



Instituto Superior de Tecnologias Avançadas

Licenciatura Informática

Aplicação de Biblioteca Eletrónica

Realizado por João Carlos Rodrigues da Fonseca – Nº 1803

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

João Carlos Rodrigues da Fonseca

Novembro

2015

Agradecimentos

Agradeço à minha família, amigos e colegas de trabalho pela força e alento que sempre me deram para a conclusão desta licenciatura.

Ao Prof. Dr. Marco Costa, pela disponibilidade.

Agradeço ao orientador do projeto, Professor Doutor Pedro Brandão, pelo apoio e críticas construtivas dadas ao longo do projeto.

Aos restantes professores do ISTECC que contribuíram para enriquecer o meu conhecimento.

Resumo

Na sociedade em que vivemos, uma sociedade de Informação, as entidades e organizações necessitam, e dependem, do acesso à Informação de forma cada vez mais rápida e acessível. Essa Informação é facultada por Sistemas de Informação que têm vindo a ganhar importância, pelo que são encarados como recursos de origem tecnológica, mas também como parceiros estratégicos da operação diária de disponibilização eficiente de Informação às entidades que a usam.

Fazem parte dos 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 às suas necessidades, supere as suas expectativas.

Para uma organização ou entidade facultar os seus serviços de forma mais eficiente, é necessário que possua uma plataforma de disponibilização de conteúdos com a qual possa disponibilizar os serviços de que dispõem, criando assim um meio de comunicação com o utilizador, onde este pode usufruir dos mesmos de forma intuitiva.

Com o presente trabalho, pretende-se desenvolver e implementar um serviço que tem como principal objetivo a disponibilização de um conteúdo de informação (neste caso no formato *e-book*), através de um interface intuitivo que permita a procura e disponibilização dos mesmos a um utilizador registado na AD.

Palavras chave: *e-book*, AD.

Abstract

In today's world we live in an information society, entities and organizations need and depend on access to Information in an increasingly faster and accessible way. Information provided by these Information Systems has gained importance and is therefore seen as a technological resource but also as strategic partners in the daily operation for the efficient provision of information by the entities that use them.

Information Systems also include, Information Technologies which consist of applications that provide the services of an organization. These services should be available in a very effective way, with the main objective appreciation of the need for a client to provide you with service, which in addition to meet their needs, is made with the utmost attention and care.

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.

To an organization or entity providing its services more efficiently, it is necessary to possess an available platform of contents to which it can provide the services, thus creating a means of communication with the user where it can enjoy the same in a intuitive way.

The present work is intended to develop and implement a service that will have as a main objective the provision of an information content (in this case in e-book format) through an intuitive interface allowing the search and to make it available to a registered user.

Keywords: *e-book, AD.*

Abreviaturas

AD - Active Directory.

BI – Business Intelligence.

E-BOOK - Livro digital.

GUI – Interface gráfico do Utilizador.

IIS – Internet Information Services.

IP – Protocolo de Internet.

Login – Entrar numa aplicação ou Sistema.

Logoff – Sair de uma aplicação ou Sistema.

.png – Portable Network Graphics.

SQL – Structured Query Language.

SO – Sistema Operativo.

Standard – Padrão.

Password – Palavra passe.

T-SQL – Transact Structured Query Language.

TI – Tecnologia de Informação.

User - Utilizador

URL – Uniform Resource Locator.

ÍNDICE

Introdução	10
Estado de Arte	12
SQL Server	12
História da Linguagem de consulta Estruturada (SQL)	13
Tipos de comandos SQL.....	13
SQL Server Management Objects (SMO)	14
.NET Framework	15
Virtualização.....	16
Cloud computing	18
Características essenciais.....	19
Acesso à rede em banda larga.....	19
Elasticidade.....	19
Mensurável	19
Serviço Self-Service	19
Conjunto de recursos	19
Modelos de serviço	20
Infraestrutura como serviço (IaaS)	20
Plataforma como serviço (PaaS).....	20
Software como serviço (SaaS).....	20
Contextualização.....	21
Desenvolvimento	22
Desenvolvimento empírico da aplicação	26
Testes de usabilidade	72
Conclusão	74
Referências Bibliográficas.....	76

ÍNDICE DE FIGURAS E GRÁFICOS

Figure 1 -Diagrama da arquitectura Logica	24
Figure 2 - library	26
Figure 3 - Variaveis da aplicação	27
Figure 4 - Focus	27
Figure 5 - Maxlenght list	28
Figure 6 - RESET CONTRO LIST	28
Figure 7 - ENABLE/DISABLE CONTROL LIST	28
Figure 8 - Error Handling	29
Figure 9 - ToolTip	29
Figure 10 - Image Sizing	30
Figure 11 - Character counter	30
Figure 12 - Panel	31
Figure 13 - Panel control	31
Figure 14 – E-book parameters	32
Figure 15 - Grid init	33
Figure 16 – Grid row Data	34
Figure 17 - grid command	35
Figure 18 - Image array	36
Figure 19 - Book Find	36
Figure 20 - Alphabar Event	37
Figure 21 - Create button	37
Figure 22 - Alphabar	38
Figure 23 - Alphabar trigger	38
Figure 24 - App Close	38
Figure 25 - Add book	39
Figure 26 - Cancel book	39
Figure 27 - Form Load	40
Figure 28 - Autor event	41
Figure 29 - Control focus event	41
Figure 30 - cover load	42
Figure 31 - file check	42
Figure 32 - check content	43
Figure 33 - Book save	44
Figure 34 - book class	44
Figure 35 - Dataitem class	45
Figure 36 - book delete	46
Figure 37 - book get by id	47
Figure 38 - book get	48
Figure 39 - book update or insert	49
Figure 40 - adoparameter class	50
Figure 41 - Tadostoredprocedure class	51
Figure 42 - parameter load	52
Figure 43 - get data	53
Figure 44 - create data	54
Figure 45 - user class	55
Figure 46 - ad authentication	56
Figure 47 - user validation	57
Figure 48 - table properties	58

Figure 49 - procedure del book by id.....	58
Figure 50 - procedure get book by id.....	59
Figure 51 - procedure update book	60
Figure 52 - procedure get book by id.....	60
Figure 53 - books table	61
Figure 54 - Frontend	62
Figure 55 - code behind	62
Figure 56 - filtered view	63
Figure 57 - book view.....	64
Figure 58 - e-book view.....	65
Figure 59 - book removal	66
Figure 60 - remove success message	67
Figure 61 - alpha bar.....	67
Figure 62 - add book view.....	68
Figure 63 - filter view	69
Figure 64 - search type view.....	70
Figure 65 - app exit.....	71
Figure 66 - etapas de testes (Kekare Harshada, http://www.buzzle.com/ 2013).....	72
Figure 67 - RESULTADO DOS TESTES.....	73

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.

A realização deste trabalho tem como objetivo o desenvolvimento de um Serviço de forma a facultar uma plataforma de pesquisa e acesso de utilizadores de uma biblioteca.

Os objetivos específicos são: existência 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 generalista entre várias entidades privadas e públicas, quer na área mais comum de Bibliotecas, quer de sistemas de utilização doméstica para organização de conteúdos digitais, foi possível reconhecer um vasto leque de abordagens para o nosso objetivo.

A escolha do tema para o Projeto Global deriva assim, antes de mais, de uma primeira sugestão por parte do orientador do projeto, O Professor Doutor Pedro Brandão, e também de uma procura sobre o tema nos meios disponíveis.

A metodologia usada para abordar este trabalho, passou por uma pesquisa bibliográfica sobre os temas a analisar e que seriam usados 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 apresentada no Estado de Arte. Seguidamente será apresentado o levantamento de requisitos considerados necessários para a implementação do projeto, tais como usabilidade, permissões e manipulação de dados e desenho do "frontend".

No seguimento, é apresentado o desenvolvimento da aplicação. Aqui é utilizado o Microsoft Visual Studio para desenvolver o "frontend" e o SQL Server que serve o "backend" da nossa aplicação, onde se faz a introdução e o armazenamento da informação propriamente dita.

Após a conclusão destas etapas, ir-se-á desenvolver um conjunto de testes de utilização e regressão ao projeto, por forma a validar que os objetivos propostos são atingidos.

O presente projeto foi dividido em três grandes macro blocos, que se descrevem de seguida.

Numa primeira parte será executada a pesquisa bibliográfica de forma a executar o Estado de Arte. Esta fase leva à 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.

Numa segunda parte, encontramos o desenvolvimento propriamente dito da aplicação, onde são apresentados todos os passos necessários para a conclusão do projeto e as ferramentas usadas no mesmo.

Na terceira parte são apresentados os resultados e conteúdos da bateria de testes de usabilidade e regressão executados contra a versão final do projeto.

Para concluir o trabalho, será ainda apresentada uma conclusão com base em todo o trabalho elaborado e a experiência adquirida durante o mesmo, incluindo potenciais contratempos e incidentes encontrados na obtenção dos requisitos definidos.

