPdf }, false);
            if (App_user.Is_Admin)
                EnabledObject(new List<object> { btnSave, btnCancel, TxtId, txtAutor, txtNome, txtDescricao, bedtCapa, bedtPdf, imgCapa }, true);

            collapsePanel(new List<int> { 1 }, sContainer, false);
            if (cellElement.RowInfo != null)
                gridRow_Data(gridBooks.SelectedRows[0].ViewInfo.CurrentRow);
            else
                gridRow_Data(cellElement.RowInfo);
        }
    }
}
```

FIGURE 25 – GRID ROW DATA

A função GridView\_CellClick, permite ao pressionar a capa do livro, transferir a informação referente ao livro, para os respetivos campos, usando a gridRow\_Data, anteriormente referenciada.

Essa função tem dois métodos de funcionamento distinto com base no utilizador em causa, bloqueia os campos para edição para utilizadores não administradores ou mete em modo de edição para utilizadores administradores.

<sup>21</sup> <http://docs.telerik.com/devtools/wpf/controls/radgridview/overview2>

```

void GridView_CommandCellClick(object sender, EventArgs e)
{
    resetObject(new List<object> { lblError });
    GridCommandCellElement clickedCmdColumn = sender as GridCommandCellElement;

    int row_id = new_book = -1;
    if (clickedCmdColumn.RowIndex == null)
        row_id = Int32.Parse(gridbooks.SelectedRows[0].Cells["TxtID"].Value.ToString());
    else
        row_id = Int32.Parse(clickedCmdColumn.RowIndex.Cells["TxtID"].Value.ToString());

    if (clickedCmdColumn.ColumnInfo.Name.Equals("CmdPdf"))
    else if (clickedCmdColumn.ColumnInfo.Name.Equals("CmdDel"))
    {
        RadMessageBox.SetThemeName(themeName);
        DialogResult dr = RadMessageBox.Show("Deseja remover livro?", "i-book", MessageBoxButtons.OKCancel, RadMessageIcon.Question, MessageBoxDefaultButton.Button1);
        if (dr.ToString().ToUpper().Equals("OK"))
        {
            row_edit = row_pdf = -1;
            collapsePanel(new List<int> { 1, 2 }, sContainer, true);
            funcoes.del_book(row_id, ref errorMsg);
            if (!string.IsNullOrEmpty(errorMsg))
                writeStatus(lblError, errorMsg, Color.Maroon, true, true, MessageBoxButtons.OK, RadMessageIcon.Error);
            else
            {
                clickedCmdColumn.RowIndex.Delete();
                writeStatus(lblError, "Registo apagado com sucesso!", Color.Green, true, false, MessageBoxButtons.OK, RadMessageIcon.None);
            }
        }
    }
}

```

**FIGURE 26 - GRID COMMAND**

A função GridView\_CommandCellClick permite duas funcionalidades distintas que passamos a descrever:

- Comando de "PDF": O comando de "PDF" permite reaver da base de dados o "array"<sup>22</sup> de bytes correspondente a imagem virtual do livro mostrando de seguida o painel com toda a informação correspondente.
- Comando de Delete: O comando de Delete, que apenas é visualizado por administradores, permite apagar o livro, com toda a sua informação, da base de dados. A ação de delete é apenas executada através de uma caixa de diálogo<sup>23</sup> de confirmação.

```

private byte[] StreamFile(string fileName)
{
    FileStream fs = new FileStream(fileName, FileMode.Open, FileAccess.Read);
    byte[] ImageData = new byte[fs.Length];
    fs.Read(ImageData, 0, System.Convert.ToInt32(fs.Length));
    fs.Close();
    return ImageData;
}

```

**FIGURE 27 - IMAGE ARRAY**

A função de StreamFile permite fazer a transformação de um ficheiro para um "array" de "bytes" de modo a ser gravado na base de dados.

<sup>22</sup> [https://msdn.microsoft.com/en-us/library/system.array\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.array(v=vs.110).aspx)

<sup>23</sup> <http://docs.telerik.com/devtools/wpf/controls/radwindow/features/predefined-dialogs>

Foi decidido usar a conversão para um "array" de bytes de forma a evitar problemas com o armazenamento de informação na base de dados.

```

private void btnFind_Click(object sender, EventArgs e)
{
    resetObject(new List<object> { lblError });
    resetObject(new List<object> { TxtId, txtNome, txtDescrição, bdtpCapa, imgCapa });
    collapsePanel(new List<int> { 1, 2 }, scContainer, true);

    if (((DataItem)ddOption.SelectedItem).Value.Equals("-1"))
    {
        RadMessageBox SetThemeName(funcoes.c_ThemeName);
        RadMessageBox.Show("Selecione uma opção, por favor.", "Informação", MessageBoxButtons.OK, RadMessageIcon.Info, MessageBoxButtonsDefaultButton.Button1);
        ddOption.ShowDropDown();
        return;
    }

    if (((DataItem)ddFilter.SelectedItem).Value.Equals("-1"))
    {
        RadMessageBox SetThemeName(funcoes.c_ThemeName);
        RadMessageBox.Show("Selecione uma opção, por favor.", "Informação", MessageBoxButtons.OK, RadMessageIcon.Info, MessageBoxButtonsDefaultButton.Button1);
        ddFilter.ShowDropDown();
        return;
    }

    gridBooks.DataSource = null;
    row_edit = row_pdf = -1;

    if (App_user.Is_Admin)
        gridMode(gridBooks, lstCol_Admin);
    else
        gridMode(gridBooks, lstCol_User);

    List<Book> lstBook = funcoes.get_Book(((DataItem)ddOption.SelectedItem).Value, ((DataItem)ddFilter.SelectedItem.Value), txtFind.Text, ref errorMsg);
    if (!string.IsNullOrEmpty(errorMsg))
        writeStatus(lblError, errorMsg, Color.Maroon, true, true, MessageBoxButtons.OK, RadMessageIcon.Error);
    else
        gridBooks.DataSource = lstBook;
}

```

**FIGURE 28 - BOOK FIND**

A função `btnFind_Click`, permite realizar pesquisas na base de dados, segundo os critérios fornecidos pelo utilizador. Esta função é associada ao botão de pesquisa, que se encontra da barra de navegação.

```

void btn_Click(object sender, EventArgs e)
{
    if (((RadioButton)sender).Tag.ToString().Length == 1)
        txtFind.Text = string.Format("{0}", ((RadioButton)sender).Tag);
    else
        pnlAlfa.Visible = false;
}

```

**FIGURE 29 - ALPHABAR EVENT**

A função `btn_Click` permite ao clicar na lista de abecedário passar a sua letra associada a cada botão para a caixa de pesquisa. Esta função é associada ao botão do abecedário, que se encontra na barra de navegação<sup>24</sup>.

<sup>24</sup> <https://msdn.microsoft.com/en-us/library/windows/apps/xaml/dn440584.aspx>

```

public RadButton makeButton(string name, string text, string tag, string imgPath, Point pointBtn, int width, int height, string tooltip, DockStyle dock)
{
    object resource;
    RadButton btn = new RadButton();
    btn.Name = name;

    if (!string.IsNullOrEmpty(imgPath))
    {
        ResourceSet resourceSet = i_book2.Properties.Resources.ResourceManager.GetResourceSet(System.Globalization.CultureInfo.CurrentCulture, true, true);
        foreach (System.Collections.DictionaryEntry entry in resourceSet)
        {
            if (((string)entry.Key).ToLower().Equals(imgPath.ToLower()))
            {
                resource = entry.Value;
                btn.Image = (Image)entry.Value;
                break;
            }
        }
    }

    btn.EnableTheming = true;
    btn.ThemeClassName = funcoes.c_ThemeName;
    btn.Location = pointBtn;
    btn.Width = width;
    btn.Height = height;
    btn.Dock = dock;
    btn.ButtonElement.BorderElement.BoxStyle = BorderBoxStyle.SingleBorder;
    btn.ButtonElement.BorderElement.Width = 1;
    btn.ButtonElement.BorderElement.UpdateLayout();
    btn.ImageAlign = ContentAlignment.MiddleCenter;
    btn.TextImageRelation = TextImageRelation.ImageBeforeText;
    btn.ThemeClassName = "TelerikMetro";
    btn.UseCompatibleTextRendering = true;
    btn.Click += new EventHandler(btn_Click);
}

btn.Tag = tag;
if (!string.IsNullOrEmpty(tooltip)) btn.ButtonElement.ToolTipText = tooltip;
btn.Click += new EventHandler(btn_Click);
return btn;
}

```

**FIGURE 30 - CREATE BUTTON**

A função makeButton permite criar botões do tipo RadButton<sup>25</sup> dinamicamente. Esta função é usada para construir a barra do abecedário.

```

private void set_AlphaBar()
{
    string _path = string.Empty;
    List<RadButton> LstButtons = new List<RadButton>();
    int nbtn = 0;
    int btnspace = 26;
    int btnstart = 2;
    for (int i = 65; i <= 90; i++)
        LstButtons.Add(makeButton(string.Format("btn_{0}", (char)i), string.Empty, string.Format("{0}", (char)i),
            string.Format("{0}letter_{1}", _path, (char)i), new Point(nbtn++ * btnspace + btnstart, 1), 26, 26, string.Format("Letra {0}", (char)i),
            DockStyle.None));

    LstButtons.Add(makeButton(string.Format("btn_{0}", (char)88), string.Empty, "*",
        string.Format("{0}letter_{1}", _path, (char)88), new Point(nbtn++ * btnspace + btnstart, 2), 26, 26, string.Format("Todo"), DockStyle.None));
    LstButtons.Add(makeButton(string.Format("btn_closepal"), string.Empty, string.Empty,
        string.Format("{0}cross_button", _path), new Point(nbtn++ * btnspace + btnstart, 2), 26, 26, string.Format("Fechar painel"), DockStyle.Right));

    foreach (RadButton btn in LstButtons)
        pnllfa.Controls.Add(btn);
    ThemeResolutionService.ApplyThemeToControlTree(this, funcoes.c_ThemeName);
}

```

**FIGURE 31 - ALPHABAR**

---

<sup>25</sup> <http://www.telerik.com/help/winforms/buttons-button-overview.html>

A função set\_Alphabar construi a barra de navegação que se encontra no topo da aplicação imediatamente por debaixo da barra de procura. Esta barra é composta por as letras do alfabeto permitindo uma procura rápida pela primeira letra do título do livro pretendido.

```
private void btnAlfa_Click(object sender, EventArgs e)
{
    resetObject(new List<object> { lblError });
    if (!pnlAlfa.Visible) pnlAlfa.Visible = true;
}
```

FIGURE 32 - ALPHABAR TRIGGER

A função btnAlfa\_Click permite que a barra de navegação do alfabeto fique visível.

```
private void btnExit_Click(object sender, EventArgs e)
{
    RadMessageBox.SetThemeName(funcoes.c_ThemeName);
    DialogResult dr = RadMessageBox.Show("Deseja sair da Aplicação?", funcoes.c_AppName, MessageBoxButtons.OKCancel, RadMessageBoxIcon.Question, RadMessageBoxDefaultButton.Button1);
    if (dr.ToString().ToUpper() == "OK") Application.Exit();
}
```

FIGURE 33 - APP CLOSE

A função btnExit\_Click, permite ao clicar no botão de *On/Off* localizado no topo esquerdo da barra de navegação execute o fecho da aplicação após a confirmação através de uma caixa de diálogo.

