

Integração de Informação

Isabel Juliana Raposo Primo



Integração de Informação

Isabel Juliana Raposo Primo

Dissertação para a obtenção do Grau de **Mestre em Matemática**
Área de Especialização em **Computação**

Júri

Presidente: Alexander Kovacec

Orientador: José Carlos de Gouveia Teixeira

Vogal: Pedro Henrique e Figueiredo Quaresma de Almeida

Data: Fevereiro de 2014

Resumo

Este trabalho de dissertação de mestrado está focalizado na resolução de um problema de crucial importância para a gestão operacional e estratégica moderna: a integração de informação. Com a crescente facilidade no acesso e no armazenamento de informação, um gestor vê-se frequentemente confrontado com grandes volumes de informação, não uniformizada e desagregada.

Depois de uma revisão da literatura relacionada com o desenvolvimento de *software* e o armazenamento e acesso à informação, foi feita a análise de requisitos de um protótipo de um sistema capaz de aceder a diferentes fontes de dados, uniformizar a informação e agregá-la num modelo de dados único e, após a agregação da informação, processá-la.

A análise de requisitos foi utilizada na programação do protótipo, utilizando o paradigma da programação orientada para os objetos. Para mostrar a aplicabilidade do trabalho desenvolvido, criou-se um cenário de utilização que constituiu o caso de estudo.

Palavras Chave: Desenvolvimento de *Software*, Ciências da Computação, Base de dados, Fontes de dados, Processamento da Informação.

Abstract

This master thesis was intended to solve a problem with crucial importance on modern operational and strategic management: the information integration. With the increasing easy access to the information and its storage, the manager is frequently confronted with large amounts of information which is not uniform or aggregated.

After a literature review about software development and information storage and access, it was made the requirements analysis of system prototype able to access several data source and to process the information after their uniformization and aggregation into a single model.

The requirements analysis was applied in the programming of the prototype, using object-oriented programming. Finally, a case study is presented to show the applicability of the work previously made.

Keywords: Software Development, Computer Science, Database, Data Sources, Information Processing.

Agradecimentos

Em primeiro lugar gostaria de agradecer a Deus por tudo que tem feito na minha vida.

Ao meu orientador Prof. Doutor José Carlos Teixeira, por me ter apontado os caminhos acadêmicos necessários para a construção desta produção científica.

À minha família, em especial a minha mãe Maria Adelaide Primo e a minha irmã Dra. Adelaide Primo, pelo apoio incondicional, amor, carinho e dedicação que sempre demonstraram nesta difícil caminhada da minha vida.

E a todos aqueles que contribuíram de forma direta ou indiretamente, em especial ao Dr. Luís Caridade e ao Prof. Doutor Hermenegildo Avelino, expresso aqui o meu reconhecimento e a minha gratidão pela sua colaboração.

Conteúdo

1	Introdução	1
1.1	Enquadramento	1
1.1.1	Objetivos Gerais	1
1.2	Organização da Dissertação	2
2	Revisão Literatura	3
2.1	Introdução às Fontes de Dados	3
2.1.1	MySQL	3
2.1.2	MSAccess	4
2.1.3	Ficheiro CSV	4
2.1.4	Ficheiro XLS/XLSX	4
2.2	Acesso ao Modelo Universal de Dados	4
2.2.1	Características e Recursos das Tecnologias Através do Acesso ao Modelo Universal de Dados	5
2.2.1.1	Open DataBase Connectivity (ODBC)	5
2.2.1.2	Object Linking and Embedding for DataBases (OLE DB)	5
2.2.1.3	Microsoft ActiveX Data Objects (ADO)	6
2.2.2	Relações entre as Diferentes Tecnologias UDA	6
2.2.3	ODBC	7
2.2.3.1	Arquitetura ODBC	7
2.2.4	OLE DB	8
2.2.4.1	Arquitetura OLE DB	9
2.3	Manipulação de Dados	9
2.3.1	Diagrama da Arquitetura ADO.NET	10
2.3.2	System.Data	10
2.3.2.1	DataSet	10
2.3.2.2	DataTable	11
2.3.2.3	DataRow	12
2.3.2.4	DataRelation	12
2.3.3	System.Data.Odbc e System.Data.OleDb	12
2.4	Engenharia de Software	12
2.4.1	Requisitos	13
2.4.2	Requisitos Funcionais	13
2.4.3	Requisitos não Funcionais	13
2.4.3.1	Requisitos de Produto	14
2.4.3.2	Requisitos de Processo	14
2.4.3.3	Requisitos Externos	14
2.4.4	Casos de Uso	15
2.4.4.1	Ator	15
2.4.4.2	Caso de Uso	15

2.4.4.3	Cenários	16
2.4.4.4	Associação entre Ator e Caso de Uso	16
2.4.4.4.1	Inclui (Relacionamento de Inclusão)	16
2.4.4.4.2	Extend (Relacionamento de Extensão)	16
2.4.4.4.3	Generalização	17
2.4.4.4.4	Diagrama de Casos de Uso	17
2.4.5	Storyboard	17
2.4.6	Paradigma de Programação	17
2.4.6.1	Diagrama de Classes	17
2.4.7	Casos de Teste	18
2.4.7.1	Erros no Desenvolvimento do Sistema	18
2.4.7.2	Testes de Sistema	19
2.4.7.2.1	Testes Unitários	19
2.4.7.2.2	Testes de Integração	20
2.4.7.2.3	Testes Funcionais	20
2.4.7.2.4	Testes de Aceitação	20
2.4.7.2.5	Testes de Regressão	20
2.5	Soluções Atuais Existentes no Mercado	20
2.5.1	Navicat Premium (Multiple Databases GUI)	20
2.5.2	DB Visualizer	21
2.6	Justificação para Realização do Trabalho	21
2.6.1	Tecnologias Escolhidas	21
2.6.2	Análise de Soluções Encontradas no Mercado	22
3	Análise de Requisitos	23
3.1	Requisitos Genéricos de Utilização	23
3.2	Diagrama de Casos de Uso	23
3.3	Descrição de Casos de Uso	24
3.4	Requisitos não Funcionais do Sistema	28
3.4.1	Usabilidade	28
3.4.2	Desempenho	28
3.4.3	Interoperabilidade	28
3.4.4	Segurança	28
3.4.5	Implementação	29
3.5	Storyboard do Sistema	29
4	Desenvolvimento do Protótipo	31
4.1	Estruturas de Dados Usadas	31
4.2	Classes	31
4.2.1	Diagrama de Classes do Sistema	32
4.2.2	Classe Login	32
4.2.3	Classe ligarBd	33
4.2.4	Classe LigaFich	33
4.2.5	Classe AdionarFD	33
4.2.6	Classe VisualizarDados	34
4.2.7	Classe Processamento	35
4.2.8	Classe Encerrar	36
4.2.9	Classe Info	36
4.3	Ambiente de Trabalho	36
4.4	Testes	38

5	Caso de Estudo	39
5.1	Ambiente do caso de estudo	39
5.2	Contextualização do Problema	40
5.3	Descrição de Exemplos do Problema	40
6	Discussão, Conclusões e Recomendações	47
6.1	Ultrapassagem de Dificuldades	47
6.2	Cumprimento de Metas e Soluções Propostas	48
6.3	Aprendizagem e Possíveis Melhoramentos no Futuro	48
Anexo A		55
A.1	Storyboard do Sistema	55
Anexo B		61
B.1	Plano de Testes	61
Anexo C		89
C.1	Tabelas Originais e Unificadas	89

Capítulo 1

Introdução

1.1. Enquadramento

Nos dias de hoje, em diversos tipos de instituições e empresas, há uma crescente necessidade de integrar a informação proveniente de diferentes fontes de dados.

Esta diversidade pode tornar-se num grande problema quando existe a necessidade de aceder a todos os dados, assim como a manipulá-los, pois na maioria das vezes são heterogéneos, ou seja, têm origem em processos independentes e utilizam diferentes modelos de dados e diversas representações para os mesmos.

Por isso, quando há necessidade de acesso e manipulação de dados provenientes de diferentes origens, é muito importante a integração desses diferentes tipos e modelos de dados num único formato, facilitando assim a utilização da informação de acordo com as necessidades de cada utilizador.

Um exemplo da situação apresentada pode ser a monitorização da temperatura em várias divisões de diversos edifícios, para assegurar boas condições de habitabilidade. Para isto é importante ter um sistema que seja capaz de integrar os dados recolhidos nos diferentes sistemas instalados nos diversos edifícios, para permitir a utilização de um sistema unificado de gestão da eficiência energética relacionada com as condições de habitabilidade.

Esta dissertação, inserida no mestrado em Matemática em Computação, teve como objetivo a análise deste tipo de situações e a proposta de uma solução para o processo de acesso e de integração de diferentes fontes de informação, como bases de dados e ficheiros, num modelo de dados unificado, para posterior utilização da informação noutros processos e contentores de informação.

1.1.1. Objetivos Gerais

O presente trabalho tem como principais objetivos:

- Analisar os problemas relacionados com a integração de informação proveniente de diferentes fontes de dados;

- Analisar métodos de acesso a fontes de dados;
- Conceber e desenvolver um modelo para o acesso à informação existente em diferentes fontes de dados;
- Desenvolver filtros de pesquisa sobre as tabelas acedidas, de forma a facilitar a visualização dos dados;
- Criar métodos de conversão de dados entre unidades diferentes;
- Unificar os dados provenientes das tabelas abordadas nos pontos anteriores, com intuito de facilitar o seu processamento;
- Guardar a unificação de dados e os resultados do processamento efetuado numa nova estrutura de dados permanente.

1.2. Organização da Dissertação

Este trabalho desenvolve-se ao longo de 6 capítulos, encontrando-se estruturados da seguinte forma:

No segundo capítulo é feita uma revisão da literatura mais diretamente relacionadas com o trabalho, nomeadamente acesso e manipulação de dados e engenharia de *software*. Ainda neste capítulo são referidas algumas soluções atuais existentes no mercado com alguma semelhança com a desenvolvida.

No terceiro capítulo é apresentada a análise de requisitos do protótipo desenvolvido, nomeadamente requisitos funcionais, não funcionais, os casos de uso e o *storyboard*.

O quarto capítulo centra-se no desenvolvimento do protótipo, apresentando o diagrama de classes, uma breve descrição do código e do ambiente gráfico, sendo ainda referenciados os testes.

No quinto capítulo é apresentado um caso de estudo do sistema.

Por fim, no sexto capítulo é feita uma análise geral dos resultados obtidos e perspectivado o futuro deste trabalho.

Capítulo 2

Revisão Literatura

Neste capítulo são abordadas as principais tecnologias relacionadas com o armazenamento, acesso e manipulação de informação que foram consideradas na realização deste trabalho, bem como a disciplina de engenharia de *software*.

É ainda feita uma breve revisão de soluções existentes no mercado de alguma forma relacionadas com o tema central deste trabalho.

2.1. Introdução às Fontes de Dados

Fonte de dados é qualquer conjunto de registos de dados armazenados.

Fontes de dados podem ser, entre outras, bases de dados (MySQL, Oracle, MSAccess, SQL Sever, etc.), ficheiros (CSV, XML, XLS/XLSX, etc.).