ESTADO DE ARTE

SQL SERVER

A Microsoft trabalhou inicialmente com a Sybase Corporation numa versão do SQL Server para OS/2¹. Quando a Microsoft abandonou o sistema OS/2 em favor de seu novo sistema operativo, Windows NT², decidiu reforçar o mecanismo do SQL Server e ajudar a modificar o código para o Windows NT. O produto resultante foi o Microsoft SQL Server 4 para Windows NT. (Janssen Cory , 2012)

Ao longo do tempo, a Microsoft assumiu a responsabilidade do mais desenvolvimento do SQL Server. Com a versão 6, a Microsoft estava no controle completo do software. Engenheiros da Sybase continuaram a desenvolver a sua plataforma de base de dados para ser executada em Windows NT, enquanto a Microsoft continuou a melhorar o SQL Server 6.

Hoje em dia a Microsoft está entre os melhores criadores de software, e o SQL Server é um dos principais estandartes da Microsoft no mundo das base de dados, trazendo mais facilidade de utilização de um sistema de base de dados, invés da total dependência do ANSI SQL. Fornece um conjunto de ferramentas de administração e uma extensão da linguagem SQL chamada T-SQL (Transact SQL), em que são tratados os dois da mesma maneira. (Molinaro Anthony, 2005)

¹ <http://techland.time.com/2012/04/02/25-years-of-ibms-os2-the-birth-death-and-afterlife-of-a-legendary-operating-system/>

² http://www.theregister.co.uk/2013/08/20/nt_at_20/

HISTÓRIA DA LINGUAGEM DE CONSULTA ESTRUTURADA (SQL)

Durante a década de 1970, a IBM inventou uma linguagem de computador projetado especificamente para consultas a base de dados, esta linguagem foi chamada SEQUEL (Structured English Query Language). A sigla SEQUEL foi posteriormente alterada para SQL porque "SEQUEL" é uma marca comercial da empresa de aeronaves Hawker Siddeley, sediada no Reino Unido.

Com o passar do tempo, a língua foi ampliada, de tal modo que não é apenas uma linguagem para consultas, mas também pode ser usada para criar bases de dados e gerir a segurança do motor de base de dados. Atualmente, são diferentes do SQL em uso. Microsoft SQL Server usa uma versão chamada T-SQL, ou Transact-SQL. (Oppel Andy, 2004)

TIPOS DE COMANDOS SQL

Linguagem de definição de dados (Data Definition Language/DDL), permite a criação, restruturação e eliminação de objetos da base de dados.

Principais comandos: CREATE, ALTER e DROP

Linguagem de manipulação de dados (Data Manipulation Language/DML), permite manipular dados nos objetos da base de dados.

Principais comandos: INSERT, UPDATE e DELETE.

Linguagem de consulta de dados (Data Query Language/DQL), usado para efetuar consultas sobre a base de dados, pode conter varias opções e clausulas.

Único comando: SELECT

Linguagem de controlo de dados (Data Control Language/DCL), usado para permitir o acesso aos dados na base de dados.

Principais comandos: GRANT e REVOKE.

Esquema de Base de Dados os comandos, também conhecidos com a linguagem de definição de dados (DDL), são usados para definir o banco de dados, tabelas e outros objetos que foram utilizados para gerir a estrutura da base de dados.

Comandos para o gerir os dados, também conhecida como linguagem de manipulação de dados (DML), são muito utilizados como o original comandos SQL, que manipulam, adicionam ou excluem os dados na base de dados. (Goldstein Jackie, 2005)

SQL SERVER MANAGEMENT OBJECTS (SMO)

Os objetos de gestão do SQL Server, são uma biblioteca .Net, que permite o acesso e a gestão de todos os objetos do SQL Server.

Existe também bibliotecas para replicação (Replication Management Objects [RMO]) e para automatizar tarefas para gerir os serviços de análise (Analysis Management Objects [AMO]).

O objetivo da API SMO é fornecer um interface de programação completo e fácil de usar e com o objetivo de automatizar, administrar e manter o SQL Server.

Pode-se usar qualquer linguagem de programação .NET, com o Delphi .Net ou C#, para realizar as mesmas tarefas que se consegue usado o Management Studio. (Rankins Ray, 2013)

A API SMO oferece três tipos de classes:

- Classe de Instância, fornecem o acesso a todos os objetos do SQL Server e suas propriedades. Exemplos de instância classe incluem Server, banco de dados, tabelas, exibições e procedimentos armazenados.
- Classe utilitária, podem ser usadas para tarefas de gestão e administração, tais como backup, restauração, postos de trabalho, o Database Mail e gestão de Service Broker.
- Classe de Script, fornece a funcionalidade de scripting avançado.

.NET FRAMEWORK

O .NET é um framework de desenvolvimento de software da Microsoft que fornece um ambiente de programação controlada em que o software pode ser desenvolvido, instalado e executado em sistemas operacionais baseados em Windows.

O .NET é uma peça central da estratégia de desenvolvimento da Microsoft além de ser um elemento fulcral na competição com o Java. (Chappell David, 2006)

As principais características do projeto são:

- Interoperabilidade: Permite que programas desenvolvidos em .NET acedam às funcionalidades nos programas desenvolvidos fora de .NET.
- Runtime comum: Também conhecida como Common Language Runtime (CLR), permite que os programas desenvolvidos em .NET exibam comportamentos comuns no uso de memória, tratamento de exceções e segurança.
- Independência de idiomas: A especificação Common Language Infrastructure (CLI) permitir o intercâmbio de tipos de dados entre dois programas desenvolvidos em diferentes idiomas.

- Base Class Library: Uma biblioteca de código para funções mais comuns - usado por programadores para evitar reescrever trechos repetitivos de código.
- Facilidade de implantação: Existem ferramentas para garantir a facilidade de instalação de programas, sem interferir com as aplicações instaladas anteriormente.
- Segurança: Programas desenvolvidos em .NET são baseados em um modelo de segurança comum.

VIRTUALIZAÇÃO

A virtualização é a criação de uma versão virtual (em vez de real) de algo, tais como um sistema operativo, um servidor, um dispositivo de armazenamento ou recursos da rede.

A virtualização do sistema operativo é a utilização de software para permitir que um pedaço de hardware possa executar várias imagens de sistemas operativos ao mesmo tempo. A tecnologia teve o seu início em mainframes, décadas atrás, permitindo que administradores de sistemas encontrassem uma forma de evitar o desperdício de energia e de processamento, algo extremamente caro na altura. (Portnoy Matthew , 2012)

Em 2005, o software de virtualização foi adoptado de forma mais rápida do que se imaginava, incluindo pelos especialistas da área. Há três áreas de TI onde a virtualização se encontra a fazer grandes avanços: virtualização de rede, virtualização de armazenamento e virtualização de servidores.

A virtualização de rede é um método de combinar os recursos disponíveis de uma rede, dividindo-se a largura de banda disponível para canais, cada um dos quais é independente dos outros, e cada um dos quais pode ser atribuído (ou transferido) para um servidor ou dispositivo em particular, em tempo real. A ideia passa por colocar a virtualização a dissimular a verdadeira complexidade da rede, separando-a em partes geríveis, muito parecida com a técnica de particionar um disco rígido de forma a tornar a gestão de arquivos mais simples.

A virtualização de armazenamento é o agrupamento de armazenamento físico de vários dispositivos de armazenamento de rede, para o que parece ser um único dispositivo de armazenamento que é gerido a partir de uma consola central. A virtualização de armazenamento é habitualmente usada em redes de armazenamento local (SANs). (Kusnetzky Dan , 2011)

A virtualização de servidores é a obsfucação de recursos do servidor (incluindo o número e identidade dos servidores físicos individuais, processadores e sistemas operacionais) de utilizadores do servidor. A intenção é poupar o utilizador de ter que entender e gerir os detalhes complexos dos recursos do servidor, aumentando a partilha de recursos, utilização e manutenção da capacidade de expansão a posteriori.

A virtualização pode ser vista como parte de uma tendência geral nas empresas de TI, que inclui computação autónoma, um cenário em que os ambientes de TI serão capazes de se gerir a si mesmo com base na atividade percepionada, e "utility computing", em que o poder de processamento do servidor é visto como um recurso que os clientes podem pagar apenas quando necessário. O objectivo habitual da virtualização é centralizar as tarefas administrativas, melhorando a escalabilidade e as cargas de trabalho. (Kusnetzky Dan , 2011)

CLOUD COMPUTING

A definição de cloud computing do NIST (National Institute of Standards and Technology) refere que este é um modelo de computação que permite o acesso ininterrupto a um conjunto de recursos disponibilizados em rede, tais como: armazenamento, redes de comunicações, servidores, aplicações e serviços, que podem ser rapidamente distribuídos e atualizados com um mínimo esforço.

O conceito de cloud computing, é a capacidade de computação ilimitada disponível e flexível na nuvem, é tudo aquilo que fica nos bastidores da ligação. (Woodford Chris, 2014)

Todos os problemas advindos da largura de banda, armazenamento, processamento, fiabilidade ou segurança são postos de parte.

As respostas às necessidades em termos de tecnologias de informação serão satisfeitas com uma ligação a nuvem.