```
private void btnAddBook_Click(object sender, EventArgs e)
{
    gridBooks.DataSource = null;
    resetObject(new List<object> { lblError });
    row_edit = row_pdf = -1;
    if (new_book.Equals(-1))
    {
        new_book = 1;
        resetObject(new List<object> { TxtId, txtAutor, txtNome, txtDescricao, bdtCapa, bdtPdf, imgCapa });
        collapsePannel(new List<int> { 2 }, sContainer, true);
        collapsePannel(new List<int> { 1 }, sContainer, false);
        sizePannel(1, new Size(0, 300));
        txtAutor.Focus();
    }
    else
    {
        new_book = -1;
        collapsePannel(new List<int> { 1 }, sContainer, true);
    }
}
```

FIGURE 34 - ADD BOOK

A função `btnAddBook_Click` permite ao utilizador administrador a inserção de um novo livro, com todos os seus dados correspondentes. No caso do *e-book* de um livro anterior já se encontrar expandido, colapsa primeiro esse painel e apresenta de seguida o painel de inserção de dados.

```
private void btnCancel_Click(object sender, EventArgs e)
{
    new_book = -1;
    resetObject(new List<object> { TxtId, txtAutor, txtNome, txtDescricao, bedtCapa, bedtPdf, imgCapa });
    collapsePannel(new List<int> { 1 }, sContainer, true);
}
```

FIGURE 35 - CANCEL BOOK

A função `btnCancel_Click` por sua vez, cancela a inserção de um livro invocada pela função `btnAddBook_Click`, fechando o painel e inicializa a variável<sup>26</sup> `new_book`.

```
private void mainForm_Load(object sender, EventArgs e)
{
    resetObject(new List<object> { lblUser, lblError });

    OpenFileDialog dialog = (OpenFileDialog)bedtCapa.Dialog;
    dialog.Filter = "PNG Files (*.png)|*.png|JPG Files(*.jpg)|*.jpg|All Files (*.*)|*.*";

    OpenFileDialog dialogPdf = (OpenFileDialog)bedtPdf.Dialog;
    dialogPdf.Filter = "Adobe PDF Files (*.pdf)|*.pdf";

    sContainer.EnableCollapsing = true;
    sContainer.UseSplitterButtons = false;
    sContainer.BeginInit();
    collapsePannel(new List<int> { 1, 2 }, sContainer, true);
    sContainer.EndInit();

    set_MaxLength(new List<object> { txtAutor, txtNome }, 120);
    set_MaxLength(new List<object> { txtDescricao }, 450);

    writeText(lblC_Autor, txtAutor);
    writeText(lblC_Nome, txtNome);
    writeText(lblC_Descricao, txtDescricao);

    writeToolTip(btnExit, "Sair");
    writeToolTip(btnAddBook, "Adicionar livro");
    writeToolTip(btnFind, "Pesquisa");
    writeToolTip(btnAlfa, "Alfabeto");

    pnlAlfa.Visible = false;
    set_AlphaBar();

    App_user = new user(Environment.UserName);
    if (App_user.Is_Admin)
        writeStatus(lblUser, String.Format("Admin: {0}", Environment.UserName), Color.Black, true, false, MessageBoxButtons.OK, RadMessageIcon.None);
    else
        writeStatus(lblUser, String.Format("User: {0}", Environment.UserName), Color.Black, true, false, MessageBoxButtons.OK, RadMessageIcon.None);

    DataItem di = new DataItem();
    ddOption.Items.Clear();
    ddOption.DisplayMember = "Option";
    ddOption.DataSource = di.get_FindOption();

    ddFilter.Items.Clear();
    ddFilter.DisplayMember = "Option";
    ddFilter.DataSource = di.get_Filter();

    ThemeResolutionService.ApplyThemeToControlTree(this, funcoes.c_ThemeName);
}
```

FIGURE 36 - FORM LOAD

---

<sup>26</sup> [https://www.le.ac.uk/users/rjm1/cotter/page\\_20.htm](https://www.le.ac.uk/users/rjm1/cotter/page_20.htm)

A função mainForm\_Load permite a inicialização dos objetos do "Form"<sup>27</sup> no arranque do programa.

Essa inicialização faz o *reset* das "Labels"<sup>28</sup> e carrega as "Combo Boxes"<sup>29</sup> na barra de navegação, carrega os filtros para as caixas de diálogo do *e-book* e da Capa, fixa o tamanho dos campos de Autor, Nome, Descrição etcetera ..., esconde os dois painéis do *e-book* e dos campos, cria a barra do alfabeto, inicializa a *grid* conforme os atributos de utilizador - verificando primeiro se o utilizador existe no seu grupo de AD<sup>30</sup> e por final define o tema de arranque do "Form"<sup>31</sup>.

```
private void txtAutor_Enter(object sender, EventArgs e)
{
    writeText(lblC_Autor, txtAutor);
}

private void txtAutor_TextChanged(object sender, EventArgs e)
{
    writeText(lblC_Autor, txtAutor);
}
```

FIGURE 37 - AUTOR EVENT

A imagem anterior apresenta os eventos associados as caixas de texto, referentes aos dados do *e-book*.

A função writeText já foi anteriormente descrita.

```
private void txtAutor_KeyPress(object sender, KeyPressEventArgs e)
{
    if (e.KeyChar == 13) goto_Forward(new List<object> { txtAutor, txtNome, txtDescricao, bedtCapa, bedtPdf }, sender);
}
```

FIGURE 38 - CONTROL FOCUS EVENT

<sup>27</sup> <http://www.telerik.com/help/winforms/forms-and-dialogs-overview.html>

<sup>28</sup> <http://www.telerik.com/help/winforms/panels-and-labels-label-overview.html>

<sup>29</sup> <http://www.telerik.com/help/winforms/dropdown-and-listcontrol-dropdownlist-overview.html>

<sup>30</sup> [https://msdn.microsoft.com/en-us/library/aa746492\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/aa746492(v=vs.85).aspx)

<sup>31</sup> [https://msdn.microsoft.com/en-us/library/system.windows.forms.form\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.windows.forms.form(v=vs.110).aspx)

O evento<sup>32</sup> txtAutor\_KeyPress também associado às caixas de texto, referentes aos dados do *e-book*, e vai fazer uso da função goto\_Forward, que faz essencialmente é colocar o cursor no controlo seguinte, de uma lista de controlos passada por valor.

```
private void bedtCapa_ValueChanged(object sender, EventArgs e)
{
    try
    {
        if (File.Exists(bedtCapa.Value))
        {
            imgCapa.Image = Image.FromStream(new MemoryStream(StreamFile(bedtCapa.Value)));
        }
    }
    catch
    {
        imgCapa.Image = Properties.Resources.sample;
        bedtCapa.Value = string.Empty;
    }
}
```

**FIGURE 39 - COVER LOAD**

A função bedtCapa\_ValueChanged permite validar se a *path*<sup>33</sup> física que o utilizador administrador introduzido contém uma imagem válida, no caso de tal não se verificar, automaticamente a função realiza o carregamento de uma imagem padrão previamente definida nos recursos do programa.

```
private bool checkFile_Exists(string filename, string msgError, bool showWin)
{
    bool result = true;
    try
    {
        if (!File.Exists(filename))
        {
            result = false;
            if (showWin)
            {
                RadMessageBox.SetThemeName(funcoes.c_ThemeName);
                RadMessageBox.Show(msgError, funcoes.c_AppName, MessageBoxButtons.OKCancel, RadMessageIcon.Error, MessageBoxButtonsDefaultButton.Button1);
            }
        }
    }
    catch { result = false; }
    return result;
}
```

**FIGURE 40 - FILE CHECK**

A função checkFile\_Exists, vai verificar se um ficheiro existe em um *path* passado por valor. Esta função é usada na gravação dos dados do *e-book*.

---

<sup>32</sup> [https://msdn.microsoft.com/en-us/library/17sde2xt\(v=VS.100\).aspx](https://msdn.microsoft.com/en-us/library/17sde2xt(v=VS.100).aspx)

<sup>33</sup> [https://msdn.microsoft.com/en-us/library/system.io.path\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.io.path(v=vs.110).aspx)

```

private bool checkField_Empty(object ctrl, string msgError, bool showWin, bool gotoField)
{
    bool result = false;
    if (ctrl.GetType() == typeof(RadTextBox))
    {
        if (string.IsNullOrEmpty(((RadTextBox)ctrl).Text))
        {
            result = true;
            if (showWin)
            {
                RadMessageBox.SetThemeName(funcoes.c_ThemeName);
                RadMessageBox.Show(msgError, funcoes.c_AppName, MessageBoxButtons.OKCancel, RadMessageIcon.Error, MessageBoxButtonsDefaultButton.Button1);
            }

            if (gotoField) ((RadTextBox)ctrl).Focus();
        }
    }
    else if (ctrl.GetType() == typeof(RadBrowseEditor))
    {
        if (string.IsNullOrEmpty(((RadBrowseEditor)ctrl).Value))
        {
            result = true;
            if (showWin)
            {
                RadMessageBox.SetThemeName(funcoes.c_ThemeName);
                RadMessageBox.Show(msgError, funcoes.c_AppName, MessageBoxButtons.OKCancel, RadMessageIcon.Error, MessageBoxButtonsDefaultButton.Button1);
            }

            if (gotoField) ((RadBrowseEditor)ctrl).SelectNextControl(((Control)ctrl), true, true, false, false);
        }
    }
    return result;
}

```

FIGURE 41 - CHECK CONTENT

A função `checkField_Empty` valida se os campos preenchidos pelo utilizador administrador se encontram vazios, em caso de não se encontrarem preenchidos desencadeia uma mensagem de erro informando de tal e posicionando o cursor no campo que se encontra vazio ou com erro.

```

private void btnSave_Click(object sender, EventArgs e)
{
    resetObject(new List<object> { lblError });
    if (checkField_Empty(txtAutor, "Preencha o campo [ Autor ], por favor!", true, true)) return;
    if (checkField_Empty(txtName, "Preencha o campo [ Nome ], por favor!", true, true)) return;
    if (checkField_Empty(txtDescricao, "Preencha o campo [ Descrição ], por favor!", true, true)) return;

    if (string.IsNullOrEmpty(txtId.Text))
    {
        if (checkField_Empty(bedtCapa, "Preencha o campo [ Capa ], por favor!", true, true)) return;
        if (checkField_Empty(bedtPdf, "Preencha o campo [ Ficheiro ], por favor!", true, true)) return;
    }

    if (!string.IsNullOrEmpty(bedtCapa.Value))
        if (!checkFile_Exists(bedtCapa.Value, "Imagen inexistente!", true)) return;

    if (!string.IsNullOrEmpty(bedtPdf.Value))
        if (!checkFile_Exists(bedtPdf.Value, "Pdf inexistente!", true)) return;

    funcoes.insUpd_Book(TxtId.Text, txtName.Text, txtDescricao.Text, txtAutor.Text,
        string.IsNullOrEmpty(bedtCapa.Value) ? null : StreamFile(bedtCapa.Value),
        string.IsNullOrEmpty(bedtPdf.Value) ? null : StreamFile(bedtPdf.Value),
        ref errorMsg);

    if (!string.IsNullOrEmpty(errorMsg))
        writeStatus(lblError, errorMsg, Color.Maroon, true, true, MessageBoxButtons.OK, RadMessageIcon.Error);
    else
    {
        gridBooks.DataSource = null;
        resetObject(new List<object> { TxtId, txtAutor, txtName, txtDescricao, bedtCapa, bedtPdf, imgCapa });
        writeStatus(lblError, "Registo adicionado com sucesso.", Color.Green, true, false, MessageBoxButtons.OK, RadMessageIcon.None);
    }
}

```

FIGURE 42 - BOOK SAVE

A função `btnSave_Click` permite a gravação do dados de um *e-book*, após esses terem passado por um processo de validação.

O resultado dessa ação pode mostrar dois estados, sucesso ou erro.

Em qualquer um dos estados é apresentado uma janela com a mensagem respetiva.