Neste estudo serão consideradas as bases de dados, o MySQL e MSAccess e, no caso de ficheiros, o ficheiro CSV e XLS/XLSX.

2.1.1. MySQL

O MySQL é um sistema de gestão de bases de dados ¹(SGBDs) [27] relacionais, *open source* ² [24] que suporta a linguagem SQL ³ (*Structured Query Language*) [7]. É um dos SGBDs de utilização profissional muito utilizado.

O MySQL foi desenvolvido e é disponibilizado pela empresa *MySQL AB Limited Company* [10], que atualmente vende um conjunto de serviços e produtos relacionados com a tecnologia MySQL [23].

¹**Sistema de Gestão de Bases de Dados** é um conjunto de ferramentas que permite gerenciar e manipular toda a informação que se encontra armazenada numa base de dados.

²**Open Source software** é um *software* que pode ser livremente usado, modificado e partilhado (de forma modificada ou não) por qualquer utilizador. O *software* com código aberto é desenvolvido por muitos utilizadores e distribuído com licenças que obedecem às regras do que significa ser um código aberto.

³**SQL** é uma linguagem que é destinada a especificar a informação acerca de cada relação, incluindo: esquema de cada relação; restrições de integridade; domínios de valores associados com cada atributo; informação de segurança e autorização para cada relação; etc..

2.1.2. MSAccess

O MSAccess (também conhecido simplesmente por Access) é um sistema de gestão de bases de dados da *Microsoft*. Combina o motor *Microsoft Jet Database* com uma interface gráfica e as ferramentas de desenvolvimento. O MSAccess tem um formato próprio para armazenar dados baseado no motor *Access Jet Database*. Consegue também importar, exportar ou ligar-se diretamente a informação guardada por outras aplicações ou bases de dados [20].

2.1.3. Ficheiro CSV

CSV (*Comma Separated Values*), é um formato de ficheiro que armazena valores separados por vírgulas e que, em cada linha, tem um registo diferente. É, por isso, adaptado a informação tabelada. Por serem bastante simples, estes ficheiros são comuns nas mais diversas plataformas e sistemas operativos.

O CSV é uma implementação particular de ficheiros de texto separados por um delimitador, que usa a vírgula e a quebra de linha para separar os valores. O formato também usa as aspas em campos no qual são usados os caracteres reservados (vírgula e quebra de linha). Essa robustez no formato torna o CSV mais amplo e ao mesmo tempo mais simples que outros formatos digitais do mesmo segmento [5].

2.1.4. Ficheiro XLS/XLSX

O ficheiro XLSX é uma folha de dados do *Microsoft Office Excel*. Este formato suporta cálculos, gráficos, ferramentas, tabelas dinâmicas e uma linguagem de programação chamada *macro Visual Basic for Applications*. Este ficheiro tem sido muito utilizado para as plataformas *Microsoft Windows e Mac OS X* [26].

2.2. Acesso ao Modelo Universal de Dados

O acesso ao modelo universal de dados (do inglês *Universal Data Access* ou UDA) é uma solução de alto nível da *Microsoft* que permite a utilização de um pacote de tecnologias que para a integração de diversas fontes de dados, relacionais ou não-relacionais [30].

O produto primário de implementação das tecnologias para o UDA é a *Microsoft Data Access Components* (MDAC). O MDAC pode ser implementado em diversas tecnologias, incluindo aplicações *desktop*, arquitetura cliente/servidor e até mesmo em arquiteturas distribuídas.

A tecnologia MDAC inclui ainda os seguintes dados de acesso a interface de aplicações de programação (do inglês *Application programming interface* ou API): *Microsoft ActiveX Data Objects* (ADO); *Object Linking and Embedding for DataBases* (OLE DB) e o *Open DataBase Connectivity*(ODBC).

Cada uma destas tecnologias exibe uma API diferente e um conjunto de diferentes funcionalidades, para diferentes tipos de ambientes e aplicações. Cada uma destas tecnologias UDA pode ser usada em conjuntos com outras, numa abordagem por camadas. Isto significa que a escolha da API pode ser baseada nos recursos da mesma e não sobre se um provedor OLE DB ou *driver* ODBC foi ou não implementado para uma determinada fonte de dados.

2.2.1. Características e Recursos das Tecnologias Através do Acesso ao Modelo Universal de Dados

2.2.1.1. Open DataBase Connectivity (ODBC)

ODBC é uma API amplamente usada para acesso a dados. É baseado nas especificações de *X/Open* da chamada CLI (*Call-Level Interface*) e na ISO/IEC para API de bases de dados. Usa SQL como a sua linguagem primária de acesso a bases de dados.

O ODBC inclui os seguintes recursos e características: utiliza uma API baseada em linguagem C; tem um grande número de controladores disponíveis para uma grande variedade de fontes de dados; é adequado para o acesso a fontes de dados relacionais; suporta tanto SQL, como procedimentos baseados em técnicas de acesso a dados e tem suporte em plataformas não - *Windows* [30].

2.2.1.2. Object Linking and Embedding for DataBases (OLE DB)

OLE DB é um COM ⁴ (*Component Object Model*)[36] API baseado no acesso a dados, desenvolvido após o ODBC. Este componente foi concebido para proporcionar não só o acesso a dados relacionais, mas também aos não relacionais de uma maneira eficiente. Por causa de a sua interface ser baseada em COM, a API tem evoluído e aprimorado para adicionar novos recursos, como *Online Analytical Processing* (OLAP) e apoio a *data mining*.

As suas características e recursos baseiam-se no seguinte: COM-based API; interfaces nativas COM de alta performance adequadas para o acesso com recurso à linguagem C++; suporte para fontes de dados relacionais ou não; suporte OLAP; e

suporte para *data mining*[30].

2.2.1.3. Microsoft ActiveX Data Objects (ADO)

ADO é uma interface de alto nível para exibição de dados do provedor OLE DB aos utilizadores de automação OLE. Foi projetado para simplificar APIs de acesso a dados como a *Data Access Objects* (DAO) e a *Remote Data Objects* (RDO) colocando ao mesmo nível os seus modelos hierárquicos de programação. Esta tecnologia também suporta recursos avançados como: trabalhar com dados *offline* e estar acessível a todas as linguagens de compilação automática, tais como *VBScript*, *JScript* e *Visual Basic*.

As suas características e recursos baseiam-se no seguinte: COM-based; suporte de automatização; uso de dados *offline*; acesso aos dados do lado do cliente por *Remote Data Service* (RDS); persistência de registos de dados e API Simples [30].

2.2.2. Relações entre as Diferentes Tecnologias UDA

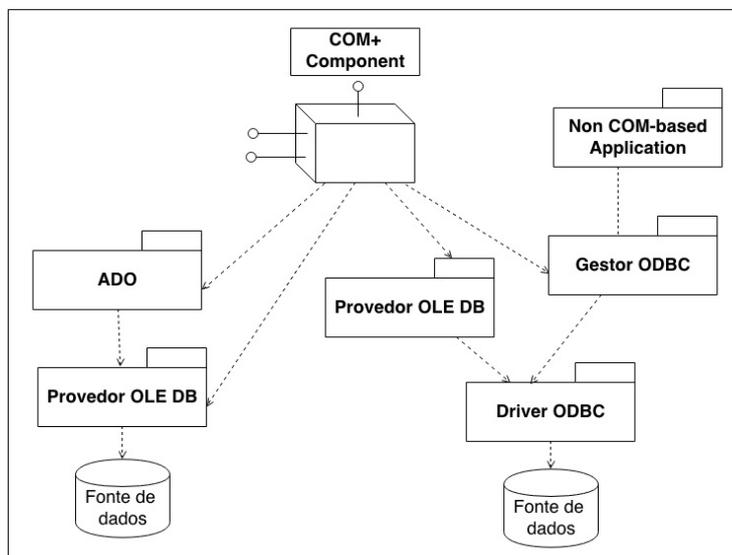


Figura 2.1: Tecnologias UDA [30]

Como se pode verificar na figura 2.1, os provedores OLE DB e os *drivers* ODBC são os componentes do UDA que acedem diretamente as fontes de dados. O gestor

⁴COM é um sistema independente da plataforma de utilização que permite desenvolver objetos (componentes binários) que interagem com outros objetos. Esses objetos podem ser executados no mesmo processo, num outro processo, e, até, numa máquina remota. Os objetos COM podem ser compreendidos, como disponibilizando quer um conjunto de dados, quer um conjunto de funções relacionadas de acesso e manipulação desses mesmos dados.

ODBC expõe a API ODBC para os utilizadores ODBC. Para tornar as fontes de dados disponíveis para os utilizadores OLE DB, a *Microsoft* proporciona uma outra componente MDAC que é conhecida por provedor OLE DB para ODBC. Esta componente converte uma API OLE DB para API ODBC e desempenha um papel de um gestor de ODBC para um driver ODBC.

É de referir que o nível mais alto é o ADO. Este por sua vez, não acede diretamente a qualquer fonte de dados. No entanto, esta tecnologia utiliza o provedor OLE DB para aceder os dados.

Este trabalho, irá contemplar as tecnologias ODBC e OLE DB que serão então descritas com mais pormenor.

2.2.3. ODBC

O ODBC é um padrão para aceder a sistemas de gestão de bases de dados. Este padrão define um conjunto de interfaces que permitem o uso de linguagens de programação como: C#, Java, PHP, C++, etc., com intuito de aceder a diferentes tipos de base de dados sem a necessidade de programar métodos de acesso específicos.

O conector ODBC utiliza uma implementação específica da linguagem SQL com a qual a aplicação comunica com a base de dados de forma visível, permitindo, por exemplo, que um mesmo programa utilize simultaneamente o MSAccess, o MySQL e o SQL Server sem a necessidade de mudanças na sua camada de dados. O uso destas interfaces está condicionado à existência de *drivers* ODBC específicos para base de dados a que se deseja aceder.

O padrão ODBC, foi desenvolvido em 1992 pelo *SQL Access Group* com objetivo de generalizar, por uma aplicação, a utilização de diferentes sistemas de gestão de base de dados [11].

2.2.3.1. Arquitetura ODBC

A arquitetura ODBC é constituída por 4 componentes principais: aplicação, gestor de *drivers*, *drivers* e fonte de dados, conforme se pode ver na figura 2.2 [33].

A aplicação executa todo o processamento. Esta chama funções do ODBC para submeter instruções SQL e obtém os resultados do gestor de *drivers*.

A aplicação consegue aceder a uma fonte de dados, ligando-se através de comandos SQL que permitem a leitura e escrita dos campos da fonte de dados.

O gestor dos *drivers* ODBC processa as chamadas das funções ODBC e passa-as para um *driver* ODBC, resolvendo possíveis conflitos de versões que possam existir.