Muitas empresas estão entregando serviços da nuvem:

- Google, tem uma nuvem privada que usa para a entrega de muitos serviços diferentes para seus usuários, incluindo acesso a email, aplicações de documento, traduções de texto, mapas, web analytics e muito mais.
- Microsoft, tem o Microsoft Sharepoint serviço online que permite a inteligência de conteúdo e de negócios movido para a nuvem. A Microsoft atualmente disponibiliza os seus aplicativos do office na nuvem.
- Salesforce executa seu aplicativo definido para seus clientes em uma nuvem, e seus produtos Force.com e Vmforce.com fornecem aos desenvolvedores com plataformas para construir serviços de nuvem personalizada.

(Kavis Michael J., 2014)

CARACTERÍSTICAS ESSENCIAIS

ACESSO À REDE EM BANDA LARGA

Os recursos estão disponíveis através da rede e podem ser acedidos de qualquer lugar a qualquer hora e através de diferentes plataformas.

ELASTICIDADE

A alocação de recursos podem ser feita rapidamente e de forma automática, para responder às necessidades dos consumidores. Da mesma forma, a libertação dos mesmos acontece quando já não são necessários.

MENSURÁVEL

Os recursos são controlados e otimizados de forma automática, efetuando para esse efeito várias medições a nível de processamento efetuado, armazenamento utilizado, largura de banda utilizado.

O uso dos recursos deve poder ser monitorizado e controlado de forma transparente, tanto para o fornecedor, como para o consumidor do serviço utilizado.

SERVIÇO SELF-SERVICE

Um utilizador pode dispor de forma unilateral e automática, recursos de computação (tempo de servidor ou armazenamento em rede) conforme necessário e de forma automática. Tudo sem necessidade de interação humana com o fornecedor do serviço.

CONJUNTO DE RECURSOS

Os recursos computacionais do fornecedor do serviço são reunidos para servir vários utilizadores, usando um modelo de múltiplos utilizadores, com diferentes recursos físicos e virtuais, atribuídos dinamicamente de acordo com os pedidos destes últimos. Há um sentido de independência de localização em que geralmente o cliente não tem nenhum controlo ou conhecimento sobre o local exato dos recursos fornecidos, mas pode ser capaz de especificar

o local num nível mais elevado de abstração (por exemplo, país, estado ou centro de dados). Armazenamento, processamento, memória e largura de banda de rede são exemplos destes recursos. (Rhoton John, 2009)

MODELOS DE SERVIÇO

Os recursos distribuídos na computação em nuvem existem sob forma de serviços, dos quais podemos subscrever: Software como serviço (SaaS), a plataforma como serviço (PaaS) e infraestrutura como serviço (IaaS).

Estes três tipos de serviço diferem no controle que se tem sobre a informação, e por outro lado, no que se pode esperar que o fornecedor possa fazer por si.

INFRAESTRUTURA COMO SERVIÇO (IAAS)

Infraestrutura como serviço (IaaS) - consumidores tem a capacidade de controlar e gerir os sistemas operacionais, aplicativos, armazenamento e conectividade de rede, mas não pode controlar a infraestrutura na nuvem.

PLATAFORMA COMO SERVIÇO (PAAS)

Plataforma como serviço (PaaS) - os consumidores compra o acesso às plataformas, permitindo-lhes implantar seus próprios softwares e aplicativos na nuvem.

Os sistemas operacionais, armazenamento, servidores e o acesso à rede não são gerenciados pelo consumidor, mas controla as aplicações implementadas.

SOFTWARE COMO SERVIÇO (SAAS)

Software como serviço (SaaS) - é a capacidade de o consumidor ter acesso e usar aplicativos ou serviço que estão hospedados na nuvem.

As aplicações são acessíveis ao cliente a partir de vários dispositivos ou plataformas.

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 então um período de procura de exemplos onde nos pudéssemos apoiar e recolher ideias para o projeto. Esta pesquisa passou pela utilização de diversos softwares de armazenamento de e-books para uso privado, 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 massificação de informação disponível a cada momento e o aumento drástico de novas publicações, tanto as bibliotecas tradicionais como as bibliotecas privadas de consumidores mais exigentes, testemunharam um desafio enorme. Esse desafio passava 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 o mesmo: Algumas baseadas nos anteriores sistemas de organização bibliotecário (tal como o Sistema decimal de Dewey³), outros em sistemas completamente informatizados, baseados em plataformas de serviços como o Sharepoint⁴ (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 pretendeu a construção de numa plataforma que permitisse algo diferente da simples indexação de resultados de livros, que permitisse algo mais do que apenas a disponibilização de dados básicos tais como título, editora, escritor etc.. Pretendíamos assim que a nossa aplicação permitisse 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,

³ https://www.oclc.org/content/dam/oclc/services/brochures/211422usb_dewey_services.pdf

⁴ <https://products.office.com/pt-pt/SharePoint/sharepoint-2013-overview-collaboration-software-features>

fosse possível a manipulação, de forma funcional, de todos os dados numa base de dados persistente.

DESENVOLVIMENTO

Nesta fase do trabalho são apresentadas as etapas que constituem o desenvolvimento e operação da aplicação. Essas etapas passam pelo levantamento de requisitos, testes de usabilidade e regressão por exemplo. São ainda apresentados vários conceitos usados no desenvolvimento deste projeto.

A ideia de desenvolver esta aplicação em particular, surgiu no seguimento da hipótese da escolha de tema para o Projeto Global que nos foi apresentada pelo Orientador do Projeto.

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.

zado 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.

Architecture and Requisite Gathering

O conceito por trás da implementação deste projeto, passa pela ideia de integração entre uma biblioteca virtual e a sua congénere física. Para tal será facultado um Frontend gráfico (Aplicação) para um utilizador que ao autenticar-se no Domínio dessa mesma biblioteca (física) poderá 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.⁵

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.⁶
- 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 2014.⁷⁸
- Clientes (múltiplos) de Windows 8, servindo de host para a aplicação.⁹

⁵ [https://technet.microsoft.com/en-us/library/cc782657\(v=ws.10\).aspx](https://technet.microsoft.com/en-us/library/cc782657(v=ws.10).aspx)

⁶ <https://technet.microsoft.com/en-us/library/hh801901.aspx>

⁷ <http://blogs.technet.com/b/dataplatforminsider/archive/2013/06/05/sql-server-2014-a-closer-look.aspx>

⁸ <https://technet.microsoft.com/en-us/windowsserver/dd448604.aspx>

⁹ <https://technet.microsoft.com/en-us/library/hh832030.aspx>

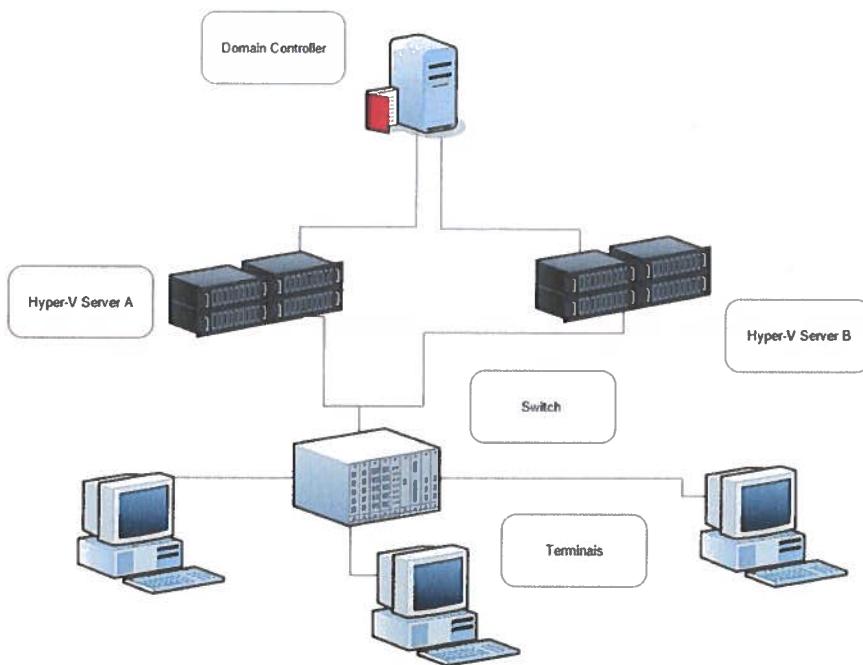


FIGURE 1 -DIAGRAMA DA ARQUITECTURA LOGICA

Sendo assim, um utilizador da biblioteca chegaria a um conjunto de terminais disponíveis (PC físico, por exemplo), procederia ao seu login com uma conta pré criada e distribuída pela Administração da mesma Biblioteca, e ao logar-se, realiza a validação do perfil do user. Caso não seja reconhecido com perfil de administração, apenas acede a uma vista que se resume a consulta, sem qualquer possibilidade de manipulação, apenas possível ao acesso com 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"¹⁰ que esteja 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 Admi-

¹⁰ <http://www.acsu.buffalo.edu/~fineberg/mfc158/week10lecture.htm>

nistrador, executa a mesma) a referida informação na tabela de conteúdo previamente populada.