```
public class book
{
    public Int32 id { get; set; }
    public string name { get; set; }
    public string description { get; set; }
    public string author { get; set; }
    public byte[] book_pdf { get; set; }
    public Image book_img { get; set; }
}
```

FIGURE 43 - BOOK CLASS

A classe<sup>34</sup> *book* define todos os parâmetros de um *e-book*.

Entre os quais, os atributos de id, nome, descrição, autor, o *e-book* do livro e a imagem da sua capa.

---

<sup>34</sup> <https://msdn.microsoft.com/en-us/library/x9afc042.aspx>

```

class DataItem
{
    private string option;
    private string value;

    public string Option { get { return this.option; } }
    public string Value { get { return this.value; } }

    public DataItem() { }

    public DataItem(string option, string value)
    {
        this.option = option;
        this.value = value;
    }

    public DataItem[] get_FindOption()
    {
        DataItem[] result = new DataItem[5] { new DataItem("...Selecione uma opção...", "-1"),
                                              new DataItem("por nome / autor / descrição", "0"),
                                              new DataItem("por nome", "1"),
                                              new DataItem("por autor", "2"),
                                              new DataItem("por descrição", "3"),
                                         };
        return result;
    }

    public DataItem[] get_FinderFilter()
    {
        DataItem[] result = new DataItem[7] { new DataItem("...Selecione uma opção...", "-1"),
                                              new DataItem("igual a", "0"),
                                              new DataItem("diferente de", "1"),
                                              new DataItem("contendo", "2"),
                                              new DataItem("não contendo", "3"),
                                              new DataItem("começa com", "4"),
                                              new DataItem("termina com", "5"),
                                         };
        return result;
    }
}

```

FIGURE 44 - DATAITEM CLASS

A classe DataItem contém os métodos `get_FindOption` e `get_FinderFilter`, que vão alimentar os controlos usados nos filtros.

Como podemos ver na imagem as opções de filtro são pré-definidas de modo a simplificar a pesquisa do utilizador, e de modo a evitar que os resultados sejam demasiado abrangentes, retornando demasiada informação.

O objetivo é de proporcionar uma pesquisa rápida e eficaz.

```

public static void del_book(Int32 id, ref string error)
{
    error = string.Empty;
    SqlConnection tempconn = new SqlConnection(cConnectionString);
    try
    {
        if (tempconn.State == ConnectionState.Closed) tempconn.Open();

        List<string> visibleFields = new List<string>();
        TADOStoredProcedure adoSpP = new TADOStoredProcedure("del_BOOK_BY_ID", tempconn);
        adoSpP.addParam(new TADOParameter("@id", id));
        adoSpP.open(visibleFields, CommandType.StoredProcedure, 200);
    }
    catch (Exception ex)
    {
        error = "Erro [ del_book ] - " + ex.Message;
    }
    finally
    {
        if (tempconn.State == ConnectionState.Open) tempconn.Close();
    }
}

```

FIGURE 45 - BOOK DELETE

O método `del_book` é usado para a remoção de um livro e da sua informação completa da base de dados.

Tal ação é realizada passando o `id` como parâmetro para a "stored procedure"<sup>35</sup> `del_BOOK_BY_ID`, caso se verifique um erro uma exceção é capturada e de seguida apresentado um erro, através do *pop up* de uma mensagem.

O erro apresentado é construído através da mensagem que é capturada pela exceção fazendo referência ao método que foi invocado, de modo a permitir ao utilizador compreender de forma rápida o que poderá ter ocorrido de errado.

---

<sup>35</sup> <https://msdn.microsoft.com/en-us/library/ms190782.aspx>

```

public static List<book> get_Book_id(Int32 id, ref string error)
{
    error = string.Empty;
    SqlConnection tempconn = new SqlConnection(cConnectionString);
    try
    {
        if (tempconn.State == ConnectionState.Closed) tempconn.Open();

        List<string> visibleFields = new List<string>();
        TADOStoredProcedure adoSpP = new TADOStoredProcedure("get_BOOK_BY_ID", tempconn);
        adoSpP.addParam(new TADOParameter("@id", id));
        adoSpP.open(visibleFields, CommandType.StoredProcedure, 200);
        I Enumerable<book> dt = (from m in adoSpP.getDataTable().AsEnumerable()
                                    select new book
                                    {
                                        id = m.Field<Int32>("ID"),
                                        name = m.Field<string>("NAME"),
                                        description = m.Field<string>("DESCRIPTION"),
                                        author = m.Field<string>("AUTHOR"),
                                        book_img = Image.FromStream(new MemoryStream(m.Field<byte[]>("BOOK_IMAGE"))),
                                        book_pdf = m.Field<byte[]>("BOOK_PDF"),
                                    }).ToList();
        return dt.ToList();
    }
    catch (Exception ex)
    {
        error = "Erro [ get_Book_id ] - " + ex.Message;
        return null;
    }
    finally
    {
        if (tempconn.State == ConnectionState.Open) tempconn.Close();
    }
}

```

**FIGURE 46 - BOOK GET BY ID**

O método `get_Book_id` é usado para retornar a informação de um livro e da sua informação completa da base de dados.

Tal ação é realizada passando o `id` como parâmetro para a "stored procedure" `get_BOOK_BY_ID`, caso se verifique um erro uma exceção é capturada e de seguida apresentado um erro, através do *pop up* de uma mensagem.

O erro apresentado é construído através da mensagem que é capturada pela exceção fazendo referência ao método que foi invocado, de modo a permitir ao utilizador compreender de forma rápida o que poderá ter ocorrido de errado.

```

public static List<book> get_Book(string field, string filter, string text, ref string error)
{
    error = string.Empty;
    SqlConnection tempconn = new SqlConnection(c_ConnectionString);
    try
    {
        if (tempconn.State == ConnectionState.Closed) tempconn.Open();

        List<string> visibleFields = new List<string>();
        TADOStoredProcedure adoSpP = new TADOStoredProcedure("get_BOOK_FIND", tempconn);
        adoSpP.addParam(new TADOParameter("@findField", field));
        adoSpP.addParam(new TADOParameter("@findFilter", filter));
        adoSpP.addParam(new TADOParameter("@findText", text));

        adoSpP.open(visibleFields, CommandType.StoredProcedure, 200);
        IEnumerable<book> dt = (from m in adoSpP.getDataTable().AsEnumerable()
                                select new book
                                {
                                    id = m.Field<Int32>("ID"),
                                    name = m.Field<string>("NAME"),
                                    description = m.Field<string>("DESCRIPTION"),
                                    author = m.Field<string>("AUTHOR"),
                                    book_img = Image.FromStream(new MemoryStream(m.Field<byte[]>("BOOK_IMAGE"))),
                                }).ToList();
        return dt.ToList();
    }
    catch (Exception ex)
    {
        error = "Erro [ get_Book ] - " + ex.Message;
        return null;
    }
    finally
    {
        if (tempconn.State == ConnectionState.Open) tempconn.Close();
    }
}

```

**FIGURE 47 - BOOK GET**

O método `get_Book` é usado para a pesquisa de um livro ou livros com os parâmetros seguintes:

- Nome
- Autor
- Descrição
- Combinação dos anteriores

De seguida podem ser aplicados filtros tais como:

- Igual a
- Diferente de
- Contendo
- Não contendo
- Começa com
- Termina com

Passando por final a informação que se pretende pesquisar.

Tal ação é realizada passando o `id` como parâmetro para a "stored procedure" `get_BOOK_Find`, caso se verifique um erro uma exceção é capturada e de seguida é apresentado um erro, através do `pop up` de uma mensagem, erro esse que é construído através da mensagem que é capturada pela exceção fazendo referência ao método que foi invocado.

```

public static void insUpd_Book(string id, string name, string description, string author, byte[] book_image, byte[] book_pdf, ref string error)
{
    error = string.Empty;
    SqlConnection tempconn = new SqlConnection(cConnectionString);
    try
    {
        if (tempconn.State == ConnectionState.Closed) tempconn.Open();

        List<string> visibleFields = new List<string>();
        TADOStoredProc adoSpP = new TADOStoredProc("INS_UPD_BOOK", tempconn);
        if (!string.IsNullOrEmpty(id)) adoSpP.addParam(new TADOParameter("@id", id));
        adoSpP.addParam(new TADOParameter("@name", name));
        adoSpP.addParam(new TADOParameter("@description", description));
        adoSpP.addParam(new TADOParameter("@author", author));
        if (book_image != null) adoSpP.addParam(new TADOParameter("@book_image", book_image));
        if (book_pdf != null) adoSpP.addParam(new TADOParameter("@book_pdf", book_pdf));
        adoSpP.open(visibleFields, CommandType.StoredProcedure, 200);
    }
    catch (Exception ex)
    {
        error = "Error [ del_book ] - " + ex.Message;
    }
    finally
    {
        if (tempconn.State == ConnectionState.Open) tempconn.Close();
    }
}

```

**FIGURE 48 - BOOK UPDATE OR INSERT**

O método `insUpd_Book` vai inserir ou fazer *update* aos dados do objeto Livro dependendo do parâmetro `id`.

Se o parâmetro for *null* desencadeia uma ação de *insert*, no caso de o parâmetro incluir um `id` pré existente e feito então um *update*.

Tal ação é realizada passando o `id` como parâmetro para a stored procedure `INS_UPD_BOOK`, caso se verifique um erro uma exceção é capturada e de seguida apresentado um erro, através do pop up de uma mensagem.

O erro apresentado é construído através da mensagem que é capturada pela exceção fazendo referência ao método que foi invocado.

Esta opção permite ao utilizador compreender de forma rápida o que poderá ter ocorrido de errado.

```

public class TADOParameter
{
    private string paramName;
    private Object paramValue;

    public TADOParameter()
    {
        paramName = "";
        paramValue = null;
    }

    public TADOParameter(string paramName, Object paramValue)
    {
        this.paramName = paramName;
        this.paramValue = paramValue;
    }

    public void setParamName(string paramName)
    {
        this.paramName = paramName;
    }

    public string getParamName()
    {
        return this.paramName;
    }

    public void setParamValue(Object paramValue)
    {
        this.paramValue = paramValue;
    }

    public Object getParamValue()
    {
        return this.paramValue;
    }
}

```

**FIGURE 49 - ADOPARAMETER CLASS**

A Classe TADOParameter vai ser usada para guardar os parâmetros e os seus valores usados nas chamadas das “store procedures”.

Estes parâmetros podem ser usados para colocar valores na base de dados, ou ler dados da mesma.

```

public class TADOStoredProcedure
{
    private string procedureName;
    private List<TADOParameter> paramList = new List<TADOParameter>();
    private SqlConnection dbconnection;
    private DataTable dataTable;
    private DataSet dataSet;

    public TADOStoredProcedure()
    {
        procedureName = "";
        paramList = null;
        dbconnection = null;
        dataTable = null;
    }

    public TADOStoredProcedure(string procedureName,
        SqlConnection dbconnection)
    {
        this.procedureName = procedureName;
        this.dbconnection = dbconnection;
    }

    public void setParamList(List<TADOParameter> paramList)
    {
        this.paramList = paramList;
    }
}

```

FIGURE 50 - ADOSTOREDPROCEDURE CLASS

A classe TADOStoredProcedure inicializa a ligação a base de dados e passando como parâmetro o nome da StoredProcedure, além disso passa uma lista, item a item, da classe TADOParameter como parâmetro também.