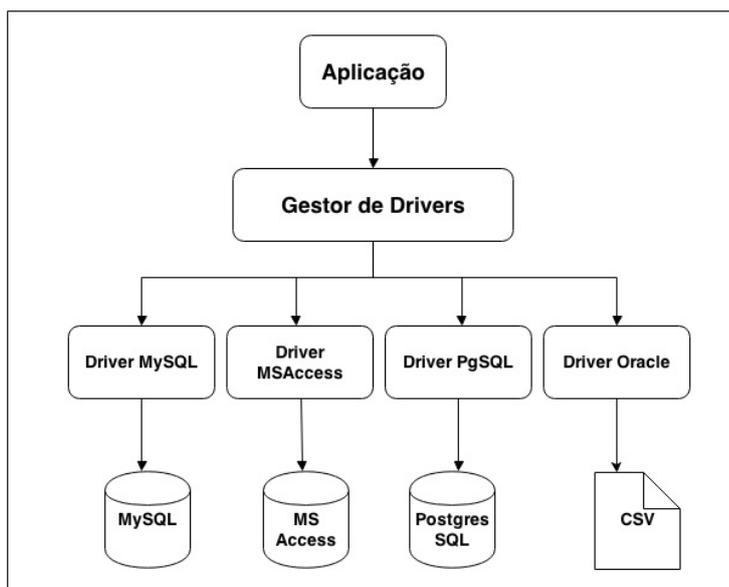


Figura 2.2: Arquitetura do ODBC

Em termos de gestor de *drivers* ODBC, o *Windows* possui uma ferramenta denominada *ODBC Data Source Administrator*, que é responsável por controlar todos os *drivers* e é através dela que se criam novos ficheiros DSN (*Data Source Name*). Os sistemas operativos diferentes do *Windows* podem escolher versões *open source* do gestor de drivers ODBC, como por exemplo *unixODBC* ou *iODBC*, ou então optar por versões comerciais.

O *driver* ODBC processa as chamadas da função ODBC, submete pedidos SQL para uma fonte de dados e retorna os resultados para a aplicação. É ainda capaz de modificar um comando da aplicação para que fique de acordo com a sintaxe suportada pela base de dados associada.

A fonte de dados consiste nos dados a que um utilizador pretende aceder e o seu sistema de gestão da base de dados.

2.2.4. OLE DB

OLE DB é um COM API baseado em acesso a dados relacionais ou não, desenvolvido após ODBC.

Baseado na tecnologia OLE, é uma especificação para um conjunto de interfaces de acesso a dados, projetado para possibilitar que uma variedade de fontes de dados, de todos os tipos e tamanhos, consigam interagir. A sua heterogeneidade é portanto uma das suas características mais importantes.

O único critério imposto por esta tecnologia é que os dados resultantes de um

comando sejam apresentados na forma de uma tabela. Assim qualquer dado pode ser manipulado, desde que possa ser interpretado como uma linha e uma coluna de uma tabela [25].

2.2.4.1. Arquitetura OLE DB

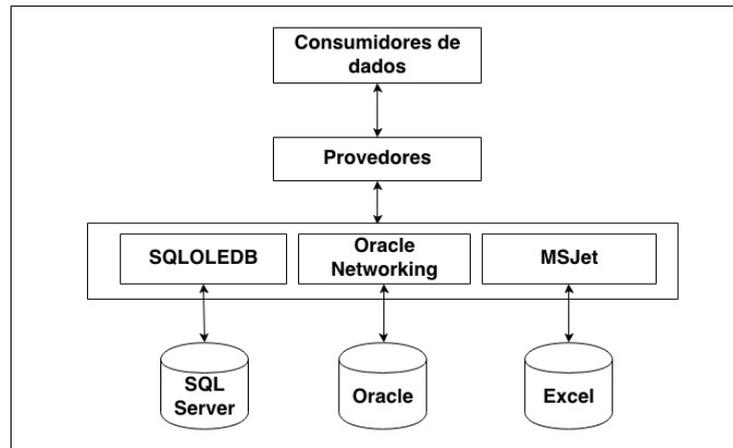


Figura 2.3: Arquitectura do OLE DB

A arquitetura OLE DB foi projetada centrando-se no conceito de consumidor e provedor. Os consumidores de dados são aplicações ou sistemas que utilizam interfaces OLE DB de um provedor para terem acesso a uma grande variedade de dados.

Os provedores de dados respondem aos pedidos dos consumidores através da sua implementação de um conjunto de interfaces COM, disponibilizando a informação em formato de tabela. Estes permitem ao consumidor determinar o formato da tabela que quer usar.

2.3. Manipulação de Dados

Quando se trata de acesso a dados imagina-se logo uma base de dados e na forma de como se deve extrair a informação que é necessária. Esta é na verdade uma visão um tanto simplista pois os dados podem estar armazenados sob diversas formas.

Uma das formas de aceder aos dados é através de uma nova tecnologia ADO.NET, que consiste num conjunto de classes definidas pela *.NET framework* (localizadas no System.Data) que pode ser utilizado para aceder dados armazenados numa fonte de dados.

A principal componente da arquitetura ADO.NET é o DataSet. O DataSet oferece o acesso a tabelas, linhas, colunas, relacionamentos e *constraints*.

O ADO.NET foi projetado, com base no conceito de DataSets que possibilitam a manipulação dos dados de uma forma “*offline*” [21].

2.3.1. Diagrama da Arquitetura ADO.NET

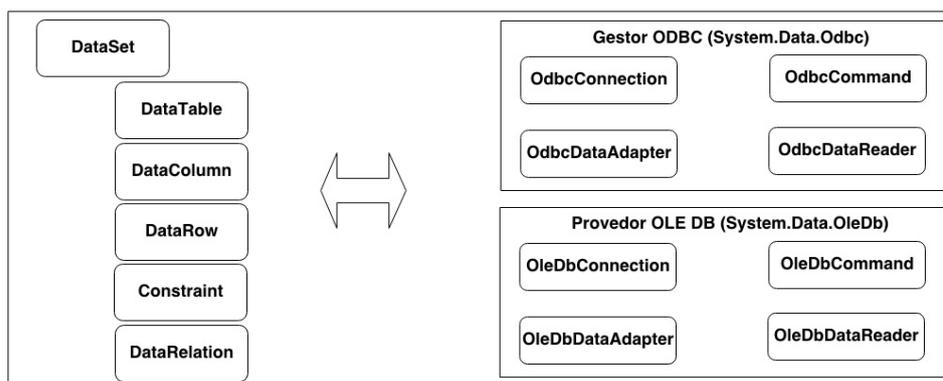


Figura 2.4: Diagrama da Arquitetura ADO.NET [21]

Conforme se pode ver na figura 2.4, a arquitetura ADO.NET é composta por duas componentes: provedores de dados *.Net framework* e o DataSet. A funcionalidade desta arquitetura é feita através do *namespace* System.Data. É de referir que esta arquitetura é implementada em aplicações de clientes, como *Windows Forms* ou páginas HTML criadas pelo *ASP.NET* [21].

2.3.2. System.Data

O *namespace* System.Data fornece o acesso a classes que representam a arquitetura ADO.NET. Essas classes são independentes da fonte de dados. O System.Data contém: DataSet, DataTable, DataRow e DataRelation [21].

2.3.2.1. DataSet

O DataSet é um cache ⁵ [16] de memória dos dados recuperados de uma fonte de dados, sendo um dos principais componentes da arquitetura ADO.NET.

Conforme se pode ver na figura 2.5, a estrutura do DataSet contém tabelas, cada uma um conjunto de linhas e colunas, e os relacionamentos entre elas.

⁵Cache é uma coleção de dados de acesso rápido, interno a um sistema, que serve de intermediário entre um operador de um processo e o dispositivo de armazenamento ao qual esse operador acede.

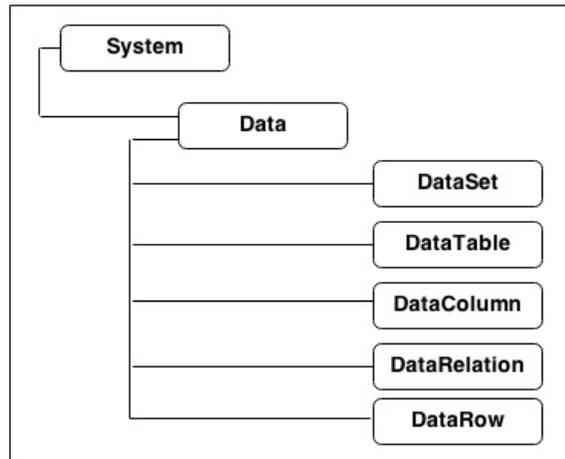


Figura 2.5: Estrutura do DataSet [21]

Os dados internos contidos num DataSet são mantidos no formato XML (*eX-tensible Markup Language*)⁶ [29] e a estrutura do DataSet é definida pelo XSD⁷ [37] (*XML Schema Definition Language*). Assim XML e DataSet estão estritamente ligados. O componente DataSet comporta-se como uma “mini base de dados”, ou seja, é capaz de manipular várias tabelas, relacionamentos, índices e visões dentro de uma mesma estrutura.

2.3.2.2. DataTable

O DataTable é um objeto chave dentro do *namespace* System.Data. Este objeto apresenta as seguintes propriedades:

Columns: Contém uma lista de todas as colunas contidas numa tabela;

Constraints: Regras de dados que são aplicadas a uma tabela;

ChildRelations: Conjunto de relacionamentos que define as relações entre as tabelas filhas e o DataTable;

PrimaryKey: Matriz de objetos DataColumn que representa todas as colunas com as chaves primárias de um DataTable particular;

Rows: Fornecem os dados atuais contidos numa tabela. Contém um conjunto de objetos DataRow;

TableName: Representa o nome do DataTable.

⁶O XML, é uma linguagem de anotação descritiva extensível. Segundo, *World Wide Web Consortium* (W3C), a linguagem disponibiliza um formato para a descrição de dados.

⁷XSD é uma linguagem baseada no formato XML para definição de regras de validação em documentos XML.

2.3.2.3. DataRow

O objeto DataRow é um componente essencial do objeto DataTable. É o *container* para as linhas de dados dentro de uma tabela e permite as funcionalidades: criar, obter, atualizar e excluir.

2.3.2.4. DataRelation

A classe DataRelation contém os dados requeridos para descrever o relacionamento entre os objetos DataTable através dos objetos DataColumn. Ao criar uma DataRelation cria-se um relacionamento Pai/Filho entre duas tabelas, onde as colunas utilizadas para concretizar o relacionamento devem ser do mesmo tipo.

2.3.3. System.Data.Odbc e System.Data.OleDb

Estes namespaces contêm as classes que fornecem a funcionalidade para aceder dados através de um conector ODBC ou através do OLE DB. Algumas das classes mais importantes destes namespaces são descritas abaixo:

OdbcConnection ou OleDbConnection: Representa a ligação a uma fonte de dados.

OdbcCommand ou OleDbCommand: Representa uma instrução SQL ou procedimento armazenado para manipular os dados de uma fonte de dados. Esta classe não pode ser herdada.

OdbcDataReader ou OleDbDataReader: Fornece uma maneira de ler um fluxo de dados reenviado de uma determinada fonte de dados. Esta classe não pode ser herdada.

OdbcDataAdapter ou OleDbDataAdapter: Representa um conjunto de comandos de dados e uma ligação a uma fonte de dados que são utilizados para preencher o DataSet e atualizar a fonte de dados. Esta classe não pode ser herdada.