Apos 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, se encontram virtualizados correndo sobre a Plataforma VMware Workstation¹¹. Numa implementação real, tais serviços seriam realizados em servidores físicos isolados e, em última instancia, os clientes terminais poderiam ser apenas clientes Thin client¹² de forma a aceder ao Sistema Operativo virtualizado disponível pelos Servidores Hyper-V.

¹¹ https://pubs.vmware.com/Release_Notes/en/workstation/12pro/workstation-1201-release-notes.html

¹² <http://searchnetworking.techtarget.com/definition/thin-client>

DESENVOLVIMENTO EMPÍRICO DA APLICAÇÃO

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

As imagens contém parte do *code-behind* e 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.

```
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 2 - 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 da Telerik.

Estes controlos são de sobejá importância pois permitem-nos costumizar de forma extremamente ágil quer os temas / *layout* do "frontend" quer aceder a customização a nível dos botões de seleção e caixas de seleção.

Estas são as bibliotecas usadas na aplicação, sendo um projeto da Telerik, vêm de raiz com bibliotecas próprias. Os controlos da "framewok" de DevCraft¹³ da Telerik, são de sobejia importância pois permitem-nos ser muito mais versáteis que os controlos¹⁴ 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 3 - VARIÁVEIS DA APLICAÇÃO

Declaração¹⁵ de variáveis da aplicação.

```
private void goto_Forward(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 4 - FOCUS

Esta função permite colocar o *focus*¹⁶ no objeto pretendido.

¹³ <http://www.telerik.com/devcraft> devcraft

¹⁴ <https://msdn.microsoft.com/en-us/library/windows/desktop/dn742484%28v=vs.85%29.aspx>

¹⁵ <https://msdn.microsoft.com/en-us/library/office/gg264241.aspx>

¹⁶ <http://markhobley.yi.org/glossary/mousefocus.html#sloppyfocus>

```

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 5 - MAXLENGTH LIST

Esta função permite parametrizar a propriedade “MaxLength” do objeto com o tamanho 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 6 - RESET CONTROL LIST

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

A função em causa é chamada por exemplo no *use case*¹⁷ 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(RadButton)) ((RadButton)ctrl).Enabled = onoff;
        else if (ctrl.GetType() == typeof(RadTextBox)) ((RadTextBox)ctrl).Enabled = onoff;
    }
}

```

FIGURE 7 - ENABLE/DISABLE CONTROL LIST

Esta função é usada para *enabled / disable* dos objetos passados por valor. A função é usada no caso de desejarmos ver os dados de um e-book.

¹⁷ <http://alistair.cockburn.us/Use+cases%2c+ten+years+later>

```

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, msgBtn, msgIcon, MessageBoxButtons.DefaultButton.Button1);
    }
}

```

FIGURE 8 - ERROR HANDLING

Esta função é usada para controlo de "error handling"¹⁸ e 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(RadButton)) ((RadButton)ctrl).ButtonElement.ToolTipText = tooltip;
}

```

FIGURE 9 - TOOLTIP

A função writeToolTip¹⁹ permite parametrizar o objeto inicialmente de forma a passar um valor de texto para a "label"²⁰ que é visualizada através da ação "mouse hover"²¹. A função é usada no arranque da aplicação para inicializar os botões da barra de navegação.

¹⁸ [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)

¹⁹ <http://www.telerik.com/products/aspnet-ajax/tooltip.aspx>

²⁰ <http://www.telerik.com/help/winforms/panels-and-labels-label-overview.html>

²¹ <http://www.w3.org/TR/html5/dom.html#the-title-attribute mouse hover>

```

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 10 - IMAGE SIZING

Na função `getResizedImage` encontramos o código usado para fazer o 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"²².

```

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

```

FIGURE 11 - CHARACTER COUNTER

A função `writeText` permite informar o utilizador, através de uma `RadLabel` da quantidade de caracteres que foi ou poderá ser introduzida em uma `RadTextBox`, passada por valor. Essa função é usada na introdução de um novo e-book.

²² <http://www.linfo.org/gui.html>

```

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

```

FIGURE 12 - 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 13 - PANEL CONTROL

A função de `collapsePannel`, é responsável pelo tratamento da apresentação visual dos painéis^{23, 24}.

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

²³ <http://publib.boulder.ibm.com/cgi-bin/bookmgr/BOOKS/F29BDG00/GLOSSARY>

²⁴ [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 14 – 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 15 - GRID INIT

A função gridMode, permite a inicialização da "grid"²⁵, consoante o tipo de utilizador em uso.

A função é usada no arranque na aplicação, para configurar as várias propriedades e eventos

²⁵ <http://docs.telerik.com/devtools/wpf/controls/radgridview/overview2>

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;
        collapsePannel(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 });
            collapsePannel(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);

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

FIGURE 16 – 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 passa a modo de edição para utilizadores administradores.

```

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.RowInfo.Cells["TxtID"].Value.ToString());

    if (clickedCmdColumn.ColumnInfo.Name.Equals("CmdPdf"))
    {
        row_edit = -1;
        collapsePanel(new List<int> { 1 }, sContainer, true);
        if (row_pdf.Equals(Int32.Parse(clickedCmdColumn.RowInfo.Cells["TxtID"].Value.ToString())))
        {
            row_pdf = -1;
            collapsePanel(new List<int> { 2 }, sContainer, true);
        }
        else
        {
            row_pdf = row_id;
            List<Book> lstbook = funcos.get_Book_id(row_id, ref errorMsg);
            if (!string.IsNullOrEmpty(errorMsg))
                writeStatus(lblError, errorMsg, Color.Maroon, true, true, MessageBoxButtons.OK, RadMessageIcon.Error);
            else
            {
                collapsePanel(new List<int> { 2 }, sContainer, false);
                pdfViewer.LoadDocument(new MemoryStream(lstbook[0].book_pdf));
            }
        }
    }
    else if (clickedCmdColumn.ColumnInfo.Name.Equals("CmdDel"))
    {
        RadMessageBox.SetThemeName(funcos.c_ThemeName);
        DialogResult dr = RadMessageBox.Show("Deseja remover livro?", funcos.c_AppName, MessageBoxButtons.OKCancel, RadMessageIcon.Question, MessageBoxButtonsDefaultButton.Button1);
        if (dr.ToString().ToUpper() .Equals("OK"))
        {
            row_edit = row_pdf = -1;
            collapsePanel(new List<int> { 1, 2 }, sContainer, true);
            funcos.del_book(row_id, ref errorMsg);
            if (!string.IsNullOrEmpty(errorMsg))
                writeStatus(lblError, errorMsg, Color.Maroon, true, true, MessageBoxButtons.OK, RadMessageIcon.Error);
            else
            {
                clickedCmdColumn.RowInfo.Delete();
                writeStatus(lblError, "Registro apagado com sucesso!", Color.Green, true, false, MessageBoxButtons.OK, RadMessageIcon.None);
            }
        }
    }
}

```

FIGURE 17 - 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"²⁶ 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 apenas pode ser executada através de uma caixa de diálogo²⁷ de confirmação.

²⁶ [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)

²⁷ <http://docs.telerik.com/devtools/wpf/controls/radwindow/features/predefined-dialogs>

```

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 18 - 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. 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, txtAutor, txtNome, txtDescricao, bdtCapa, bdtPdf, imgCapa });
    collapsePanel(new List<int> { 1, 2 }, sContainer, 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, MessageBoxDefaultButton.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, MessageBoxDefaultButton.Button1);
        ddFilter.ShowDropDown();
        return;
    }

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

    if (App_user.Is_Admin)
        gridMode(gridBooks, 1stCol_Admin);
    else
        gridMode(gridBooks, 1stCol_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 19 - 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 20 - 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 navegação²⁸.

```
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.EnableFocusing = 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 = 0;
        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 21 - CREATE BUTTON

A função `makeButton` permite criar dinamicamente botões do tipo `RadButton`²⁹. Esta função é usada para construir a barra do abecedário.

²⁸ <https://msdn.microsoft.com/en-us/library/windows/apps/xaml/dn440584.aspx>

²⁹ <http://www.telerik.com/help/winforms/buttons-button-overview.html>

```

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 = 0; i <= 9; 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, 2), 26, 26, string.Format("Letra {0}", (char)i),
            DockStyle.None));
    LstButtons.Add(makeButton(string.Format("btn_0", (char)55), string.Empty, "", string.Format("[0]", (char)55),
        string.Format("{0}letter_{1}", _path, (char)55), new Point(nbtn++ * btnspace + btnstart, 2), 26, 26, string.Format("Tudo"), 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)
        pnlAlfa.Controls.Add(btn);
    ThemeResolutionService.ApplyThemeToControlTree(this, funcoes.c_ThemeName);
}

```

FIGURE 22 - ALPHABAR

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 23 - 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, RadMessageIcon.Question, MessageBoxButtons.DefaultButton1);
    if (dr.ToString().ToUpper() == "OK") Application.Exit();
}

```

FIGURE 24 - APP CLOSE

A função `btnExit_Click`, permite ao clicar no botão de On/Off localizado no topo esquerdo da barra de navegação, a execução do 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, bedtCapa, bedtPdf, 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 25 - 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 26 - CANCEL BOOK

A função `btnCancel_Click` por sua vez, cancela a inserção de um livro invocada pela função `btnAddBook_Click`, fechando o painel e inicializando a variável³⁰ `new_book`.

³⁰ https://www.le.ac.uk/users/rjm1/cotter/page_20.htm

```

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

    OpenFileDialog dialog = (OpenFileDialog)bedtCaps.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();
    collapsePanel(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");

    pnAlfa.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 27 - FORM LOAD

A função `mainForm_Load` permite a inicialização dos objetos do "Form"³¹ no arranque do programa. Essa inicialização faz o *reset* das "Labels"³² e carrega as "Combo Boxes"³³ 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 AD³⁴ e por final define o tema de arranque do "Form"³⁵.