```

public void addParam(string paramName, Object paramValue)
{
    if (paramValue != null)
        this.paramList.Add(new TADOParameter(paramName, paramValue));
}

public void addParam(TADOParameter param)
{
    this.paramList.Add(param);
}

public void open(List<string> visibleFields, CommandType SQLtype)
{
    if (this.dbconnection != null)
    {
        SqlCommand sqlComm = new SqlCommand(this.procedureName, this.dbconnection);
        sqlComm.CommandType = SQLtype;
        foreach (TADOParameter param in this.paramList)
            sqlComm.Parameters.AddWithValue(param.getParamName(), param.getParamValue());
        this.dataTable = createDataTable(sqlComm.ExecuteReader(), visibleFields);
    }
}

```

FIGURE 51 - PARAMETER LOAD

O método addParam pode ser invocado de duas formas diferentes. O método é usado para carregar a lista de parâmetros usada nas chamadas das "store procedures".

O método *open* executa um tipo de comando de SQL<sup>36</sup>.

```
public DataTable getDataTable()
{
    return this.dataTable;
}

private bool existsInStringList(string value, List<string> stringList)
{
    bool result = false;
    for (int i = 0; i < stringList.Count; i++)
        if (value.ToUpper().Equals(stringList[i].ToUpper()))
    {
        result = true;
        break;
    }
    return result;
}
```

FIGURE 52 - GET DATA

O método getDataTable<sup>37</sup> retorna todos os dados referentes ao(s) *e-book(s)* adquiridos da base de dados.

Apenas são devolvidos valores caso exista na base de dados *e-books* com os atributos de pesquisa pretendidos pelo utilizador

O método existsInStringList valida se o valor passado na variável "value", se encontra na lista de *strings* stringList.

---

<sup>36</sup> [https://msdn.microsoft.com/en-us/library/system.data.commandtype\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.data.commandtype(v=vs.110).aspx)

<sup>37</sup> <https://msdn.microsoft.com/en-us/library/system.data.datatable.aspx>

```

private DataTable createDataTable(SqlDataReader dataReader, List<string> visibleFields)
{
    if (dataReader != null)
    {
        DataTable dt = new DataTable("dtResult_" + DateTime.Now.ToString("ss"));
        DataRow dr;

        for (int i = 0; i < dataReader.FieldCount; i++)
        {
            if (existsInStringList(dataReader.GetName(i), visibleFields) || visibleFields.Count == 0)
                dt.Columns.Add(dataReader.GetName(i), dataReader.GetFieldType(i));
        }

        while (dataReader.Read())
        {
            dr = dt.NewRow();
            for (int i = 0; i < dt.Columns.Count; i++)
            {
                dr[i] = dataReader[dt.Columns[i].ColumnName];
            }
            dt.Rows.Add(dr);
        }
        dataReader.Close();
        return dt;
    }
    return null;
}

```

FIGURE 53 - CREATE DATA

O método CreateDataTable devolve os dados retornados pela storeprocedure, no formato de uma variável do tipo DataTable<sup>38</sup>.

```

public class user
{
    private const string dc_user = "XXXXXX";
    private const string dc_pass = "XXXXXX";
    private const string dc_name = "XXXXXX";
    private const string dc_ou = "DC=vXXXX,DC=XXXX";

    private string user_name;
    private bool is_admin;

    public user(string user_name)
    {
        this.user_name = user_name;
        this.is_admin = checkIsAdmin();
    }

    public string User_Name { get { return this.user_name; } }

    public bool Is_Admin { get { return this.is_admin; } }
}

```

FIGURE 54 - USER CLASS

---

<sup>38</sup> [https://msdn.microsoft.com/en-us/library/system.data.datatable\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.data.datatable(v=vs.110).aspx)

A classe *user* é usada para validar se o utilizador *logado* detém privilégios de administrador ou não.

Para tal vai a "Active Directory" validar no grupo dos administradores se o utilizador em causa se encontra no mesmo.

No caso de não ser validado com este grupo, por não lhe pertencer, fica no campo *is\_admin* a falso, *logando* o utilizador como utilizador sem privilégios de administração.

A validação não é feita contra um segundo grupo de utilizadores autorizados pois o projeto assume que o utilizador ao *logar-se* na sua máquina virtual está a aceder a um conjunto de programas autorizados, entre os quais se encontra esta aplicação.

No caso de este pressuposto não se validar a mesma chamada iria validar de seguida no *loop* contra um Segundo grupo de utilizadores.

No caso do mesmo não se encontrar nele (estamos a utilizar o "Windows Authentication"<sup>39</sup> para tal) a aplicação fecharia indicando o erro de autenticação.

```
private PrincipalContext GetPrincipalContext()
{
    PrincipalContext PC = new PrincipalContext(ContextType.Domain, dc_name, dc_ou, ContextOptions.Signing, dc_user, dc_pass);
    return PC;
}

private UserPrincipal GetUser(string s_uname)
{
    PrincipalContext PC = GetPrincipalContext();
    UserPrincipal Up = UserPrincipal.FindByIdentity(PC, s_uname);
    return Up;
}

private GroupPrincipal GetGroup(string s_gname)
{
    PrincipalContext PC = GetPrincipalContext();
    GroupPrincipal GP = GroupPrincipal.FindByIdentity(PC, s_gname);
    return GP;
}
```

FIGURE 55 - AD AUTHENTICATION

---

<sup>39</sup> <https://msdn.microsoft.com/en-us/library/ff647396.aspx>

Este conjunto de funções é usado especificamente para validar o utilizador (e seus atributos) comunicando diretamente com a AD, usando para isso a função PrincipalContext<sup>40,41,42</sup>, UserPrincipal<sup>43</sup> e a GroupPrincipal<sup>44</sup>.

```
private bool checkIsAdmin()
{
    bool _is_admin = false;
    try
    {
        UserPrincipal UP = GetUser(this.user_name);
        GroupPrincipal GP = GetGroup("Administrators");

        if (UP != null && GP != null)
        {
            _is_admin = GP.Members.Contains(UP);

            foreach (Principal p in GP.GetMembers(true))
            {
                if (UP.SamAccountName.Equals(p.SamAccountName))
                {
                    _is_admin = true;
                    break;
                }
            }
        }
        catch { _is_admin = false; }
    return _is_admin;
}
```

FIGURE 56 - USER VALIDATION

O método chekIsAdmin faz a validação se o utilizador contém os privilégios de administrador ou não.

No caso de não pertencer ao grupo de administradores o utilizador é colocado por omissão no grupo de privilégios de utilizador regular e acede então a aplicação em modo de consulta.

<sup>40</sup> [https://msdn.microsoft.com/en-us/library/bb356158\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/bb356158(v=vs.110).aspx)

<sup>41</sup> [https://msdn.microsoft.com/en-us/library/system.directoryservices.accountmanagement.contexttexttype\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.directoryservices.accountmanagement.contexttexttype(v=vs.110).aspx)

<sup>42</sup> [https://msdn.microsoft.com/en-us/library/system.directoryservices.accountmanagement.contexttopitions\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.directoryservices.accountmanagement.contexttopitions(v=vs.110).aspx)

<sup>43</sup> [https://msdn.microsoft.com/en-us/library/system.directoryservices.accountmanagement.userprincipal\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.directoryservices.accountmanagement.userprincipal(v=vs.110).aspx)

<sup>44</sup> [https://msdn.microsoft.com/en-us/library/system.directoryservices.accountmanagement.groupprincipal\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.directoryservices.accountmanagement.groupprincipal(v=vs.110).aspx)

Name	Data Type	Allow Nulls
ID	int	<input type="checkbox"/>
NAME	varchar(120)	<input checked="" type="checkbox"/>
DESCRIPTION	varchar(650)	<input checked="" type="checkbox"/>
AUTHOR	varchar(120)	<input checked="" type="checkbox"/>
BOOK_IMAGE	varbinary(MAX)	<input checked="" type="checkbox"/>
BOOK_PDF	varbinary(MAX)	<input checked="" type="checkbox"/>
INSERT_DATE	datetime	<input checked="" type="checkbox"/>
UPDATE_DATE	datetime	<input checked="" type="checkbox"/>

FIGURE 57 - TABLE PROPERTIES

A aplicação contém uma tabela única que contém os seguintes atributos:

- ID
- Nome
- Descrição
- Autor
- Imagem
- PDF
- Data de inserção
- Data de alteração

Sendo que tanto as imagens como os *e-books* são guardados num formato "varbinary"<sup>45</sup>.

```
CREATE PROCEDURE [dbo].[del_BOOK_BY_ID]
@id INT
AS
BEGIN
    DELETE FROM [BOOKS] WHERE ID = @id
END
GO
```

FIGURE 58 - PROCEDURE DEL BOOK BY ID

---

<sup>45</sup> [https://msdn.microsoft.com/en-US/library/swbw1ewb\(v=vs.80\).aspx](https://msdn.microsoft.com/en-US/library/swbw1ewb(v=vs.80).aspx)

A "procedure" del\_BOOK\_BY\_ID é usada para apagar diretamente todos os registros de um livro passando apenas o seu id interno único.

Este procedimento é apenas usado no caso em que o utilizador que se encontra a utilizar a aplicação tem privilégios de administrador.

```
CREATE PROCEDURE [dbo].[get_BOOK_FIND] --1, 1, 'yet'
    @findField INT, -- 0 TUDO | 1 NAME | 2 AUTHOR | 3 DESCRIPTION
    @findFilter INT, -- 0 EQUAL | 1 NOT EQUAL | 2 CONTAINS | 3 NOT CONTAINS | 4 STARTS WITH | 5 ENDS WITH
    @findText VARCHAR(100) = NULL
AS
BEGIN
    DECLARE @SQL_TEXT VARCHAR(250) = NULL,
            @SQL_WHERE VARCHAR(250) = NULL,
            @SQL_FILTER VARCHAR(150) = NULL

    SET @SQL_TEXT = 'SELECT ID, NAME, [DESCRIPTION], AUTHOR, BOOK_IMAGE, BOOK_PDF, INSERT_DATE FROM BOOKS '
    IF (@findFilter IN (0,2,4,5)) SET @SQL_FILTER = 'LIKE '
    IF (@findFilter IN (1,3)) SET @SQL_FILTER = 'NOT LIKE '

    IF (@findFilter IN (0)) SET @SQL_FILTER = @SQL_FILTER + '%%' + REPLACE(@findText,'%', '') + '%%'
    ELSE IF (@findFilter IN (2,3)) SET @SQL_FILTER = @SQL_FILTER + '(%%' + REPLACE(@findText,'%', '') + '%)'
    ELSE IF (@findFilter IN (1,4)) SET @SQL_FILTER = @SQL_FILTER + '(%%' + REPLACE(@findText,'%', '') + '%%%)'
    ELSE IF (@findFilter IN (5)) SET @SQL_FILTER = @SQL_FILTER + '(%%' + REPLACE(@findText,'%', '') + '%%)'

    ELSE SET @SQL_FILTER = ''

    IF (@findField = 0)
    BEGIN
        SET @SQL_WHERE = 'WHERE NAME ' + @SQL_FILTER +
                        ' OR AUTHOR ' + @SQL_FILTER +
                        ' OR [DESCRIPTION] ' + @SQL_FILTER
    END
    ELSE IF (@findField = 1) SET @SQL_WHERE = 'WHERE NAME ' + @SQL_FILTER
    ELSE IF (@findField = 2) SET @SQL_WHERE = 'WHERE AUTHOR ' + @SQL_FILTER
    ELSE IF (@findField = 3) SET @SQL_WHERE = 'WHERE [DESCRIPTION] ' + @SQL_FILTER
    ELSE SET @SQL_WHERE = ''

    EXECUTE (@SQL_TEXT + @SQL_WHERE)
END
```

FIGURE 59 - PROCEDURE GET BOOK BY ID

A "procedure" get\_BOOK\_FIND é usada para pesquisar na base de dados através de um conjunto de filtros de pesquisa passados pela aplicação e de seguida validados contra os dados existentes na tabela de dados.