2.4. Engenharia de Software

Engenharia de *software* é um ramo das ciências da computação direcionada para a especificação, desenvolvimento e manutenção de sistemas.

Na sua base a engenharia de *software* utiliza modelos abstratos que auxiliam na especificação, projeção, implementação e manutenção de sistemas, avaliando e garantindo as suas qualidades.

As principais fases da engenharia de *software* segundo [14] para o ciclo de vida de um sistema consistem na: Iniciação ou Concepção: define o escopo do projeto; Elaboração: define os requisitos e a arquitetura; Construção: desenvolve o sistema; e na Transição: implementação do sistema.

Pela sua importância destacamos a engenharia de requisitos que, segundo [2] corresponde ao processo de aquisição, refinamento e verificação das necessidades do cliente para um sistema, tendo por objectivo uma especificação completa e correta dos requisitos de *software*. Compreende o levantamento, análise, documentação, validação e verificação da qualidade dos requisitos.

2.4.1. Requisitos

Os requisitos são a descrição do que o sistema deverá fazer e das restrições sobre as quais vai operar.

Análise de requisitos trata da especificação dos requisitos necessários para que o sistema funcione como o cliente pretende. Dentro da análise de requisitos encontram-se os casos de uso, que descrevem a interação entre os utilizadores e o sistema.

Por não ser uma ciência exata, acaba por existir uma grande diversidade de problemas na análise de requisitos, como por exemplo: os requisitos não representarem as necessidades reais dos clientes/utilizador; requisitos incompletos; dificuldades de comunicação entre os clientes, analistas e programadores, etc..

Os requisitos estão divididos em requisitos funcionais e não funcionais [2].

2.4.2. Requisitos Funcionais

Um requisito funcional (RF) define uma função de um sistema ou seu componente. Uma função é descrita como um conjunto de entradas, o seu comportamento e as saídas esperadas. Os requisitos funcionais podem ser cálculos, detalhes técnicos, manipulação de dados e de processamento e outras funcionalidades específicas que definem o que um sistema, idealmente, será capaz de realizar [38].

2.4.3. Requisitos não Funcionais

Requisitos não funcionais (RNF) especificam critérios que podem ser usados para avaliar as qualidades globais do funcionamento de um sistema, em vez de comportamentos específicos (que são avaliados pelos requisitos funcionais) [31].

Os RNF podem ser agrupados em três categorias: requisitos de produto, requisitos de processo e requisitos externos [18].

2.4.3.1. Requisitos de Produto

Especificam as características desejadas para o sistema ou subsistemas.

Requisitos de produto pode ser requisitos de:

- **Usabilidade:** Referem-se à forma como é feita a interação entre o utilizador e a interface do sistema.
- **Confiabilidade:** Referem-se à capacidade do sistema manter o seu funcionamento rotineiro ou em situações inesperadas ou hostis.
- **Segurança:** Referem-se a determinados algoritmos a ser usados ou a determinadas medidas para que não haja acessos indevidos ao sistema.
- **Desempenho:** Referem-se às restrições de velocidade do sistema.

2.4.3.2. Requisitos de Processo

Representam restrições colocadas no processo de desenvolvimento do produto. Incluem: *standards* e métodos de desenvolvimento a seguir; ferramentas (CASE) [32] a utilizar; relatórios e documentação e produzir.

Requisitos de processo podem ser requisitos de:

- **Entrega:** Referem-se à forma como o sistema será entregue ao cliente.
- **Implementação:** Referem-se as imposições na área da implementação, como por exemplo em que plataforma vai ser desenvolvido o sistema ou que linguagem vai ser usada.
- **Standards:** Referem-se aos métodos de desenvolvimento a seguir, como por exemplo guias de estilo ou determinada nomenclatura para métodos e funções a ser usado no sistema.

2.4.3.3. Requisitos Externos

Podem afetar os requisitos de produto ou de processo. Derivam do ambiente no qual o sistema é desenvolvido.

Requisitos externos podem ser requisitos de:

- **Restrições legais:** Referem-se às questões legais de leis e padrões aos quais o sistema deve obedecer.
- **Restrições económicas:** Referem-se ao orçamento existente para o projecto.

- **Interoperabilidade:** Referem-se à capacidade do sistema de lidar com interações entre diferentes tecnologias.

2.4.4. Casos de Uso

Os casos de uso são usados como parte integrante da análise de requisitos para especificar e gerir os mesmos. Assim os casos de uso trazem como benefício o apoio na obtenção dos requisitos do sistema, o permitir do aumento do grau de rastreabilidade na fase seguinte do processo de desenvolvimento do sistema e o fornecer de formas de validar os requisitos.

A representação usada para os casos de uso é gráfica e recorre a uma linguagem *standard* de modelação UML (*Unified Modeling Language*).

Os casos de usos são constituídos por diversos elementos descritos abaixo [6].

2.4.4.1. Ator

Um ator é um papel que um utilizador desempenha relativamente à interação com o sistema. É o ator que executa o caso de uso. Um utilizador pode ter vários papéis dentro do mesmo sistema.

Em UML um ator é representado pelo símbolo.

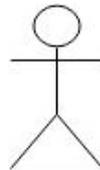


Figura 2.6: Representação do ator em UML

2.4.4.2. Caso de Uso

Um caso de uso representa uma unidade discreta da interação entre um utilizador (humano ou máquina) e o sistema. Para cada caso de uso existe uma descrição associada que descreve a funcionalidade que irá ser construída no sistema proposto. Um caso de uso pode “incluir” outra funcionalidade de caso de uso ou “estender” outro caso de uso com seu próprio comportamento. Casos de uso são tipicamente relacionados a “atores”.

Em UML a notação utilizada para representar um caso de uso é a seguinte.



Figura 2.7: Representação do caso de uso em UML

2.4.4.3. Cenários

Um cenário é uma sequência de passos ou ações, dos muitos possíveis, na execução de um caso de uso, com a finalidade de atingir um determinado objetivo.

Geralmente existe mais que uma forma de execução de um caso de uso, sendo conveniente registrar todos os possíveis cenários de execução.

2.4.4.4. Associação entre Ator e Caso de Uso

Quando um determinado ator interage com uma determinada funcionalidade do sistema, diz-se que existe uma associação entre o ator e o caso de uso. As associações são descritas abaixo [32].

2.4.4.4.1. Include (Relacionamento de Inclusão)

Recorre-se a um relacionamento de inclusão quando se tem um bloco de comportamento que é o mesmo para vários casos de uso. Nesse caso faz sentido isolá-lo e fazer com que esses casos de uso o “invocuem”. Quando se instancia o caso de uso principal, este inclui obrigatoriamente o comportamento do caso de uso incluído.

Em UML a notação utilizada para representar um *include* é o seguinte:



Figura 2.8: Representação de uma relação *include* em UML

2.4.4.4.2. Extend (Relacionamento de Extensão)

Recorre-se a um relacionamento de extensão quando se tem um caso de uso que é praticamente igual a outro, mas que faz um pouco mais ou é um pouco diferente. Nesse caso, isola-se esse caso de uso e diz-se que ele é uma “extensão” do primeiro.

Em UML a notação utilizada para representar um *extend* é o seguinte:



Figura 2.9: Representação de uma relação *extend* em UML

2.4.4.4.3. Generalização

Definição de casos à custa de outros. O Caso de uso “herda” propriedades e semântica do caso de uso base. Pode substituir especificações ou acrescentar outras.

Em UML a notação utilizada para representar a generalização é a seguinte:



Figura 2.10: Representação de uma relação de generalização em UML

2.4.4.4.4. Diagrama de Casos de Uso

O Diagrama de casos de uso descreve de forma visual a funcionalidade proposta para o sistema.

No fundo um diagrama de casos de uso é o agregar de todos os casos de uso que descrevem as várias funcionalidades que o sistema deve ter. Em UML a notação utilizada para um diagrama de casos de uso é a junção das notações dos vários símbolos acima descritos.

2.4.5. Storyboard

Storyboard é uma ilustração de imagens apresentadas sequencialmente com o intuito de pré-visualizar as funcionalidades do sistema. É usado pelos programadores como apoio à recolha de requisitos, tanto no âmbito da validação de funcionalidades, como no âmbito da validação do protótipo visual [15].

2.4.6. Paradigma de Programação

O paradigma de programação utilizado neste trabalho foi o de programação orientada a objeto (POO). Esta é uma metodologia de programação na qual os programas estão organizados como uma coleção de objetos que cooperam entre si. A representação do sistema de classes faz-se através de um diagrama de classes [28].

2.4.6.1. Diagrama de Classes

O diagrama de classes representa a estrutura do sistema, com base no conceito de classes e relações entre elas. Para o modelo de classes são identificados os objetos relevantes do sistema em estudo. Um objeto é uma ocorrência que tem interesse para o sistema e do qual se descreve identidade e comportamento. O comportamento de um objeto define como este age e reage a estímulos externos, enquanto a identidade

é o que o distingue dos demais objetos. Na prática, um objeto é uma instância de uma classe.

Os objetos de interesse para este diagrama são as classes e as relações. Uma classe é um conjunto de objetos que partilham os mesmos atributos e comportamentos. Uma relação é o que representa a ligação entre as classes.

A simbologia utilizada em UML para representar estes dois conceitos é:

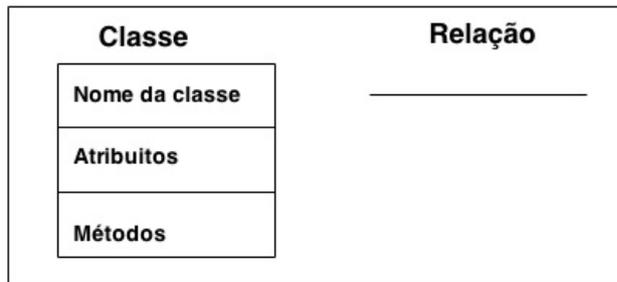


Figura 2.11: Classe e relação em UML

A relação entre classes é representada por uma linha recta e significa que as classes interagem entre si. A classe é composta pelo seu nome e pela identificação de todos seus atributos e métodos [3].

2.4.7. Casos de Teste

Casos de teste é um conjunto de regras utilizados para testar um determinado sistema. Este pode ainda ser elaborado para identificar os erros de estrutura interna do sistema, de forma a garantir a qualidade do mesmo. Também garante que os requisitos do sistema desenvolvidos estão de acordo com o pretendido.

O caso de teste deve especificar a saída esperada assim como os resultados previsto do processamento [13].

2.4.7.1. Erros no Desenvolvimento do Sistema

No processo de desenvolvimento do sistema, nem todos os erros são causados por falhas na escrita do código. Uma fonte bastante comum de erros graves é a falha na análise de requisitos [17][13]. E apesar do uso dos melhores métodos de desenvolvimento, ferramentas ou profissionais, os erros permanecem presentes, o que torna a atividade de testes fundamental durante o desenvolvimento de um sistema.

O tamanho de um projeto a ser desenvolvido e a quantidade de pessoas nele envolvidas são dois possíveis fatores que aumentam a complexidade dessa tarefa e consequentemente aumentam a probabilidade da existência de erros.