³¹ <http://www.telerik.com/help/winforms/forms-and-dialogs-overview.html>

³² <http://www.telerik.com/help/winforms/panels-and-labels-label-overview.html>

³³ <http://www.telerik.com/help/winforms/dropdown-and-listcontrol-dropdownlist-overview.html>

³⁴ [https://msdn.microsoft.com/en-us/library/aa746492\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/aa746492(v=vs.85).aspx)

³⁵ [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)

```

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 28 - AUTOR EVENT

A imagem anterior apresenta os eventos associados as caixas de texto, referentes aos dados do ebook.

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 29 - CONTROL FOCUS EVENT

O evento³⁶ 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.

³⁶ [https://msdn.microsoft.com/en-us/library/17sde2xt\(v=VS.100\).aspx](https://msdn.microsoft.com/en-us/library/17sde2xt(v=VS.100).aspx)

```

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 30 - COVER LOAD

A função `bedtCapa_ValueChanged` permite validar se a *path*³⁷ 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, RadMessageBoxIcon.Error, MessageBoxButtons.DefaultButton1);
            }
        }
    }
    catch { result = false; }
    return result;
}

```

FIGURE 31 - 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*.

³⁷ [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 32 - CHECK CONTENT

A função `checkField_Empty` valida se os campos preenchidos pelo utilizador administrador se encontram vazios e, no 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(txtNome, "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, txtNome.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, txtNome, txtDescricao, bedtCapa, bedtPdf, imgCapa });
        writeStatus(lblError, "Registo adicionado com sucesso.", Color.Green, true, false, MessageBoxButtons.OK, RadMessageIcon.None);
    }
}

```

FIGURE 33 - 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 34 - BOOK CLASS

A classe³⁸ `book` define todos os parâmetros de um *e-book*, nomeadamente os atributos de id, nome, descrição, autor, o *e-book* do livro e a imagem da sua capa.

³⁸ <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_FindFilter()
    {
        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 35 - DATAITEM CLASS

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

```

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>();
        TADOStoredProc adoSpP = new TADOStoredProc("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 36 - 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"³⁹ `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, erro esse que é construído através da mensagem que é capturada pela exceção, fazendo referência ao método que foi invocado.

³⁹ <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>();
        TADOStoredProc adoSpP = new TADOStoredProc("get_BOOK_BY_ID", tempconn);
        adoSpP.addParam(new TADOParParam("@id", id));
        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"))),
                                    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 37 - 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, erro esse que é construído através da mensagem que é capturada pela exceção fazendo referência ao método que foi invocado.

```

public static List<book> get_Book(string field, string filter, string text, 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_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 = "Error [ get_Book ] - " + ex.Message;
        return null;
    }
    finally
    {
        if (tempconn.State == ConnectionState.Open) tempconn.Close();
    }
}

```

FIGURE 38 - 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>();
        TADOSignedProcedure adoSpP = new TADOSignedProcedure("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 39 - 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 é 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, erro esse que é construído através da mensagem que é capturada pela exceção fazendo referência ao método que foi invocado.

```

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 40 - ADOPARAMETER CLASS

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

```
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 41 - TADOSTOREDPROCEDURE CLASS

A classe TADOStoredProcedure inicializa a ligação à 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.dataTables = createDataTable(sqlComm.ExecuteReader(), visibleFields);
    }
}

```

FIGURE 42 - 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 SQL⁴⁰.

⁴⁰ [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)

```
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 43 - GET DATA

O método `getDataTable`⁴¹ retorna todos os dados adquiridos da base de dados.

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

⁴¹ <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 44 - CREATE DATA

O método CreateDataTable devolve os dados retornados pela storeprocedure, no formato de uma variável do tipo DataTable⁴².

⁴² [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)

```

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 45 - USER CLASS

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 e no caso do mesmo não se encontrar nele (estamos a utilizar o "Windows Authentication"⁴³ para tal) a aplicação fecharia indicando o erro de autenticação.

⁴³ <https://msdn.microsoft.com/en-us/library/ff647396.aspx>

```

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 46 - AD AUTHENTICATION

Este conjunto de funções é usado especificamente para validar o utilizador (e seus atributos) comunicando diretamente com a AD, usando para isso a função PrincipalContext⁴⁴ ⁴⁵ ⁴⁶, UserPrincipal⁴⁷ e a GroupPrincipal⁴⁸.

⁴⁴ [https://msdn.microsoft.com/en-us/library/bb356158\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/bb356158(v=vs.110).aspx)

⁴⁵ [https://msdn.microsoft.com/en-us/library/system.directoryservices.accountmanagement.contexttype\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.directoryservices.accountmanagement.contexttype(v=vs.110).aspx)

⁴⁶ [https://msdn.microsoft.com/en-us/library/system.directoryservices.accountmanagement.contextoptions\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.directoryservices.accountmanagement.contextoptions(v=vs.110).aspx)

⁴⁷ [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)

⁴⁸ [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)

```

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 47 - USER VALIDATION

O método chekIsAdmin valida 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 à aplicação em modo de consulta.

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 48 - TABLE PROPERTIES

A aplicação contém uma tabela única que os atributos de:

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

Sendo que tanto as imagens como os *e-books* são guardados num formato "varbinary"⁴⁹.

```

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

```

FIGURE 49 - PROCEDURE DEL BOOK BY ID

⁴⁹ [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.

```
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 50 - 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 51 - PROCEDURE UPDATE BOOK

A "procedure" INS_UPD_BOOK é utilizada quer para inserir um livro e toda a sua informação por completo, ou para atualizar um ou 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 52 - 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 53 - BOOKS TABLE

"Procedure" de criação da tabela de dados. A tabela de dados contém todos os campos utilizados pela aplicação para armazenar a informação para o objeto Livro.

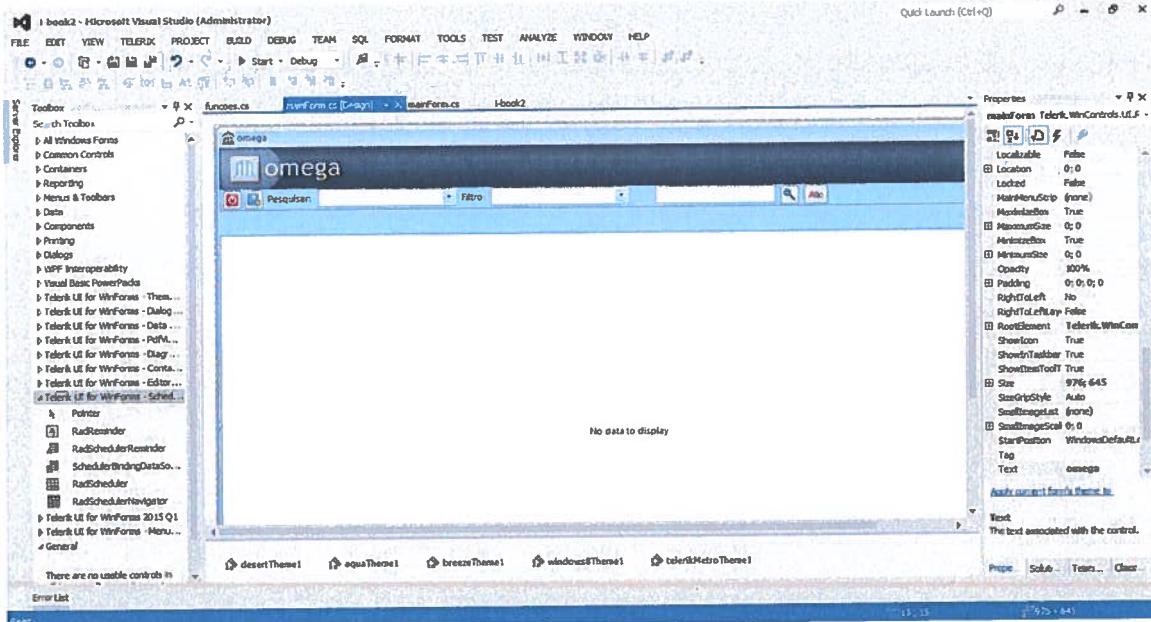


FIGURE 54 - FRONTEND

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