```

CREATE PROCEDURE [dbo].[INS_UPD_BOOK]
    @id INT = NULL,
    @name VARCHAR(120),
    @description VARCHAR(450),
    @author VARCHAR(80),
    @book_image VARBINARY(MAX) = NULL,
    @book_pdf VARBINARY(MAX) = NULL
AS
BEGIN
    IF (@id IS NULL)
        BEGIN
            INSERT INTO BOOKS (NAME, [DESCRIPTION], AUTHOR, BOOK_IMAGE, BOOK_PDF)
            SELECT @name, @description, @author, @book_image, @book_pdf
        END
    ELSE
        BEGIN
            UPDATE BOOKS
            SET NAME = @name,
                [DESCRIPTION] = @description,
                AUTHOR = @author,
                BOOK_IMAGE = CASE WHEN @book_image IS NULL THEN BOOK_IMAGE ELSE @book_image END,
                BOOK_PDF = CASE WHEN @book_pdf IS NULL THEN BOOK_PDF ELSE @book_pdf END,
                UPDATE_DATE = GETDATE()
            WHERE ID = @id
        END
END

```

FIGURE 60 - PROCEDURE UPDATE BOOK

A "procedure" INS\_UPD\_BOOK é utilizada quer para inserir um livro e toda a sua informação por completo, quer para atualizar ou, um vários, campos da sua informação.

```

CREATE PROCEDURE [dbo].[get_BOOK_BY_ID]
    @id INT
AS
BEGIN
    SELECT ID, NAME, [DESCRIPTION], AUTHOR, BOOK_IMAGE, BOOK_PDF
    FROM [BOOKS]
    WHERE ID = @id
END

```

FIGURE 61 - PROCEDURE GET BOOK BY ID

A "procedure" get\_BOOK\_BY\_ID permite a procura de um livro pelo seu id interno devolvendo de seguida todos os dados contidos para esse mesmo livro, quer o nome, autor, imagem de capa ou o *e-book*.

```

CREATE TABLE [dbo].[BOOKS] (
    [ID] [int] IDENTITY(1,1) NOT NULL,
    [NAME] [varchar](120) NULL,
    [DESCRIPTION] [varchar](650) NULL,
    [AUTHOR] [varchar](120) NULL,
    [BOOK_IMAGE] [varbinary](max) NULL,
    [BOOK_PDF] [varbinary](max) NULL,
    [INSERT_DATE] [datetime] NULL,
    [UPDATE_DATE] [datetime] NULL
) ON [PRIMARY]

```

FIGURE 62 - BOOKS TABLE

"Procedure" de criação da tabela de dados.

A tabela de dados contem todos os campos utilizados pela aplicação para armazenar a informação para o objeto Livro.

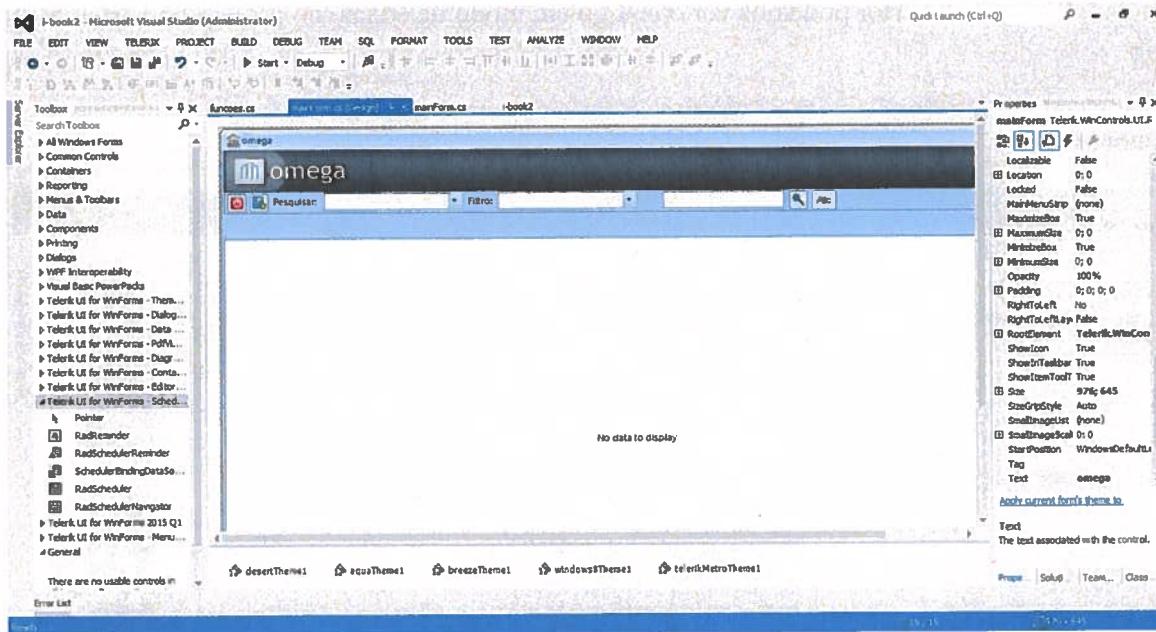


FIGURE 63 - FRONTEND

Na figura anterior pode-se encontrar o GUI do "fron-tend" da aplicação em modo de visualização de projeto no Visual Studio.

```

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

namespace L-book2
{
    public partial class mainForm : MetroFramework.Forms.MetroForm
    {
        public mainForm()
        {
            InitializeComponent();
        }

        string connectionString;
        string tableName = "L-book";
        SqlConnection conn = null;
        SqlCommand cmdInsert = new SqlCommand("insert into " + tableName + " values(@name, @desc, @author, @path, @edit)", conn);
        SqlCommand cmdDelete = new SqlCommand("delete from " + tableName + " where id = " + newBook + ")", conn);