Os erros normalmente são introduzidos na transformação de informações entre as diferentes fases do ciclo de desenvolvimento de um sistema.

Exemplo:

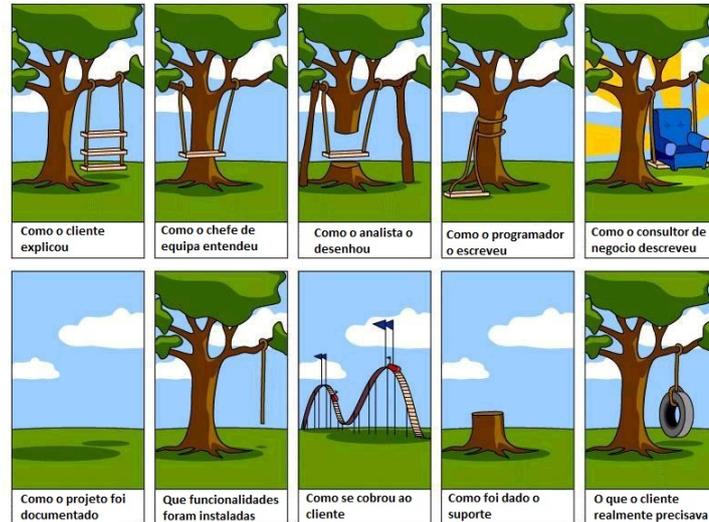


Figura 2.12: Diferentes interpretações ao longo do ciclo de desenvolvimento de um sistema

2.4.7.2. Testes de Sistema

Os testes de sistema (*software testing*) são a investigação do sistema a fim de fornecer informações sobre a sua qualidade em relação ao contexto em que ele deve operar [17][13].

Testes de sistema incluem também o processo de executar o programa ou aplicação a fim de encontrar erros no sistema (*software bugs*). Testes de sistema podem ainda ser definidos como o processo de validação e verificação de um programa/aplicação/produto tal que este: satisfaça os requisitos definidos para o *design* e o desenvolvimento; funcione como esperado; possa ser implementado com as mesmas características; e satisfaça as necessidades do cliente.

Os principais tipos de teste de sistema serão descritos abaixo [34].

2.4.7.2.1. Testes Unitários

Os testes unitários têm por objetivo analisar a menor unidade do projeto, de forma a encontrar falhas originadas por erros de lógica e de implementação em cada módulo, separadamente.

2.4.7.2.2. Testes de Integração

Estes testes têm por objetivo encontrar falhas associadas às interfaces entre os módulos, quando eles são integrados para construir a estrutura do sistema que foi estabelecida na fase de projeto.

2.4.7.2.3. Testes Funcionais

Avalia o sistema procurando falhas por meio da utilização do mesmo, como se fosse um utilizador final. Desta maneira, os testes são executados nos mesmos ambientes, com as mesmas condições e com os mesmos dados de entrada que um utilizador usaria no dia-a-dia. Em suma, verifica se o produto satisfaz os requisitos definidos na análise de requisitos.

2.4.7.2.4. Testes de Aceitação

Geralmente são realizados por um grupo restrito de utilizadores finais do sistema. Estes simulam as operações de rotina do sistema de modo a verificar se o seu comportamento está de acordo com o solicitado.

2.4.7.2.5. Testes de Regressão

Testes de regressão não correspondem a um nível de teste, mas são uma estratégia importante para redução de “efeitos colaterais”. Consiste em aplicar, a cada nova versão do sistema ou a cada ciclo, todos os testes que já foram aplicados nas versões ou ciclos de testes anteriores do sistema. Pode ser aplicado em qualquer nível de teste.

2.5. Soluções Atuais Existentes no Mercado

Com o intuito de perceber que soluções existem no mercado de alguma forma relacionados com a solução final pretendida, foi feita uma breve análise de duas soluções de *software* existente nesta área.

Para esta análise deu-se particular ênfase a características como a possibilidade de ligação a várias bases de dados e ficheiros, a manipulação de dados importados e exportação de dados finais novamente para base de dados ou ficheiros.

2.5.1. Navicat Premium (Multiple Databases GUI)

Esta ferramenta permite administrar, dentro de mesmo sistema e em simultâneo, múltiplas conexões a várias bases de dados. Dentro das bases de dados suporta-

das destacam-se MySQL, SQL Server, SQLite, Oracle e PostgreSQL, como sendo suportadas a maior parte das funcionalidades a elas associadas. É possível ainda transferir dados entre diferentes bases de dados ou entre ficheiros de texto, com formato SQL. Outras das suas características são: importação de dados de ficheiros Excel, MSAccess, CSV, XML, TXT, etc., construção de pesquisas, construção de relatórios, sincronização de dados, construção de cópias de segurança, exportação de dados para ficheiros, etc. [22].

2.5.2. DB Visualizer

O sistema DB Visualizer permite, no mesmo ambiente, uma junção de diferentes bases de dados e da execução de várias tarefas de administração sobre as mesmas. Permite a importação de dados de ficheiros CSV para tabelas já existentes no sistema ou para novas tabelas. Também exporta os dados para ficheiros CSV, Excel, XML, HTML e outros tipos de texto e para bases de dados. Permite ainda a manipulação de dados através de consultas aos dados importados e a criação, a partir de consultas, de gráficos exportáveis [9].

Foram ainda encontrados outras soluções como o Abacuc e o DbDeveloper que permitem a ligação a diferentes fontes de dados, a manipulação dos dados importados e a exportação destes para ficheiros [8] [1].

2.6. Justificação para Realização do Trabalho

2.6.1. Tecnologias Escolhidas

Findos os estudos preliminares em diversas tecnologias que permitem o acesso a diversos tipos de fontes de dados, entre elas, o ADO, o OLE DB e o ODBC foram tomadas algumas opções para iniciar o desenvolvimento do trabalho.

Optou-se pela utilização das tecnologias ODBC e OLE BD. O ODCB por ser a mais transversal a todas as aplicações e por ser a que está mais estudada e mais documentada ao nível de informação. O OLE BD por ser uma tecnologia mais recente e de indicada ligação a dados não relacionais e como tal mais voltada para a ligação aos ficheiros XLS/XLSX.

Em relação à necessidade de armazenar dados, após considerar algumas opções, optou-se por escolher uma estrutura unificante, neste caso o DataSet. O desenvolvimento deste trabalho poderia suportar apenas as tecnologias ODBC e OLE DB, mas não teria a possibilidade de trabalhar com os dados em *offline*. Não utilizando uma

estrutura unificante, por cada consulta/alteração a uma tabela, ir-se-ia efetuar um novo pedido ao servidor onde se localizam as tabelas. Desta forma, uniformizando os dados para uma estrutura que funciona em modo “*offline*”, está-se a diminuir a quantidade de pedidos ao servidor, tornando a aplicação mais rápida e eficiente.

Foram também escolhidas as fontes de dados que serão utilizadas no desenvolvimento do trabalho. No caso as bases de dados serão o MySQL e o MSAccess e os formatos de ficheiro o CSV e o XLS/XLSX.

2.6.2. Análise de Soluções Encontradas no Mercado

Após a revisão da literatura efetuada, concluiu-se também que existem soluções no mercado que apresentam alguma semelhança com a solução pretendida, mas que são limitadas, não havendo uma grande diversidade funcional entre elas.

O ponto comum entre as soluções analisadas é a capacidade de juntarem dentro da mesma solução, múltiplas conexões a bases de dados ou ficheiros. Também todas fazem manipulação de dados através de pesquisas aos dados importados. Por fim, algumas fazem também exportação dos dados finais para base de dados e outros para ficheiros.

No entanto nenhuma delas responde ao que foi previsto para a realização deste trabalho. Ou seja, uma grande parte do que se pretende com este trabalho é a unificação de dados, isto é, a seleção e junção de alguns dados de tabelas de diferentes fontes de dados, para apenas uma tabela unificada sobre a qual poderão ser aplicados processamentos estatísticos.

Capítulo 3

Análise de Requisitos

3.1. Requisitos Genéricos de Utilização

No âmbito deste trabalho, podem considerar-se os seguintes requisitos genéricos: o acesso a informação existente em diferentes fontes e contentores de informação, com tipos e estruturas específicos; integração dessa informação num modelo de dados único e o acesso uniforme à informação integrada, para sua utilização no âmbito de outros processos.

Em primeiro lugar, o utilizador deverá ser capaz de se autenticar no sistema, através de um nome de utilizador (*username*) e uma palavra-passe (*password*). Uma vez autenticado, poderá ter acesso às fontes de informação disponíveis, para consulta ou para iniciar um processo de integração de informação. Em relação à informação integrada, é possível proceder ao seu processamento, quer através da aplicação de filtros quer através do cálculo, como por exemplo o cálculo estatístico.

Depois do processamento dos dados obtidos das fontes de dados, tanto os dados unificados, como os novos dados resultantes dos cálculos efetuados, estes podem ser guardados ou numa nova base de dados ou num ficheiro, para posterior utilização.

3.2. Diagrama de Casos de Uso

O Diagrama de Casos de Uso descreve a funcionalidade proposta para o sistema.

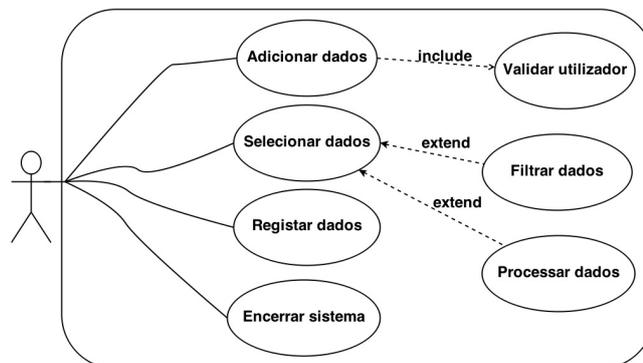


Figura 3.1: Diagrama de Casos de Uso

Conforme se pode ver na figura 3.1 o diagrama de caso de uso está constituído pelos seguintes elementos:

Ator Principal	Casos de Uso
1. Utilizador	1. Validar utilizador 2. Adicionar dados 3. Selecionar dados 4. Registrar dados 5. Encerrar sistema

Tabela 3.1: Casos de uso e atores do sistema

3.3. Descrição de Casos de Uso

Para cada caso de uso é redigido um documento com a descrição dos cenários. Assim, para cada caso de uso existe o seguinte *template*:

ID: Representa o número do caso de uso a ser descrito.

Nome: Representa o nome dado ao caso de uso em questão.

Criação e Data de criação: Representa o nome de quem criou o caso de uso e a data em que o fez.

Atores: Define qual o ator que interage com o sistema neste caso de uso.

Descrição: Descreve a interação entre o ator e o sistema.

Pré-condições: Representa as regras de estado que definem como o sistema se deve encontrar antes de reproduzir o caso de uso.

Pós-condições: Representa as regras de estado que definem como o sistema se deve encontrar após reproduzir o caso de uso.

Cenário Principal: Descreve todos os passos necessários para que o caso de uso seja reproduzido no seu cenário mais provável.