```

using Telerik.WinControls;
using Telerik.WinControls.UI;
using Telerik.WinControls.Themes;
using Telerik.WinControls.Styles;

using i_book2;

namespace i_book2
{
    public partial class mainForm : Telerik.WinControls.UI.RadForm
    {
        public mainForm()
        {
            InitializeComponent();
        }

        string errorMessage = string.Empty;
        string themeName = "TelerikMetro";
        userApp user = null;
        List<string> listBookAdmin = new List<string> { "id", "name", "description", "text", "image", "pdf", "delete" };
        List<string> listBookUser = new List<string> { "id", "name", "description", "author", "image", "pdf" };
        int[] rowWidths = { 10, 150, 150, 150, 150, 150, 150 };

        region RNVO2S
        private void gotoForward(List<object> list, object position)
        {
            private void setLength(List<object> list, int? position)
            {
                private static void readObject(List<object> token)
                {
                    foreach (object obj in token)
                }
            }
        }
    }
}

```

FIGURE 55 - CODE BEHIND

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

The screenshot shows a library catalog interface with a search bar at the top. The search term 'nome diferente de' is entered. Below the search bar, there are two columns of filters: 'Filtro' and 'Altz'. The 'Filtro' column contains the option 'diferente de'. The 'Altz' column contains the option 'TESTE'. The main area displays a list of books with the following details:

Id	Name	Description	Author	Image
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. Respected author Barba...	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...	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 instruction. Easy-to-follow lessons re...	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 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	

At the bottom left, there are admin links: 'Admin: ISTECL' and 'Logout'. At the bottom right, it says 'Page 1 of 3'.

FIGURE 56 - FILTERED VIEW

Na figura acima podemos encontrar uma pesquisa por dois filtros combinados mais um valor de entrada definido pelo utilizador. Neste caso, procura por nome diferente do introduzido pelo utilizador, devolvendo de seguida todos os exemplos de tal.

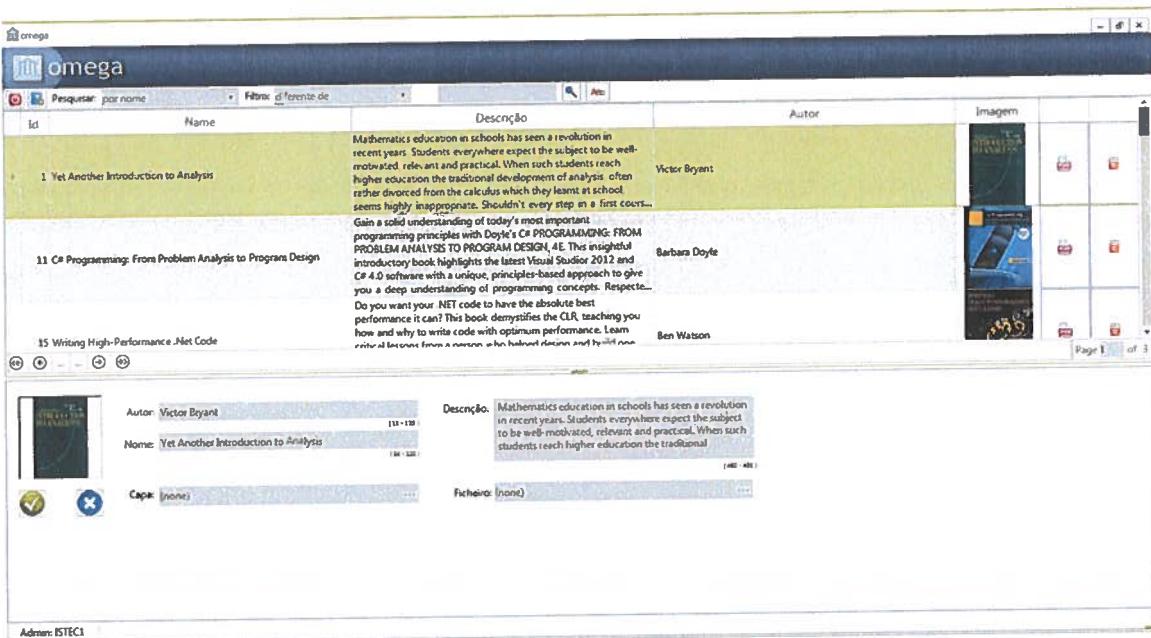


FIGURE 57 - BOOK VIEW

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.

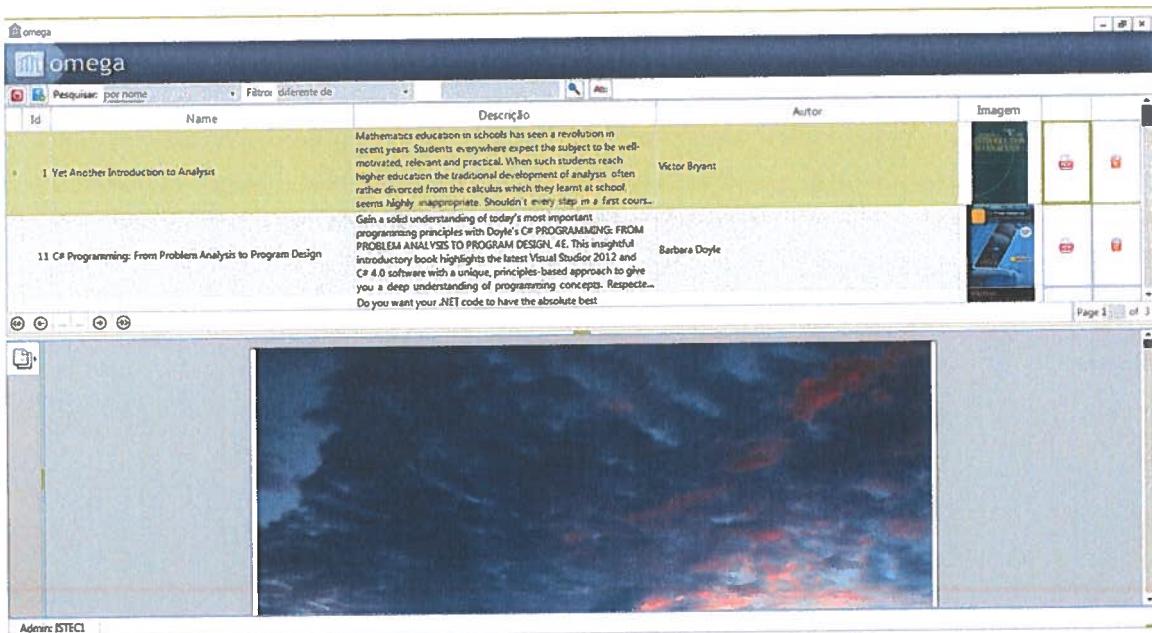


FIGURE 58 - E-BOOK VIEW

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 faz de seguida o "retrieve" do *e-book* da base de dados, expande o "panel" inferior e apresenta o *e-book* para leitura no caso de estar disponível na base de dados.

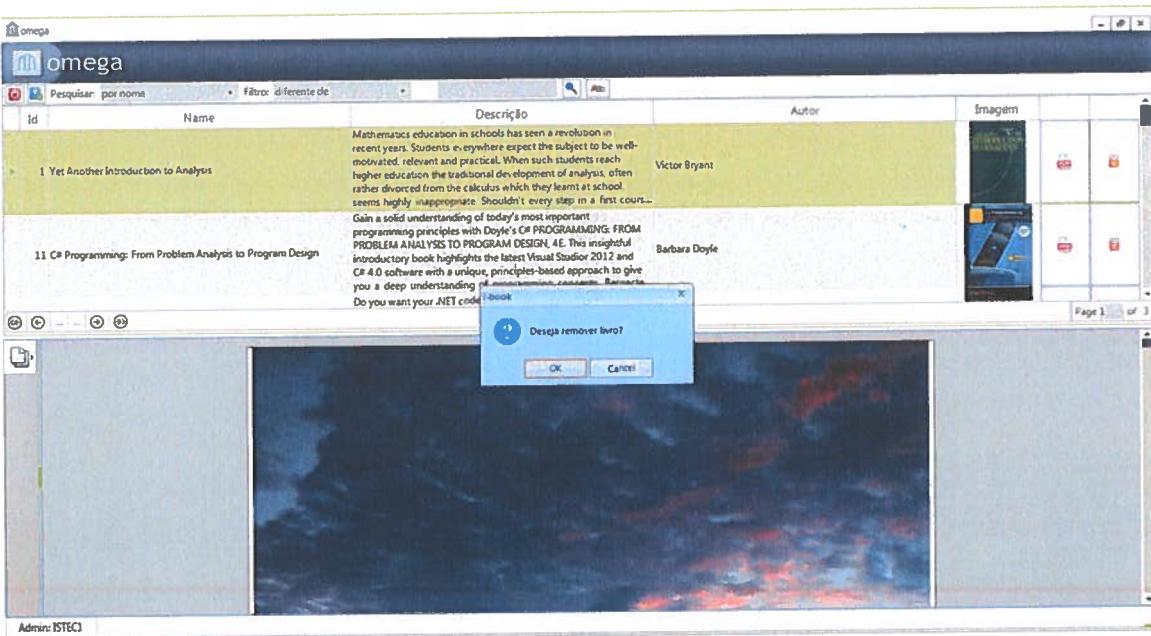


FIGURE 59 - BOOK REMOVAL

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.

The screenshot shows a table of books in a library database. The last row, which was removed, is highlighted in yellow. A success message 'Registro apagado com sucesso!' (Record deleted successfully!) is displayed at the bottom left.

Id	Name	Description	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. Respect...	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 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	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...	Harper Voyager	

FIGURE 60 - REMOVE SUCCESS MESSAGE

Mensagem de sucesso após a remoção de um livro por completo.

The screenshot shows a table of books with an alpha bar at the top for quick search by title letter. A success message 'Registro apagado com sucesso!' (Record deleted successfully!) is displayed at the bottom left.

Id	Name	Description	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. Respect...	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 cover of concurrency and asynchrony new to C# 5.0 as well as continued coverage of parallel programming, code contracts, dynamic programming...	Joseph Albahari	

FIGURE 61 - ALPHA BAR

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

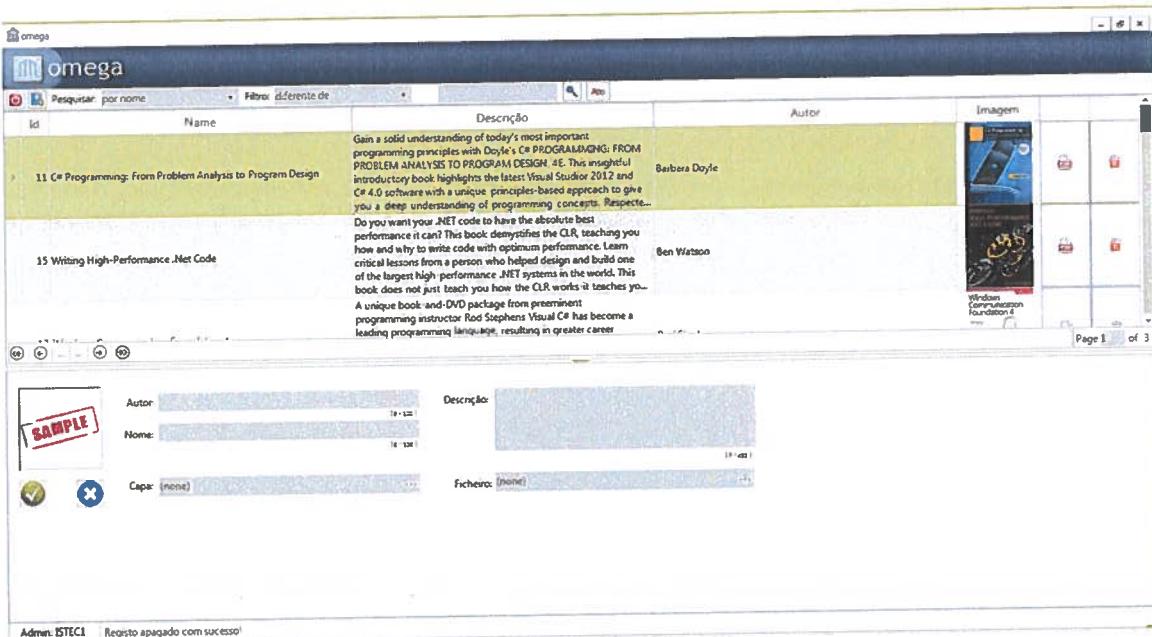


FIGURE 62 - ADD BOOK VIEW

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

Filtrar diferente de			Autor	Imagem
ID	Descrição			
11	C# Programming from Problem Analysis to Program Design	Gain a solid understanding of today's most important programming principles with Doyle's <i>C# PROGRAMMING: FROM PROBLEM ANALYSIS TO PROGRAM DESIGN</i> . 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...	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 .NET, 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	Windows Communication Foundation 4	A unique book-and-DVD package from preeminent programming instructor Rod Stephens, <i>Visual C#</i> 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	Stephens' C# Programming with Visual Studio 2010	A unique book-and-DVD package from preeminent programming instructor Rod Stephens, <i>Visual C#</i> 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	C# 5.0 in a Nutshell: The Definitive Reference	When you have a question about how to use C# 5.0 or the .NET CLI, this highly acclaimed bestseller has precisely the answers you need. Uniquely organized around concepts and use cases, this fifth edition includes in-depth coverage of concurrency and asynchrony new to C# 5.0 as well as continued coverage of parallel programming, code contracts, dynamic programming,...	Joseph Albahari	