        RegionInfo region;
        private void addForward(object objectPosition)
        {
            private void reverseLength(object objectLength)
            {
                private static void reverseObject(object object)
                {
                    foreach (object obj in object)
                }
            }
        }
    }
}

```

**FIGURE 64 - CODE BEHIND**

Na figura anterior podemos ver o código em modo de edição no projeto de Visual Studio.

Id	Name	Descrição	Autor	Imagen
11	C# Programming: From Problem Analysis to Program Design	Gain a solid understanding of today's most important programming principles with Doyle's C# PROGRAMMING FROM PROBLEM ANALYSIS TO PROGRAM DESIGN 4/E. This insightful introductory book highlights the latest Visual Studio 2012 and C# 4.0 software with a unique, principles-based approach to give you a deep understanding of programming concepts. Respected author Barbara Doyle	Barbara Doyle	
15	Writing High-Performance .Net Code	Do you want your .NET code to have the absolute best performance it can? This book demystifies the CLR, teaching you how and why to write code with optimum performance. Learn critical lessons from a person who helped design and build one of the largest high-performance .NET systems in the world. This book does not just teach you how the CLR works—it teaches you exactly what you need. A unique book-and-DVD package from preeminent programming instructor Rod Stephens, Visual C# has become a leading programming language, resulting in greater career opportunities for Visual C# programmers. This Wrox guide literally shows novice programmers how to program in C# with Microsoft Visual Studio, using both written and visual instruction. Easy-to-follow lessons re...	Ben Watson	
17	Windows Communication Foundation 4	Do you want your .NET code to have the absolute best performance it can? This book demystifies the CLR, teaching you how and why to write code with optimum performance. Learn critical lessons from a person who helped design and build one of the largest high-performance .NET systems in the world. This book does not just teach you how the CLR works—it teaches you exactly what you need. A unique book-and-DVD package from preeminent programming instructor Rod Stephens, Visual C# has become a leading programming language, resulting in greater career opportunities for Visual C# programmers. This Wrox guide literally shows novice programmers how to program in C# with Microsoft Visual Studio, using both written and visual instruction. Easy-to-follow lessons re...	Rod Stephens	
18	Stephens' C# Programming with Visual Studio 2010	Do you want your .NET code to have the absolute best performance it can? This book demystifies the CLR, teaching you how and why to write code with optimum performance. Learn critical lessons from a person who helped design and build one of the largest high-performance .NET systems in the world. This book does not just teach you how the CLR works—it teaches you exactly what you need. A unique book-and-DVD package from preeminent programming instructor Rod Stephens, Visual C# has become a leading programming language, resulting in greater career opportunities for Visual C# programmers. This Wrox guide literally shows novice programmers how to program in C# with Microsoft Visual Studio, using both written and visual instruction. Easy-to-follow lessons re...	Rod Stephens	
43	TESTE	TESTE	TESTE	
2	C# 5.0 in a Nutshell: The Definitive Reference	When you have a question about how to use C# 5.0 or the .NET CLR, this highly acclaimed bestseller has precisely the answers you need. Uniquely organized around concepts and use cases, this fifth edition	Joseph Albahari	

**FIGURE 65 - FILTERED VIEW**

Na figura acima podemos encontrar uma pesquisa por dois filtros combinados mais um valor de entrada definido pelo utilizador.

Neste caso, a procura foi efetuada por nome diferente do introduzido pelo utilizador, devolvendo de seguida todos os exemplos de *e-books* existentes na base de dados.

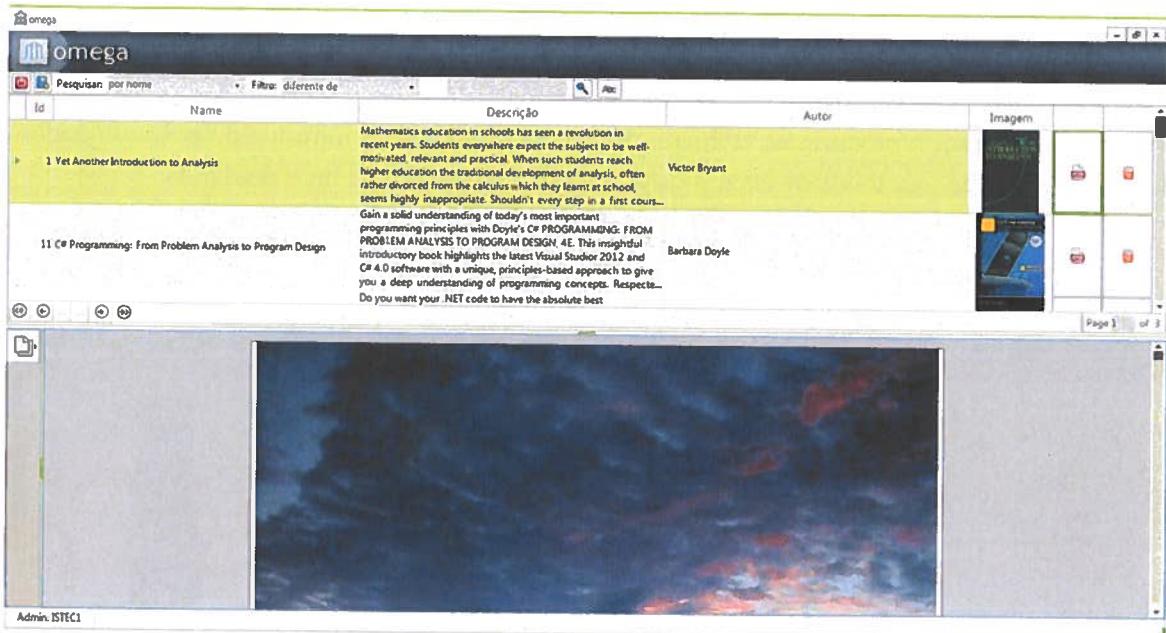
The screenshot shows a software interface with a toolbar at the top. Below the toolbar is a search bar labeled 'Pesquisar por nome' and a dropdown menu 'Filtros: diferente de'. The main area contains a table with columns: Id, Name, Descrição, Autor, and Imagem. The first row, highlighted in yellow, corresponds to the book '1 Yet Another Introduction to Analysis' by Victor Bryant. The second row corresponds to '11 C# Programming: From Problem Analysis to Program Design' by Barbara Doyle. The third row corresponds to '15 Writing High-Performance .Net Code' by Ben Watson. Below the table, there is a large panel displaying detailed information about the selected book ('1 Yet Another Introduction to Analysis'). This panel includes fields for Autor (Victor Bryant), Nome (Yet Another Introduction to Analysis), Capa (none), and Ficheiros (none). The Descrição field contains a detailed description of the book's content and purpose. At the bottom of the panel, it says 'Admin: ISTEC1'.

FIGURE 66 - BOOK VIEW

Na figura acima vemos o resultado da expansão do painel após clicar na imagem de um livro, neste caso o livro com ID1.

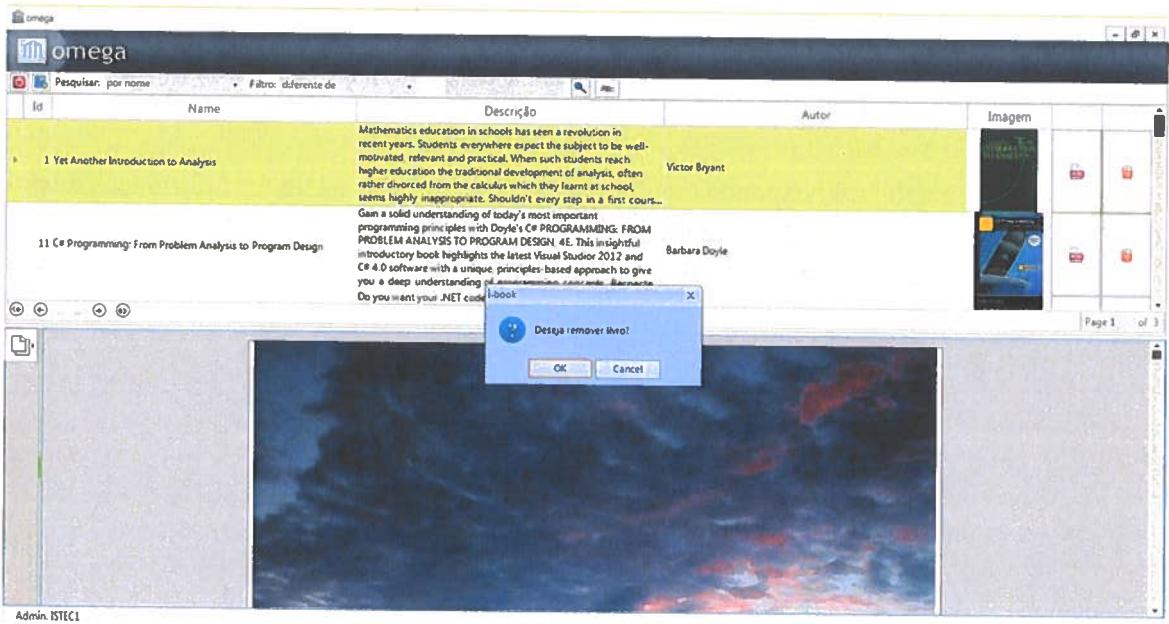
Ao clicar a aplicação expande o painel de forma a mostrar qualquer informação disponível para o objeto Livro existente na base de dados.

Podemos verificar que são apresentados todos os atributos do Livro, mesmo quando estes não possuem informação, no exemplo dado o ficheiro.



**FIGURE 67 - E-BOOK VIEW**

A figura acima mostra o resultado obtido após se clicar no símbolo de PDF de um dos livros, ao clicar-se no ícone a aplicação de seguida faz o "retrieve" do *e-book* da base de dados, expande o "painele" inferior e apresenta o *e-book* para leitura no caso de estar disponível na base de dados.



**FIGURE 68 - BOOK REMOVAL**

Na imagem anterior podemos ver a caixa de diálogo de confirmação no caso de se carregar no item de delete.

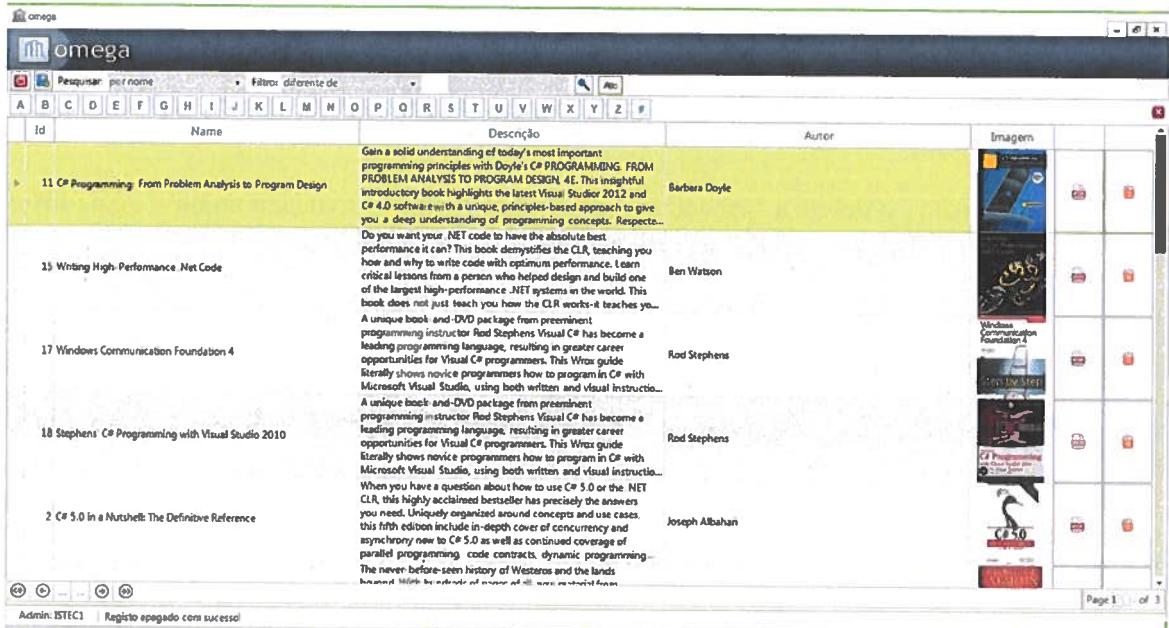
Ao ser clicado o delete a "procedure" de "delete by id" é invocada na base de dados e todos os dados associados ao id passado são removidos da mesma.

Id	Name	Descrição	Autor	Imagem	
11	C# Programming: From Problem Analysis to Program Design	Gain a solid understanding of today's most important programming principles with Doyle's C# PROGRAMMING: FROM PROBLEM ANALYSIS TO PROGRAM DESIGN, 4E. This insightful introductory book highlights the latest Visual Studio 2012 and C# 4.0 software with a unique, principles-based approach to give you a deep understanding of programming concepts. Respective...	Barbara Doyle		
15	Writing High-Performance .Net Code	Do you want your .NET code to have the absolute best performance it can? This book demystifies the CLR, teaching you how and why to write code with optimum performance. Learn critical lessons from a person who helped design and build one of the largest high-performance .NET systems in the world. This book does not just teach you how the CLR works—it teaches yo...	Ben Watson		
17	Windows Communication Foundation 4	A unique book-and-DVD package from preeminent programming instructor Rod Stephens' Visual C# has become a leading programming language, resulting in greater career opportunities for Visual C# programmers. This Wrox guide literally shows novice programmers how to program in C# with Microsoft Visual Studio, using both written and visual instruc...	Rod Stephens		
18	Stephens' C# Programming with Visual Studio 2010	A unique book-and-DVD package from preeminent programming instructor Rod Stephens' Visual C# has become a leading programming language, resulting in greater career opportunities for Visual C# programmers. This Wrox guide literally shows novice programmers how to program in C# with Microsoft Visual Studio, using both written and visual instruc...	Rod Stephens		
2	C# 5.0 in a Nutshell: The Definitive Reference	When you have a question about how to use C# 5.0 or the .NET CLR, this highly acclaimed bestseller has precisely the answers you need. Uniquely organized around concepts and use cases, this fifth edition include in-depth coverage of concurrency and asynchrony new to C# 5.0 as well as continued coverage of parallel programming, code contracts, dynamic programming—The never-before-seen history of Westeros and the lands beyond. With hundreds of pages of all-new material from George R.R. Martin. If the past is prologue, then George R.R.	Joseph Albahari		