Cenário Alternativo: Descreve todos os passos necessários para que o caso de uso seja reproduzido num cenário que não sendo o mais predominante pode também acontecer.

Exceções: descreve os erros que o sistema pode gerar durante a reprodução do caso de uso.

Inclusões: Indica se este caso de uso depende de outro caso de uso e qual.

3.3 Descrição de Casos de Uso

Caso de Uso ID:	1		
Nome:	Validar utilizador		
Criado:	Isabel Primo	Atualizado por:	Isabel Primo
Data de Criação:	14/03/2013	Última atualização:	07/10/2013
Atores:	Utilizador		
Descrição:	<ol style="list-style-type: none"> 1. O utilizador acede ao ecrã de <i>login</i> 2. Identifica-se com o seu <i>username</i> e <i>password</i> com sucesso 		
Pré-condições:	1. O utilizador tem o acesso ao sistema		
Pós-condições:	1. O utilizador consegue aceder ao sistema		
Cenário Principal:	<ol style="list-style-type: none"> 1. O utilizador inicia o sistema 2. O utilizador preenche os dados 3. O sistema valida os dados 4. O utilizador acede ao sistema 		
Cenário Alternativo:	<ol style="list-style-type: none"> 1. O sistema informa que o utilizador não tem permissões para aceder ao sistema 2. O sistema limpa as caixas de textos e permite que o utilizador preencha novamente os dados 		
Exceções:	<ol style="list-style-type: none"> 1. O sistema informa que ocorreu um erro 2. O sistema termina 		
Inclusões:	-		

Caso de Uso ID:	2		
Nome:	Adicionar dados		
Criado:	Isabel Primo	Atualizado por:	Isabel Primo
Data de Criação:	14/03/2013	Última atualização:	03/10/2013
Atores:	Utilizador		
Descrição:	1. O utilizador acede ao ecrã para escolher a fonte dados à qual se pretende ligar		
Pré-condições:	1. O utilizador acede ao ecrã adicionar dados ao sistema		
Pós-condições:	1. O utilizador consegue seleccionar os dados das fontes de dados às quais acedeu		
Cenário Principal:	1. O sistema apresenta opções para aceder ao tipo de fonte de dados pretendida		

Capítulo 3 Análise de Requisitos

	<ol style="list-style-type: none"> 2. O utilizador escolhe o tipo de fonte de dados à qual deseja aceder 3. O utilizador preenche os campos de texto necessários para a ligação à fonte de dados 4. O utilizador acede dados
Cenário Alternativo:	–
Exceções:	1. O sistema informa que os dados indicados para a ligação à fonte de dados não são válidos
Inclusões:	1. Validar utilizador

Caso de Uso ID:	3		
Nome:	Selecionar dados		
Criado:	Isabel Primo	Atualizado por:	Isabel Primo
Data de Criação:	14/03/2013	Última atualização:	03/10/2013
Atores:	Utilizador		
Descrição:	1. O utilizador poderá selecionar, aplicar filtros ou funções de processamento sobre os dados		
Pré-condições:	1. O utilizador ter acedido a uma fonte de dados		
Pós-condições:	1. O utilizador adiciona os dados selecionados à tabela unificada		
Cenário Principal:	<ol style="list-style-type: none"> 1. O utilizador seleciona dados da lista de dados apresentados 2. O sistema mostra apenas os dados selecionados na tabela unificada 		
Cenário Alternativo:	<ol style="list-style-type: none"> 1. Filtrar dados <ol style="list-style-type: none"> 1.1. O utilizador pode aplicar filtros aos dados apresentados 1.2. O sistema mostra apenas os dados filtrados 1.3. O utilizador pode remover os filtros anteriormente aplicados, caso deseje 2. Processar dados <ol style="list-style-type: none"> 2.1. O utilizador pode selecionar funções de processamento para aplicar aos dados apresentados 2.2. O sistema mostra o resultado das funções aplicadas 		
Exceções:	–		
Inclusões:	–		

3.4 Requisitos não Funcionais do Sistema

Caso de Uso ID:	4		
Nome:	Registar dados		
Criado:	Isabel Primo	Atualizado por:	Isabel Primo
Data de Criação:	14/03/2013	Última atualização:	18/04/2013
Atores:	Utilizador		
Descrição:	1. O utilizador dá instruções ao sistema para armazenar os dados unificados numa estrutura permanente		
Pré-condições:	1. Ter selecionado dados		
Pós-condições:	1. O sistema informa se os dados foram ou não guardados com sucesso		
Cenário Principal:	1. O utilizador escolhe guardar os dados		
Cenário Alternativo:	-		
Exceções:	1. O local que o utilizador escolheu guardar os dados não é válido ou está inacessível		
Inclusões:	-		

Caso de Uso ID:	5		
Nome:	Encerrar sistema		
Criado:	Isabel Primo	Atualizado por:	Isabel Primo
Data de Criação:	14/03/2013	Última atualização:	07/10/2013
Atores:	Utilizador		
Descrição:	1. O utilizador acede ao ecrã onde se encontra a opção para encerrar o sistema		
Pré-condições:	1. Estar no ecrã onde se encontra a opção encerrar o sistema		
Pós-condições:	1. O sistema termina		
Cenário Principal:	1. O utilizador escolhe a opção encerrar o sistema		
Cenário Alternativo:	1. O utilizador escolhe não encerrar o sistema 2. O sistema não termina		
Exceções:	-		
Inclusões:	-		

3.4. Requisitos não Funcionais do Sistema

No âmbito deste trabalho vão apenas ser especificados requisitos não funcionais de: usabilidade, desempenho, interoperabilidade, segurança e implementação.

3.4.1. Usabilidade

Em termos de usabilidade, todas as funcionalidades serão acedidas maioritariamente pelo *click* do rato. Os ecrãs são pensados para serem simples, acessíveis e auto-explicativos da forma como introduzir dados e obter resultados.

3.4.2. Desempenho

Em relação aos tempos de resposta do *software*, espera-se que a resposta em relação ao cálculo seja imediata, isto é, ao clicar no botão para efetuar qualquer cálculo o resultado seja imediatamente mostrado.

Em relação ao acesso às fontes de dados, nomeadamente às bases de dados, espera-se que haja um curto compasso de espera entre o clicar na opção para aceder e a devolução dos dados. Este tempo de espera será tanto maior, quanto maior for o tamanho da base de dados em questão.

Quanto ao armazenamento dos dados, espera-se que guardar para base de dados seja um processo imediato. Já o guardar para ficheiros espera-se que demore um pouco mais.

3.4.3. Interoperabilidade

Este sistema vai ser capaz de lidar com ligações a diferentes fontes de dados. Para essa ligação serão utilizados os conectores ODBC e OLE DB.

Para teste deste desenvolvimento vão ser usadas as bases de dados MySQL e MSAccess e os ficheiros no formato CSV e XLS/XLSX.

3.4.4. Segurança

Na parte da segurança o sistema deve garantir que os dados estão protegidos de acessos não autorizados. Isto é garantido pelo mecanismo de *login* em que o sistema compara os dados introduzidos pelo utilizador com os que estão guardados no ficheiro de registo e só permite que o utilizador aceda ao sistema se os dados coincidirem.

Em relação ao acesso às fontes de dados, em específico as bases de dados necessitam da devida autenticação, ou seja, da informação do IP (*Internet Protocol*) do

servidor e das credenciais de *username* e *password*. Estas informações serão fornecidas pelos proprietários das mesmas e colocadas durante a execução do programa.

3.4.5. Implementação

Nos requisitos de implementação, este sistema tem apenas a restrição de ter de ser implementado com recurso à linguagem C#, por opção de ir ao encontro das tecnologias mais atuais do mercado [15].

3.5. Storyboard do Sistema

O *Storyboard* apresentado para este sistema utiliza *mock-ups*, pelo que não pretende representar o *design* final do mesmo, mas sim, ser representativo das funcionalidades apresentadas por este. O desenvolvimento do *storyboard* do sistema encontra-se descrito no anexo A.

Capítulo 4

Desenvolvimento do Protótipo

Para a implementação do protótipo destinado e ser prova de conceito deste trabalho foi escolhida a linguagem de programação C# [19], por ser a linguagem mais adequada para o que se pretendia realizar, ser de fácil utilização e atualmente em ampla expansão no mercado de trabalho [4]. Foi ainda implementada uma interface gráfica simples mas que permita assegurar um uso intuitivo e fácil do protótipo.

Nos pontos abaixo serão referidos alguns aspetos relevantes da implementação.

4.1. Estruturas de Dados Usadas

Em relação às estruturas de dados a serem usadas foi estudada a tecnologia ADO.NET, que assenta no conceito de DataSet, ou seja, de ter uma estrutura em memória onde podem ser armazenados dados em formato de tabelas. Optou-se então por, depois de ler os dados das fontes de dados, os armazenar em DataTables dentro de um DataSet. Isto facilita a sua manipulação visto que um DataSet se comporta como uma base de dados e um DataTable como uma respectiva tabela dentro dessa base de dados. Na prática isto significa que é fácil aceder a um determinado registo pelo índice da linha e da coluna. Numa DataTable podem ser selecionados, adicionados, editados ou apagados determinados registos.

Em relação à parte mais visual do trabalho optou-se por mostrar os dados em estruturas do tipo DataGridView, que são basicamente grelhas que mostram dados provenientes de tabelas. Os DataGridView são estruturas que permitem uma interação com vários tipos de filtros próprios da linguagem C#, o que facilitou bastante o desenvolvimento da parte de pesquisa no sistema.

4.2. Classes

Como já referido na revisão da literatura, o paradigma de programação considerado neste trabalho foi o de programação orientada a objetos (POO). Optou-se por este paradigma porque a programação de sistemas complexos é mais simples se for divi-

didada em unidades básicas designadas por objetos, e respetivas classes. Além disto, a utilização de classes facilita a leitura e compreensão do código, torna a programação mais fluida e permite uma mais fácil reutilização, ou acréscimo de código no futuro. Neste caso, como o sistema que está a ser desenvolvido trabalha com uma interface visual composta por formulários, em que cada formulário é em si, uma classe, utilizar este paradigma faz ainda mais sentido.

Já a divisão do código em métodos dentro da própria classe é uma boa prática de programação que ajuda o programador a ter um código mais simples, perceptível, menos redundante e a saber sempre onde está tudo o que seja necessitar de usar.

4.2.1. Diagrama de Classes do Sistema

Conforme dito na revisão da literatura, usando POO, o diagrama de classes é uma parte imprescindível do *design* do sistema. É por isso aqui apresentado o diagrama de classes referente ao sistema em causa.