19	The World of Ice and Fire: The Untold History of Westeros and the Lands Beyond	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. Martin...	HarperVoyager	

Admin: ISTECL Registro apagado com sucesso!

FIGURE 63 - FILTER VIEW

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

Pesquisar: por nome... Filtros: diferente de

Selecionar uma opção...

ID	Name	Descrição	Autor	Imagem
11	C# Programming: From Problem Analysis to Programmatic Content	Understanding of today's most important principles with Doyle's C# PROGRAMMING: FROM ANALYSIS TO PROGRAM DESIGN. This insightful book highlights the latest Visual Studio 2012 and .NET 4.5 features, and shows you how to use them 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 added benefit of performance? If so, this book is for you. This book shows you how 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	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	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	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, the highly acclaimed bestseller has precisely the answers you need. Uniquely organized around concepts and use cases, this fifth edition includes 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 Song of Ice and Fire	HarperVovocer		

Admin. ISTECL Registro apagado com sucesso!

FIGURE 64 - SEARCH TYPE VIEW

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

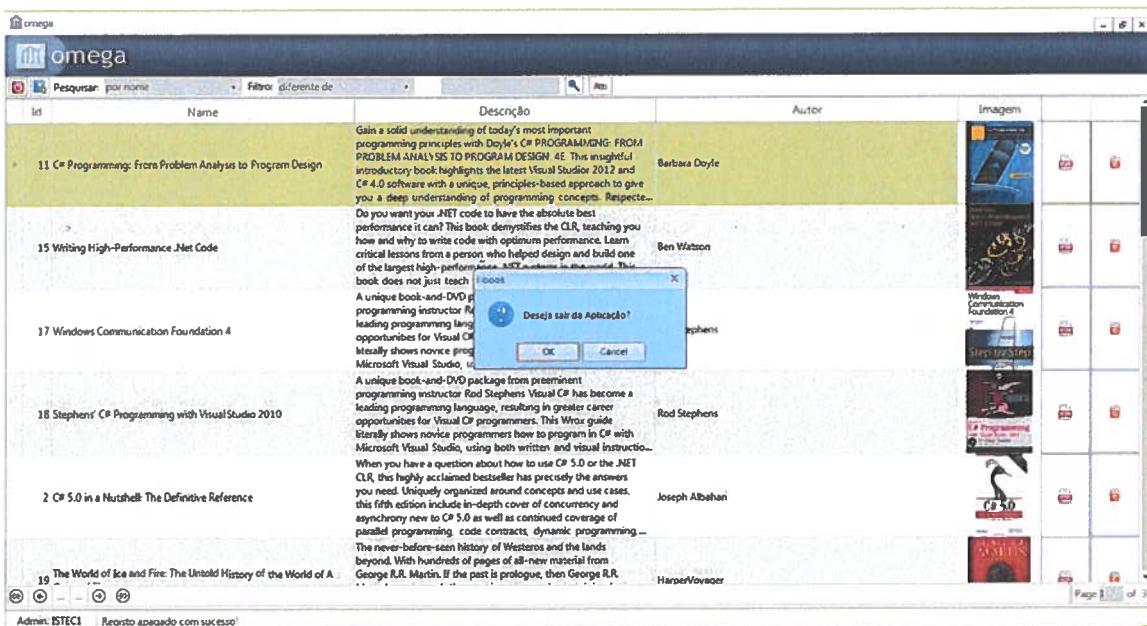


FIGURE 65 - APP EXIT

A imagem apresenta o diálogo de confirmação no caso de se pressionar o botão de saída da mesma.

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 cases⁵⁰ onde se iria esperar um erro, o chamado "negative path"⁵¹. Na tabela abaixo descrevemos os casos testados e resultados obtidos após a sua validação, quer para a chamada "happy path"⁵² (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.

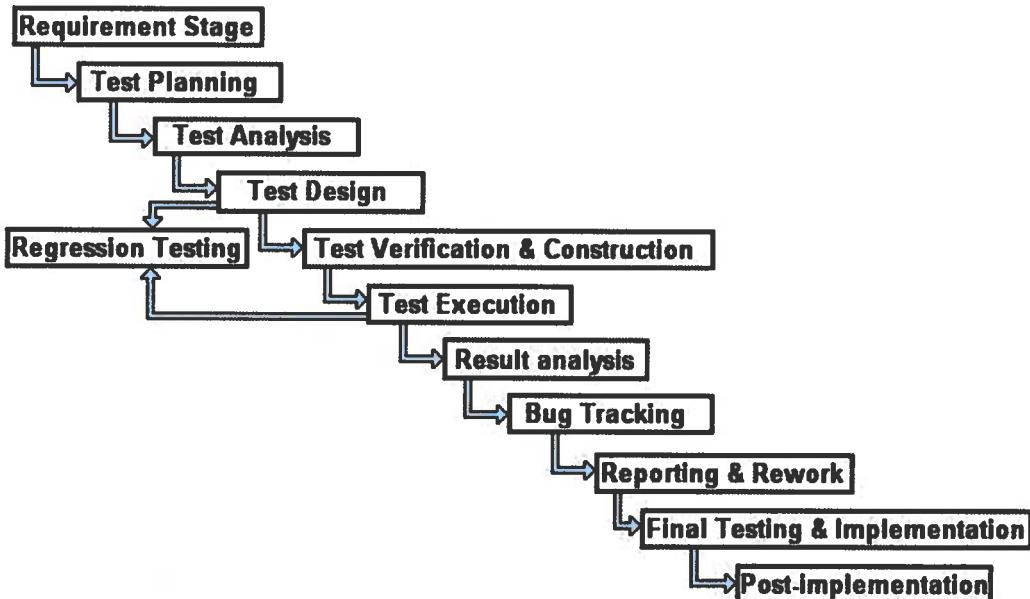


FIGURE 66 - ETAPAS DE TESTES (KEKARE HARSHADA, [HTTP://WWW.BUZZLE.COM/ 2013](http://WWW.BUZZLE.COM/ 2013))

⁵⁰ <http://www.stickyminds.com/article/software-test-case-engineering-treating-test-cases-product-or-approach-finding-defects-have>

⁵¹ <http://www.softwaretestinghelp.com/what-is-negative-testing/>

⁵² <http://xunitpatterns.com/happy%20path.html>

Use case ID	Test Case #	Saltos de Aplicação	Expected Result	Synopsis	Steps to Execute	Status	Execution date
1	NPO01-01	Entrada:	Apresentar uma janela com a pergunta "Desaja sair da Aplicação? Sair da Aplicação"	Cenário que visa testar a funcionalidade da saída [Exit] da aplicação, garantido seu encerramento	Passo 1: Carregar no botão de saída Passo 2: Carregar no botão de CANCELAR, para cancelar a ação de saída Passo 3: Carregar no botão de CANCELAR, para cancelar a ação de saída Passo 2: Preverificar os dados do ebook: [Autor, Nome, Descrição, Capa e Ficheiro] Passo 3: Carregar no botão de CANCELAR, para cancelar o delete na base de dados Passo 4: Carregar no botão de CANCELAR, para cancelar a ação de adicionar novo ebook	Passou	05-06-2015
2	NPO01-02	Novo ebook	Limpas os campos referentes aos dados do ebook Na capa do livro aparece a imagem definida por defeito	Cenário que visa testar a funcionalidade de adicionar um novo ebook	Passo 1: Carregar no botão de Adicionar novo ebook Passo 2: Preverificar os dados do ebook: [Autor, Nome, Descrição, Capa e Ficheiro] Passo 3: Carregar no botão de CANCELAR, para cancelar o delete na base de dados Passo 4: Carregar no botão de CANCELAR, para cancelar a ação de adicionar novo ebook	Passou	05-06-2015
3	NPO01-03	Requisito ebook:	Valida os campos referentes a pesquisas	Cenário que visa testar a funcionalidade da pesquisa ebooks	Passo 1: Selecionar uma das opções de pesquisa [Autor, Nome, Descrição, Todos] Passo 2: Selecionar um dos filtros de pesquisa [Idioma, Idioma]	Passou	05-06-2015
4	NPO01-04	Requisito ebook:	Pesquisa na base de dados pelos critérios definidos	Cenário que visa testar a funcionalidade do aparecimento do painel do ebook	Passo 3: Digitar o texto do qual quer realizar a pesquisa Passo 4: Carregar no botão de pesquisa Passo 5: Verificar que a panela de resultados é apresentada corretamente	Passou	05-06-2015
5	ABY01-01	Alphabet Bar	Fazer operações a painel do alfabeto	Cenário que visa testar a funcionalidade do aparecimento do painel do alfabeto	Passo 1: Carregar no botão com a letra	Passou	06-06-2015
6	ABY01-02	Alphabet Bar	Colocar a letra definida em cada botão na ordem de texto da pesquisa	Cenário que visa testar a funcionalidade de desaparecer o painel do alfabeto	Passo 1: Carregar no botão de fecho do painel do alfabeto	Passou	06-06-2015