19	The World of Ice and Fire: The Untold History of the World of A	HarperVoyager			

FIGURE 69 - REMOVE SUCCESS MESSAGE

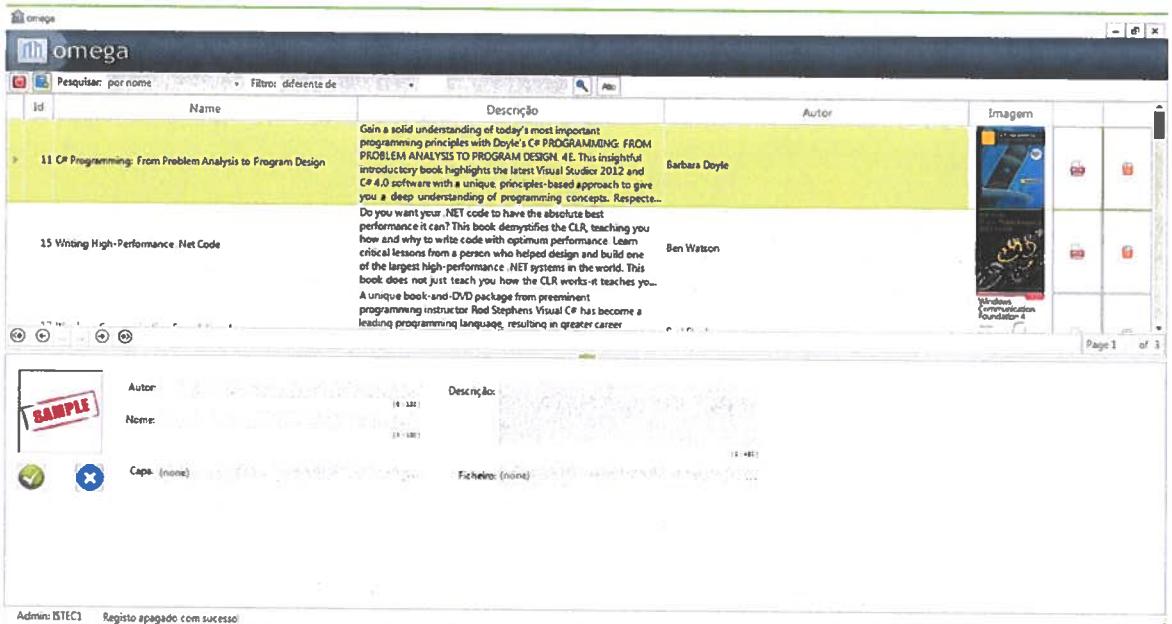
Na imagem anterior, podemos ver a mensagem de sucesso que é apresentada ao utilizador, após a remoção de um livro por completo (a verde na barra inferior da página).

Caso a remoção não ocorresse com sucesso seria apresentada uma mensagem no mesmo local com a indicação de erro a vermelho.



**FIGURE 70 - ALPHA BAR**

Barra de navegação rápida facilitando a procura pela primeira letra do Título de cada livro.



**FIGURE 71 - ADD BOOK VIEW**

Vista de edição para novo objeto Livro em modo de administração.

**FIGURE 72 - FILTER VIEW**

Id	Pesquisar: por nome / autor / descrição	Filtro: diferente de	Descrição	Autor	Imagem
11	11 C# Programs por descrição	Program Design	Gain a solid understanding of today's most important programming principles with Doyle's C# PROGRAMMING: FROM PROBLEM ANALYSIS TO PROGRAM DESIGN, 4E. This insightful introductory book highlights the latest Visual Studio 2012 and C# 4.0 software with a unique, principles-based approach to give you a deep understanding of programming concepts. Respect...	Barbara Doyle	
15	15 Writing High-Performance .Net Code		Do you want your .NET code to have the absolute best performance it can? This book demystifies the CLR, teaching you how and why to write code with optimum performance. Learn critical lessons from a person who helped design and build one of the largest high-performance .NET systems in the world. This book does not just teach you how the CLR works—it teaches you...	Ben Watson	
17	17 Windows Communication Foundation 4		A unique book-and-DVD package from preeminent programming instructor Rod Stephens Visual C# has become a leading programming language, resulting in greater career opportunities for Visual C# programmers. This Wrox guide literally shows novice programmers how to program in C# with Microsoft Visual Studio, using both written and visual instruction...	Rod Stephens	
18	18 Stephens' C# Programming with Visual Studio 2010		A unique book-and-DVD package from preeminent programming instructor Rod Stephens Visual C# has become a leading programming language, resulting in greater career opportunities for Visual C# programmers. This Wrox guide literally shows novice programmers how to program in C# with Microsoft Visual Studio, using both written and visual instruction...	Rod Stephens	
2	2 C# 5.0 in a Nutshell: The Definitive Reference		When you have a question about how to use C# 5.0 or the .NET CLR, this highly acclaimed bestseller has precisely the answers you need. Uniquely organized around concepts and use cases, this fifth edition include in-depth cover of concurrency and asynchrony new to C# 5.0 as well as continued coverage of parallel programming, code contracts, dynamic programming...	Joseph Albahari	
19	19 The World of Ice and Fire: The Untold History of the World of A		The never-before-seen history of Westeros and the lands beyond. With hundreds of pages of all-new material from George R.R. Martin. If the past is prologue, then George R.R.	HarperVoyager	

**Filtro de seleção por tipo. O filtro contém todos os atributos existentes na informação do objeto Livro.**

**FIGURE 73 - SEARCH TYPE VIEW**

Id	Name	Filtro: diferente de	Descrição	Autor	Imagem
11	11 C# Programming: From Problem Analysis to Program Design	- Selecione uma opção - por nome / autor / descrição por nome por autor	Gain a solid understanding of today's most important programming principles with Doyle's C# PROGRAMMING: FROM PROBLEM ANALYSIS TO PROGRAM DESIGN, 4E. This insightful introductory book highlights the latest Visual Studio 2012 and C# 4.0 software with a unique, principles-based approach to give you a deep understanding of programming concepts. Respect...	Barbara Doyle	
15	15 Writing High-Performance .Net Code		Do you want your .NET code to have the absolute best performance it can? This book demystifies the CLR, teaching you how and why to write code with optimum performance. Learn critical lessons from a person who helped design and build one of the largest high-performance .NET systems in the world. This book does not just teach you how the CLR works—it teaches you...	Ben Watson	
17	17 Windows Communication Foundation 4		A unique book-and-DVD package from preeminent programming instructor Rod Stephens Visual C# has become a leading programming language, resulting in greater career opportunities for Visual C# programmers. This Wrox guide literally shows novice programmers how to program in C# with Microsoft Visual Studio, using both written and visual instruction...	Rod Stephens	
18	18 Stephens' C# Programming with Visual Studio 2010		A unique book-and-DVD package from preeminent programming instructor Rod Stephens Visual C# has become a leading programming language, resulting in greater career opportunities for Visual C# programmers. This Wrox guide literally shows novice programmers how to program in C# with Microsoft Visual Studio, using both written and visual instruction...	Rod Stephens	
2	2 C# 5.0 in a Nutshell: The Definitive Reference		When you have a question about how to use C# 5.0 or the .NET CLR, this highly acclaimed bestseller has precisely the answers you need. Uniquely organized around concepts and use cases, this fifth edition include in-depth cover of concurrency and asynchrony new to C# 5.0 as well as continued coverage of parallel programming, code contracts, dynamic programming...	Joseph Albahari	
19	19 The World of Ice and Fire: The Untold History of the World of A		The never-before-seen history of Westeros and the lands beyond. With hundreds of pages of all-new material from George R.R. Martin. If the past is prologue, then George R.R.	HarperVoyager	

**FIGURE 73 - SEARCH TYPE VIEW**

O filtro permite introduzir parâmetros de exclusão e inclusão tais como “diferente de” ou “igual a”.

Tais filtros permitem uma melhor pesquisa na base de dados, limitando os valores apresentados duma forma mais eficaz.

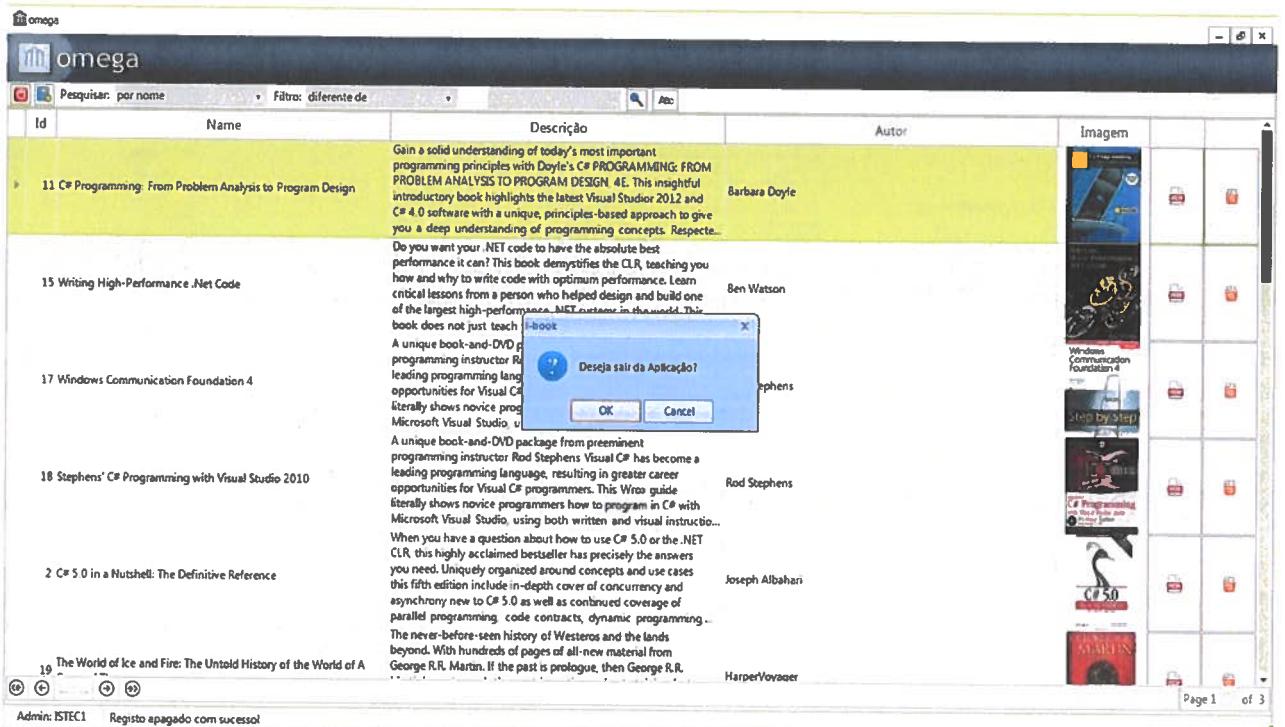


FIGURE 74 - APP EXIT

Sempre que um utilizador pretenda sair da aplicação tem á sua disponibilização um botão de sair da mesma.

Ao pressionar esse botão será apresentado um diálogo de confirmação para o mesmo validar se pretende mesmo sair da aplicação ou não.

# TESTES DE USABILIDADE

Na fase final de desenvolvimento do projeto, e de forma a podermos assegurar que todas as funcionalidades se encontravam implementadas de forma pretendida, decidimos realizar uma bateria de testes.

Esta bateria deveria conter desde os aspetos mais básicos de navegação até alguns "test cases"<sup>46</sup> onde se iria esperar um erro, o chamado "negative path"<sup>47</sup>.

Na tabela abaixo descrevemos os casos testados e resultados obtidos após a sua validação, quer para a chamada "happy path"<sup>48</sup> (regressão positiva) quer para a "negative path". Os casos encontram-se detalhados e enriquecidos com os passos usados para os realizar e resultados finais esperados.

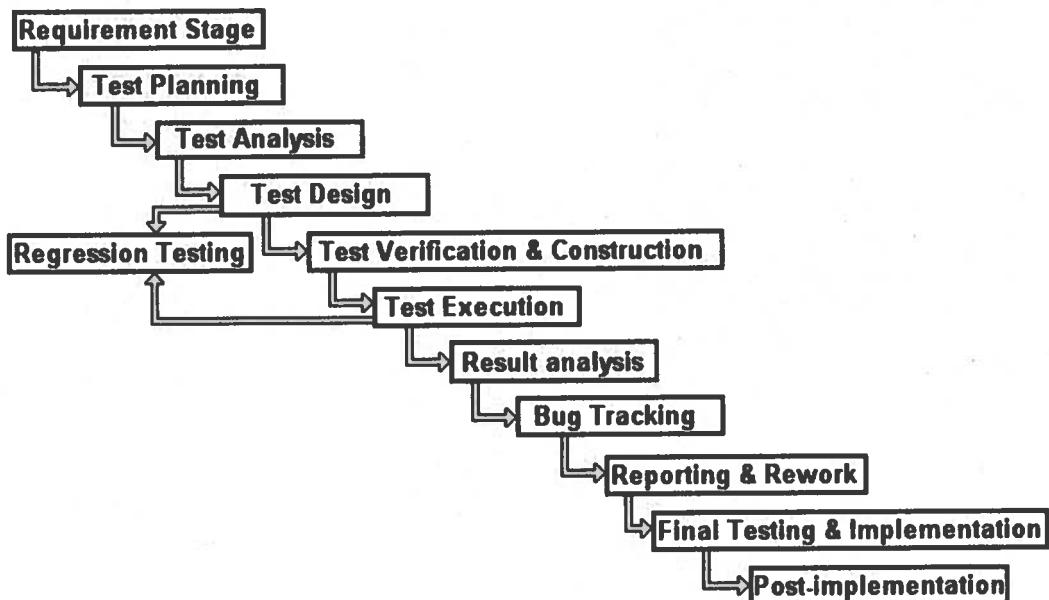


FIGURE 75 - ETAPAS DE TESTES ( KEKARE HARSHADA, HTTP://WWW.BUZZLE.COM/ 2013 )

<sup>46</sup> <http://www.stickyminds.com/article/software-test-case-engineering-treating-test-cases-product-or-approach-finding-defects-have>

<sup>47</sup> <http://www.softwaretestinghelp.com/what-is-negative-testing/>

<sup>48</sup> <http://xunitpatterns.com/happy%20path.html>

Use case ID	Test Case #	Expected Result	Steps to Execute	Status	Execution date	
		Situation	Scenarios			
1	NPV01-01	Navigation Panel Novo ebook	Apresentar uma janela com a pergunta "Descrição da Aplicação?" Sair da Aplicação Listar os campos referentes nos dados do ebook Na capa do livro apresentar a imagem definida por definição	Cenario que visa testar a funcionalidade da aplicação   Exit da aplicação, garantindo o seu encerramento Cenario que visa testar a funcionalidade das adições um novo ebook.	Pausa	05-06-2015
2	NPV01-02	Navigation Panel bem-vindo!	Pesquisar ebook Validar os campos referentes a pesquisa Pesquisar na Base de Dados pelas condições definidas	Cenario que visa testar a funcionalidade da pesquisa ebooks.	Pausa	05-06-2015
3	NPV01-03	Navigation Panel bem-vindo!	Painel do Alfabeto Fazer a pesquisar a painel do alfabeto	Cenario que visa testar a funcionalidade da pesquisa do painel do alfabeto	Pausa	05-06-2015
4	NPV01-04	Navigation Panel bem-vindo!	Pesquisar ebook Colocar a letra definida em cada posição na caixa de texto da pesquisa	Cenario que visa testar a funcionalidade da pesquisa da digitação da letra	Pausa	05-06-2015