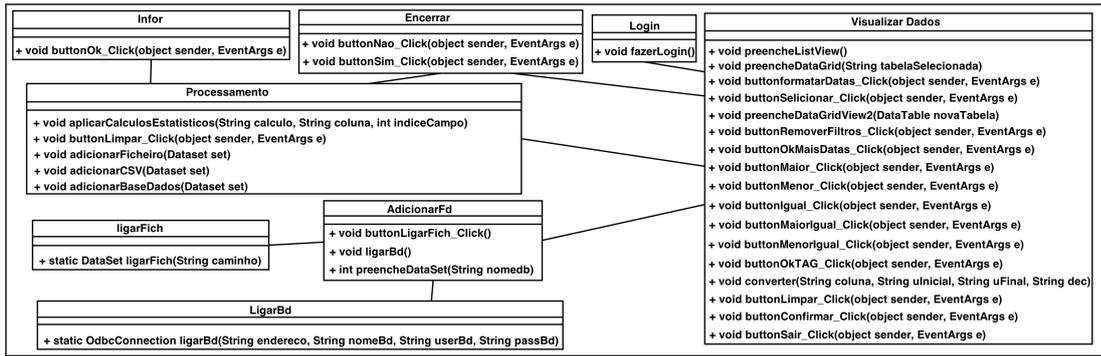


Figura 4.1: Diagrama de classes do sistema

Como se pode ver o diagrama de classes do sistema está constituído pelas classes: Login, ligarBd, LigaFich, AdionarFD, VisualizarDados, Processamento, Encerrar e a classe Info. De seguida é feita uma descrição individual de cada classe e dos seus principais métodos.

4.2.2. Classe Login

Esta classe serve para fazer o acesso ao sistema.

void fazerLogin (): Este método verifica se os dados de acesso introduzidos pelo utilizador são coerentes com os guardados no ficheiro de texto. Se forem, avança e permite acesso ao sistema. Senão forem, avisa o utilizador que os dados não estão corretos.

4.2.3. Classe ligarBd

Esta classe serve para fazer a ligação a uma base de dados.

static OdbcConnection ligaBd(String endereco, String nomebd, String userbd, String passbd): Este método abre uma ligação à base de dados, com os dados introduzidos pelo utilizador e retorna o objecto da ligação para ser validado e usado pela classe “AdicionarFd”.

4.2.4. Classe LigaFich

Esta classe serve para fazer o acesso aos ficheiros.

static DataSet ligaFich(String caminho): Este método abre um ficheiro ou a base de dados e retorna toda a sua informação para um DataSet.

4.2.5. Classe AdionarFD

Esta classe serve para adicionar dados de uma fonte de dados. Para isso chama as classes específicas para fazer a ligação a uma base de dados ou a um ficheiro, e verifica se os dados retornados por essas classes são ou não válidos.

void buttonLigarFich_Click(object sender, EventArgs e): Este método verifica se as tabelas do DataSet retornado pela classe LigaFich estão no formato suportado pelo sistema. Se estiverem, avança para classe “VisualizarDados”. Se não estiverem, informa que as tabelas não foram carregadas por não estarem de acordo com sistema.

void ligarBd(): Este método verifica se objecto OdbcConnection retornado pela classe LigaBd é válido. Se não for, dá um erro de ligação à base de dados. Se for, faz uma pesquisa na base de dados para ler o esquema das tabelas lá existentes e construir uma lista com o nome das mesmas. Seguidamente, chama o método preencheDataSet que retorna se a ligação é ou não possível, por causa das compatibilidades dos dados da base de dados com o sistema.

int preencheDataSet(String nomebd): Este método é o que vai buscar os dados às tabelas da base de dados e verifica se os dados de cada tabela estão de acordo com os que são suportados pelo sistema. Se estiverem, coloca-os no DataSet e coloca a variável “Ok” a 0 para informar que tudo está compatível. Se não estiverem, coloca a variável “Ok” a 1, para informar que não é possível carregar os dados.

4.2.6. Classe VisualizarDados

Esta classe serve para mostrar os dados de todas as tabelas adicionadas ao sistema e para sobre eles efetuar filtros e conversões. É também responsabilidade desta classe fazer a associação entre o ID e a sua descrição, bem como a definição de uma unidade para os dados a serem manipulados. É ainda esta classe que é responsável pela construção da tabela unificada.

void preencheListView(): Este método serve para preencher a lista dos nomes das tabelas adicionadas ao sistema.

void preencheDataGrid (String tabelaSelecionada): Este método serve para preencher o DataGridView com os dados da tabela selecionada na lista de tabelas. É também neste método que se fazem algumas verificações associadas ao campo data existente na tabela.

void buttonFormatarDatas_Click(object sender, EventArgs e): Este método serve para converter a data do formato apresentado para o formato desejado. Por exemplo, se o campo data vier com data e hora, o utilizador pode converter para o formato de hora apenas.

void buttonSelecionar_Click(object sender, EventArgs e): Este método serve para adicionar os dados da tabela selecionada para a DataTable correspondente à tabela unificada, fazendo todas as verificações necessárias à compatibilidade de dados.

void preencheDataGridView2 (DataTable novaTabela): Este método serve para preencher o DataGridView com dados da DataTable correspondente à tabela unificada

void buttonRemoveFiltros_Click(object sender, EventArgs e): Este método serve para limpar todos os filtros aplicados à tabela selecionada, repondo assim a tabela original.

void buttonOkMaisDatas_Click(object sender, EventArgs e): Este método serve para aplicar à tabela selecionada um filtro de datas. Este filtro é aplicado segundo o intervalo de datas escolhido pelo utilizador.

void buttonMaior_Click(object sender, EventArgs e): Este método é semelhante aos métodos void buttonMenor_Click(object sender, EventArgs e), void buttonIgual_Click(object sender, EventArgs e), void buttonMaiorIgual_Click(object sender, EventArgs e) e void buttonMenorIgual_Click(object sender, EventArgs e). Estes métodos servem para aplicar à tabela selecionada filtros numéricos ou de horas.

Este último só é possível caso o campo data da tabela tenha apenas hora.

void buttonOkTAG_Click(object sender, EventArgs e): Este método serve para criar a associação entre o ID e sua descrição. É também este método que, com base na descrição selecionada, vai preencher os tipos de unidade disponíveis para associação e conversão.

void converter (String coluna, String uInicial, String uFinal, String dec): Este método serve para aplicar as conversões desejadas aos dados do campo escolhido da tabela selecionada.

void buttonLimpar_Click(object sender, EventArgs e): Este método serve para limpar os dados da DataTable e DataGridView associadas à tabela unificada e limpar a seleção de dados da tabela selecionada.

void buttonConfirmar_Click(object sender, EventArgs e): Este método serve para enviar a DataTable da tabela unificada para classe “Processamento”.

void buttonSair_Click(object sender, EventArgs e): Este método serve para terminar o sistema.

4.2.7. Classe Processamento

Esta classe serve para efetuar cálculos estatísticos sobre os dados anteriormente selecionados. É também aqui que os dados são guardados de forma permanente.

void aplicarCalculosEstatisticos (String calculo, String coluna, int indiceCampo): Este método recebe o cálculo que vai ser efetuado, a coluna sobre a qual esse cálculo é aplicada e o índice correspondente a essa coluna dentro da comboBox onde se escolhe a coluna. O cálculo estatístico escolhido é aplicado usando os dois primeiros parâmetros recebido no método, sendo escolhida a fórmula estatística adequada ao mesmo. O último parâmetro recebido serve para impedir que o mesmo cálculo sobre a mesma coluna seja efetuado mais que uma vez, evitando assim resultados repetidos.

void buttonLimpar_Click(object sender, EventArgs e): Este método serve para limpar todos cálculos efetuados e todas as estruturas auxiliares que guardam a informação sobre quais os cálculos já efetuados.

void adicionarFicheiro (DataSet set): Este método cria um ficheiro Excel com duas folhas. Numa delas coloca todos os dados da tabela unificada e na outra todos os dados da tabela de cálculos.

void adicionarCSV(DataSet set): Este método cria um ficheiro do tipo CSV,

onde coloca a tabela unificada e tabela dos cálculos efetuados.

void adicionarBaseDados(DataSet set): Este método insere os dados da tabela unificada e da tabela de cálculos, nas respectivas tabelas já criadas na base de dados.

4.2.8. Classe Encerrar

void buttonNao_Click(object sender, EventArgs e): Este método serve para permanecer no sistema.

void buttonSim_Click(object sender, EventArgs e): Este método serve para encerrar o sistema.

4.2.9. Classe Info

void buttonOk_Click(object sender, EventArgs e): Este método serve para fechar a própria classe e voltar ao classe “Processamento”.

4.3. Ambiente de Trabalho

Com base na análise de requisitos e nos *mock-ups* funcionais, foi desenvolvido um ambiente de trabalho interativo para o protótipo que permite, de uma forma simples, testar as funcionalidades previstas. Como não se trata de um produto industrial, e não teve a colaboração de um *designer* gráfico, o foco do desenvolvimento do ambiente de trabalho foi a facilidade de execução das tarefas. Por isso, a interação com o sistema utiliza essencialmente o rato.

A interface tira partido de botões (para a interação), “*drop down boxes*”, mensagens informativas (sempre que necessário) e caixas de texto para colocar os dados de *login*, os dados de acesso à base de dados e os parâmetros de pesquisa.

A título de exemplo, apresentam-se alguns ecrãs ilustrativos do protótipo desenvolvido.