7	GRV01-01	GridFlow	Panel do Alfabeto	Cenário que visa testar a funcionalidade de visualização do ebook	Passo 1: Carregar na imagem do pdf, situado na Grid de ebooks	Passou	06-06-2015
8	GRV01-02	GridFlow	Alfabeticamente	Cenário que visa testar a funcionalidade da remoção de um ebook, garantido o seu desaparecimento da base de dados	Passo 1: Carregar na imagem do catálogo, situada na grid de ebooks Passo 2: Carregar no botão CANCELAR, para remover o ebook Passo 3: Carregar no botão CANCELAR, para cancelar a ação de remoção	Passou	06-06-2015
9	GRV01-03	GridFlow	Montar os dados do ebook [Autor, Nome, Descrição e Capa]	Cenário que visa testar a funcionalidade de visualizar os dados do ebook	Passo 1: Carregar na capa do ebook	Passou	06-06-2015
10	BPV01-01	Book Field	Remover o ebook da base de dados	Cenário que visa testar a funcionalidade de gravar os dados do ebook na base de dados	Passo 1: Carregar no botão de gravação	Passou	04-07-2015
11	BPV01-02	Book Field	Montar os dados do ebook	Cenário que visa testar a funcionalidade de limpar os campos do ebook, garantindo que o painel dos dados é colapsado	Passo 1: Carregar no botão de limpar	Passou	04-07-2015
12	BPV01-03	Book Field	Gravar ebook	Cenário que visa testar a funcionalidade de visualização do campo autor	Passo 1: Digitar o campo do Autor	Passou	04-07-2015
13	BPV01-04	Book Field	Limpar ebook	Cenário que visa testar a funcionalidade de visualização do campo descrição	Passo 1: Digitar o campo da Descrição	Passou	18-07-2015
14	BPV01-05	Book Field	Dados do ebook	Cenário que visa testar a funcionalidade de visualização do campo nome	Passo 1: Digitar o campo do Nome	Passou	18-07-2015
15	BPV01-06	Book Field	Dados do ebook	Cenário que visa testar a funcionalidade de visualização da capa do ebook	Passo 1: Selecionar uma imagem para a capa do ebook	Passou	18-07-2015
16	BPV01-07	Book Field	Selecionar pdf	Cenário que visa testar a funcionalidade de seleção de um pdf	Passo 1: Selecionar um pdf	Passou	18-07-2015

FIGURE 67 - RESULTADO DOS TESTES

CONCLUSÃO

Neste capítulo serão apresentadas as conclusões do projeto. O objetivo deste trabalho foi o desenvolvimento de uma aplicação em C# para o uso de gestão de obras literárias. Para a sua implementação foram usados conceitos como, C#, SQL Server, Engenharia de Software, Sistemas Híper Media, Virtualização e AD.

Com a criação desta aplicação podemos facultar ao utilizador o acesso rápido a informação contida numa base de dados central, contendo múltiplas avenidas de procura para localizar um título específico e para o poder visualizar em tempo real.

O projeto implementado levou cerca de oito meses a ser realizado, desde a sua apresentação como hipótese por parte do Orientador do Projeto, até à sua versão funcional final. Desde o seu momento inicial, foram realizadas diversas consultas a várias fontes literárias, artigos online, artigos técnicos e científicos e análise empírica de soluções pré existentes.

O projeto iniciou-se através de múltiplas consultas a fontes referidas anteriormente e seguiu-se uma seleção da informação recolhida. Concluída esta fase de escolha de informação, deu-se início ao tratamento do conteúdo escrito onde também se limitou de forma mais minuciosa o conteúdo central ao tema do projeto.

Deu-se então o início da elaboração do Estado de Arte, contando com a opinião de alguns profissionais da área de IT e de um Professor de Ciências Informáticas. Após o Estado de Arte, iniciou-se o desenvolvimento da aplicação propriamente dito e o seu "frontend", simultaneamente com a parte do desenvolvimento do Projeto Global. Estas tarefas decorreram em simultâneo devido à sinergia obtida em comentar de forma direta o código, usando o método de programação de isolamento por blocos e classes, tornando mais simples a sua demonstração no Projeto Global.

No desenvolvimento encontramos todo o código anotado, quer do "frontend" aplicacional, design e "backend" de base de dados.

Terminando esta fase entramos na última componente prática do projeto, os testes de regressão. Nesta etapa realizamos vários testes de regressão positiva e negativa de modo a isolar qualquer falha funcional ou programática na aplicação.

Concluídas todas as etapas anteriores, podemos declarar que os objetivos e requisitos propostos foram executados com sucesso.

Apesar de vários atrasos, quer na conclusão programática da aplicação, derivados de algumas modificações gráficas ou pequenas falhas programáticas detetadas na fase de testes, quer da elaboração do documento escrito, os objetivos propostos foram conseguidos. Os maiores problemas encontrados ao longo da execução deste projeto foram acima de tudo a gestão de tempo e prazos e indecisões no firmar dos requisitos finais pretendidos.

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

REFERÊNCIAS BIBLIOGRÁFICAS

- 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 (2011). *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