5	ABP01-01	Alphabet Bar bem-vindo!	Painel do Alfabeto Faz o desaparecer o painel do alfabeto	Cenario que visa testar a funcionalidade da desaparição do painel do alfabeto	Pausa	05-06-2015
6	ABP01-02	Alphabet Bar bem-vindo!	Visualizar ebook Mostrar o ebook no painel do painel	Cenario que visa testar a funcionalidade da visualização do ebook	Pausa	05-06-2015
7	GRV01-01	Gravar bem-vindo!	Remover ebook Apresentar uma janela com a pergunta "Querida Remover livro?" Remover o ebook da Base de Dados	Cenario que visa testar a funcionalidade da remoção de um ebook, garantindo o seu desaparecimento da base de dados.	Pausa	20-06-2015
8	GRV01-02	Gravar bem-vindo!	Dados do ebook Mostrar os dados do ebook [ Autor, Nome, Descrição e Capa ]	Cenario que visa testar a funcionalidade de visualizar os dados do ebook	Pausa	20-06-2015
9	GRV01-03	Gravar bem-vindo!	Gravar ebook Gravar os dados do ebook	Cenario que visa testar a funcionalidade de gravar os dados do ebook	Pausa	20-06-2015
10	SPR01-01	Book Field bem-vindo!	Ler par ebook Ler par o nome do autor do ebook	Cenario que visa testar a funcionalidade da leitura de um ebook, garantido o seu desaparecimento da base de dados.	Pausa	04-07-2015
11	SPR01-02	Book Field bem-vindo!	Dados do ebook Utilizar o nome do autor do ebook	Cenario que visa testar a funcionalidade da leitura de um ebook, garantido o seu desaparecimento da base de dados.	Pausa	04-07-2015
12	SPR01-03	Book Field bem-vindo!	Dados do ebook Caracteres do campo	Cenario que visa testar a funcionalidade da visualização dos dados do ebook.	Pausa	04-07-2015
13	SPR01-04	Book Field bem-vindo!	Visualizar a descrição do ebook Desponta o evento que faz mostrar o número de caracteres do campo	Cenario que visa testar a funcionalidade da visualização dos dados do ebook.	Pausa	18-07-2015
14	SPR01-05	Book Field bem-vindo!	Visualizar o nome do ebook Desponta o evento que faz mostrar o numero de caracteres do campo	Cenario que visa testar a funcionalidade da visualização dos dados do ebook.	Pausa	18-07-2015
15	BRP01-06	Book Field bem-vindo!	Selecionar imagem abre uma janela para seleção de um pdf	Cenario que visa testar a funcionalidade de seleção de uma imagem, através de uma caixa de dia opção	Pausa	18-07-2015
15	BRP01-07	Book Field bem-vindo!	Selecionar pdf Abre uma janela para seleção de um pdf	Cenario que visa testar a funcionalidade de seleção de um pdf, através de uma caixa de dia opção	Pausa	18-07-2015

FIGURE 76 – TABELA DE TESTES

## CONCLUSÃO

O objetivo deste projeto foi o desenvolvimento de uma aplicação em C# para a organização de conteúdos literários e sua informação pertencente. Para a realização do mesmo usamos conceitos e tecnologias tais como SQL Server, C#, Virtualização, AD.

O projeto levou sensivelmente 8 meses a ser desenvolvido, contando o período de recolha de material para o Estado de Arte ate aos testes finais. Sendo que a recolha de informação para o estado de arte colheu a larga maioria desse tempo (aproximadamente 4 meses).

O projeto arrancou pela recolha e consulta de diversas fontes tais como obras publicadas, outras teses disponíveis, diversos Websites de forma a abranger a maior quantidade possível de informação disponível, sendo que imediatamente de seguida se fez a filtragem do conteúdo estritamente necessário para o projeto.

O passo seguinte foi a elaboração do Estado de Arte, contando com o apoio e opinião de alguns profissionais da área de IT e de um Professor de Ciências informáticas.

Após a conclusão do anterior referido, o desenvolvimento empírico da aplicação teve início, o que nos revelou algumas omissões no desenho de alguns requisitos, escolha de "Frontend", tais como pequenos obstáculos não estimados anteriormente. Foi feita também uma gestão inteligente do tempo de forma a retirar imediatamente *snippets* de código de forma a poderem ser usados como evidencias anotadas na parte teórica do documento do Projeto Global.

Concluídas todas as etapas anteriores, podemos afirmar que os objetivos propostos foram alcançados.

Apesar de vários atrasos, quer na conclusão programática da aplicação, derivado de algumas modificações gráficas ou erros na especificação dos requisitos, quer na elaboração do documento escrito, os objetivos propostos foram atingidos.

Como um nível de auto critica que um projeto desta natureza a si obriga no final do mesmo, devo apontar que os maiores problemas passaram por uma larga inexperiência e sobreestimação do tempo necessário para realizar um conjunto de tarefas proposto (como muito bem referido, inúmeras vezes, pelo Orientador do Projeto Global) uma tentativa de estender os requisitos alem de uma aplicação simples e funcional, no fundo, tentando açambarcar "tudo e mais alguma coisa".

Apesar de todos os obstáculos encontrados e atrasos sentidos, podemos afirmar que foi uma experiencia deveras positiva e com um resultado satisfatório quer a nível do resultado obtido quer a nível do firmar de conhecimentos adquiridos no decurso da Licenciatura.



## REFERÊNCIAS BIBLIOGRÁFICAS

JANSSEN, Cory (2013)..NET Framework (.NET). (Consultado em Janeiro de 2015). Disponível no Web Site da Techopedia: <https://www.techopedia.com/definition/1243/.Net>

DALY, [Jimmy](#) (2013). What Is Virtualization? (Consultado em Janeiro de 2015). Disponível no Web Site da edtechmagazine : <http://www.edtechmagazine.com>

Woodford Chris (2014). Cloud computing. (Consultado em Janeiro de 2015). Disponível no Web Site da Explainthatstuff : <http://www.explainthatstuff.com>

VANGIE, Beal Cloud computing (the cloud)(2013). (Consultado em Janeiro de 2015). Disponível no Web Site da webopedia : <http://www.webopedia.com>

JANSSEN, Cory (2012). SQL Server. (Consultado em Janeiro de 2015). Disponível no Web Site da Techopedia: <https://www.techopedia.com/definition/1243/sql-server>.

OPPEL, Andy (2004). [Databases Demystified](#). McGraw-Hill: [San Francisco, CA](#).

GOLDSTEIN, Jackie (2005). Renaissance Computer Systems. (Consultado em Agosto de 2015). Disponível no Web Site da Technet: [https://technet.microsoft.com/en-us/library/bb264565\(v=sql.90\).aspx](https://technet.microsoft.com/en-us/library/bb264565(v=sql.90).aspx)

[RANKINS](#), Ray (2013). Microsoft SQL Server 2012 Unleashed: Estados Unidos da América: SAMS.

ANTHONY, Molinaro (2005). SQL Cookbook. Gravenstein Highway North Sebastopol: O'Reilly

PORTNOY [Matthew](#) (2012). Virtualization Essentials. Estados Unidos da América: Wiley.

KUSNETZKY, [Dan](#)  
"http://www.amazon.com/Dan-Kusnetzky/e/B0060LSTCO/ref=dp\_byline\_cont\_book\_1"  
(2011).  
"http://www.amazon.com/Dan-Kusnetzky/e/B0060LSTCO/ref=dp\_byline\_cont\_book\_1"  
Virtualization: A Manager's Guide. Gravenstein Highway North Sebastopol: O'Reilly.

KAVIS, [Michael J.](#) (2014). Architecting the Cloud: Design Decisions for Cloud Computing Service Models (SaaS, PaaS, and IaaS). Hoboken New Jersey: Wiley.

BAHGA [Arshdeep](#),[Madisetti](#) [Vijay](#) (2013). Cloud Computing: A Hands-On Approach.

RHOTON, John (2009). Cloud Computing Explained: Implementation Handbook for Enterprises. Estados Unidos da América: RP.

JANSSEN, Cory (2012). SQL Server. (Consultado em Setembro de 2015). Disponível no Web Site da Techopedia: <https://www.techopedia.com/definition/1243/sql-server>.

JANSSEN, Cory (2014). Virtualization. (Consultado em Setembro de 2015). Disponível no Web Site da Techopedia: <https://www.techopedia.com/definition/719/virtualization>.

JANSSEN, Cory (2014). Cloud Computing. (Consultado em Setembro de 2015). Disponível no Web Site da Techopedia: <https://www.techopedia.com/definition/2/cloud-computing>.

THAI Thuan L., LAM Hoang (2013). .NET Framework Essentials. Gravenstein Highway North Sebastopol: O'Reilley.

KOROTKEVITCH, Dmitri (2014). Pro SQL Server Internals. New York: Apress

CHAPPELL, David (2006). Understanding .NET (2nd Edition). Indianapolis: Addison-Wesley

SINHA, Chandan (2014). Net Framework - Quick Reference Guide. .Net Interview Cracker

PORTNOY, Matthew (2012). Virtualization Essentials. Indianapolis: SYBEX

RUEST, Danielle , RUEST Nelson (2009). Virtualization, A Beginner's Guide. New York: Mc Graw Hill

KUSNETZKY, Dan (2011). Virtualization: A Manager's Guide. Gravenstein Highway North Sebastopol: O'Reilley.

KAVIS, Michael J. (2014). Architecting the Cloud: Design Decisions for Cloud Computing Service Models (SaaS, PaaS, and IaaS). New Jersey: Wiley

RHOTON, John (2009). Cloud Computing Explained: Implementation Handbook for Enterprises. Estados Unidos da América: RP