4.3 Ambiente de Trabalho

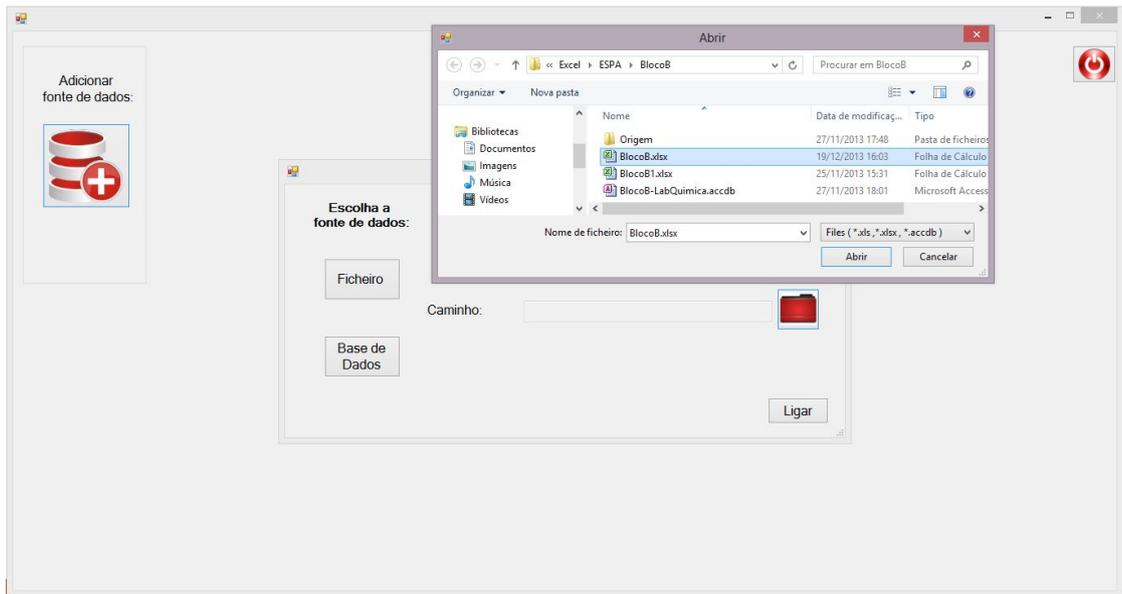


Figura 4.2: Fonte de dados “ficheiro”, através da sua localização

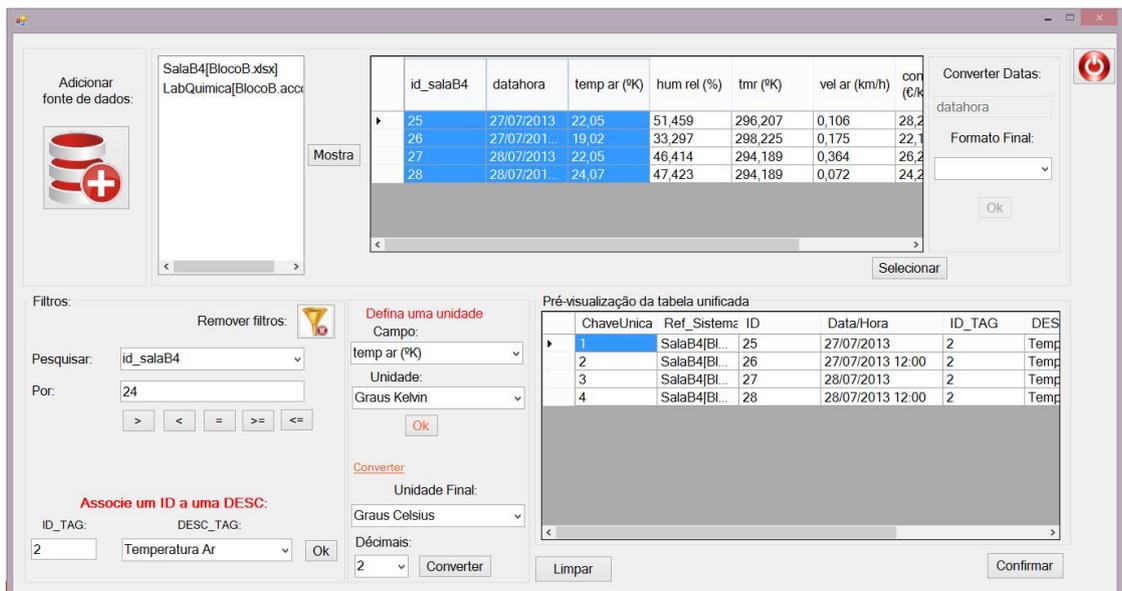


Figura 4.3: Tabela de fonte de dados, seleção de dados e tabela unificada

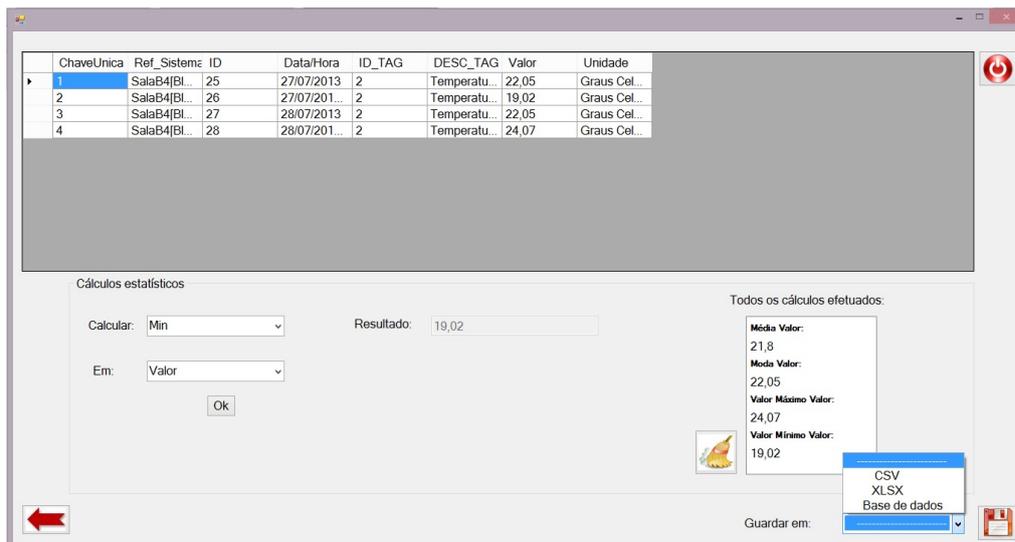


Figura 4.4: Processamento estatístico de dados

4.4. Testes

O plano de testes foi baseado na Análise de Requisitos e no *storyboard* do sistema.

No desenvolvimento deste trabalho, apenas foram aplicados testes funcionais e de aceitação. Todos os testes foram especificados de acordo com um *template* pré-definido [35], que contempla os seguintes campos:

Número do teste: A identificação atribuída a cada teste. O número do teste começa com T e de seguida pode ter um ou dois algarismos, conforme a sequência de testes.

Objetivo: É o propósito do teste.

Condições de execução: Descreve a localização necessária na aplicação aquando da realização do teste.

Entradas: Dados submetidos no(s) campo(s) a testar.

Saídas esperadas: Descreve o comportamento que a aplicação deverá devolver. No caso de o teste gerar um erro, este deverá ser corrigido tendo em conta a mensagem que iremos obter do programa.

Referência ao protótipo: Indica o que estamos concretamente a testar. Os valores poderão ser: nome do botão, *link* ou seção.

Resultado: Contém o registo do comportamento do sistema em relação ao que estava previsto. Este campo é apenas preenchido aquando da realização dos testes de aceitação.

Os testes constantes do plano de testes encontra-se no anexo B.

Capítulo 5

Caso de Estudo

Neste capítulo será apresentado um caso de estudo, que simula uma situação real, destinado a mostrar a aplicabilidade do trabalho desenvolvido, considerando a satisfação dos requisitos enumerados. O caso de estudo apresentado é inspirado numa instituição real, tendo sido simulados todos os dados e medições apresentadas.

5.1. Ambiente do caso de estudo

Considere-se a Escola Secundária Profissional A (ESPA) que é uma escola que leciona cursos técnicos e profissionais de eletricidade, química, mecânica e construção civil.

Fisicamente, a escola é constituída por duas portarias, duas garagens, um edifício central, dois blocos A e B, três campos de jogos, um departamento médico, um refeitório social, dois bares e vários WCs.

O edifício central encontra-se dividido em três alas: esquerda, centro e direita, cada uma delas com 3 pisos. A ala esquerda é constituída por um laboratório de química e um de eletrónica/telecomunicações, um arrumo para cada laboratório, duas salas de aulas especiais, uma sala de estudo, um anfiteatro, quatro WCs e um arrumo geral. Ala centro é constituída por uma secretaria-geral, dois arrumos para documentos administrativos, seis WCs, uma sala do concelho executivo, uma sala para o diretor, uma sala para o subdiretor, uma sala de reunião, uma sala de professores e uma sala de administração. A ala direita é constituída por um ginásio, um balneário, um arrumo de apoio ao ginásio, um bar dos professores, uma biblioteca, quatro WCs, uma sala de convívio, dois arrumos gerais.

O bloco A é constituído por salas de aula normais. O bloco B é também constituído por salas de aula normais, laboratórios de química e oficinas de mecânica, eletricidade e de construção civil. Ao redor dos blocos A e B existem vários WCs.

O Departamento Médico é constituído por uma sala de análises clínicas, uma receção, uma gabinete médico e uma sala para os primeiros socorros.

O refeitório é constituído por um bar, uma cozinha, uma sala de refeições, dois

WCs e um arrumo para guardar a comida.

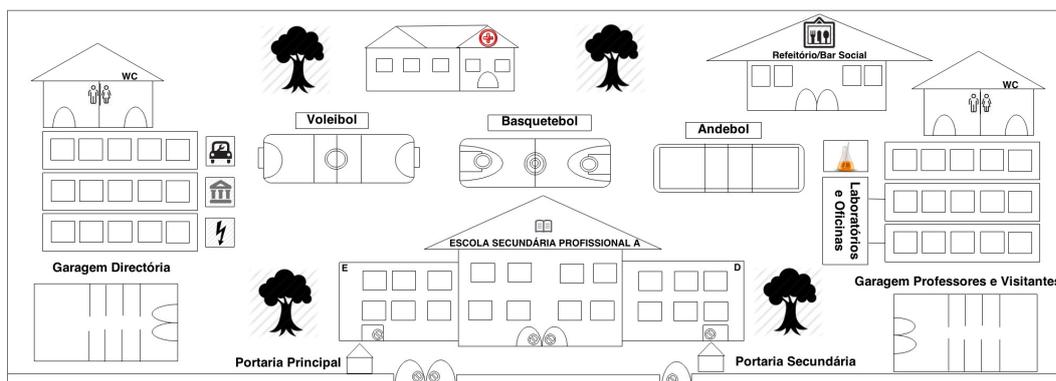


Figura 5.1: Escola Secundária Profissional A (ESPA)

5.2. Contextualização do Problema

A ESPA é uma escola bastante conceituada com mais de 6.000 alunos oriundos de todo o País. Para além dos cuidados com a qualidade de ensino, a escola tem muito cuidado com a qualidade do ambiente dentro dos vários espaços da escola.

A escola tem um sistema que mede variáveis como a temperatura, saturação de dióxido de carbono, consumo de energia, etc. Mas este sistema foi sendo implementado faseadamente e como tal, apresenta a medição dos dados, para o mesmo tipo de espaço, em unidades diferentes. Esta situação provoca bastante incómodo para quem tem de analisar comparativamente a qualidade do ambiente, visto que não é imediata a comparação entre as condições em espaços semelhantes.

Por isso, o objetivo deste cenário aplicacional é enquadrar a aplicação que leia os dados gerados pelos diferentes sistemas de medição existentes na escola, uniformize os dados de lá provenientes de forma a permitir comparações entre espaços semelhantes, faça alguns cálculos estatísticos sobre os mesmos e exporte a informação uniformizada e o resultado de alguns cálculos estatísticos para um novo ficheiro ou base de dados. Esta informação integrada e uniformizada serve de apoio à gestão corrente e estratégica da escola.

5.3. Descrição de Exemplos do Problema

Um exemplo que despoletou a necessidade do sistema de monitorização foi a gestão dos consumos de energia, de forma a fazer poupanças na conta da eletricidade, em especial nas cozinhas, refeitório social e bar privado.

Outra situação também importante é a monitorização da temperatura nos diferentes espaços, como as salas de aulas, biblioteca e secretaria-geral, para assegurar o conforto de todos os alunos, funcionários e docentes da escola.

No entanto, como se pode verificar nos cabeçalhos das tabelas 5.1 e 5.4, as medições foram feitas em unidades diferentes. Utilizando o desenvolvimento é possível gerar tabelas onde os valores aparecem filtrados nos períodos temporais pretendidos, com as unidades uniformizadas. Estas tabelas unificadas geradas pelo o sistema são apresentadas no anexo C.

A título exemplificativo é apresentado, também no anexo C, um excerto das tabelas originais. Embora o sistema tenha lidado com um número maior de dados, apresenta-se aqui apenas uma pequena parte das tabelas. O intuito desta amostra é observar o estado original dos dados, sendo assim desnecessário apresentar centenas de linhas de dados. Esta situação é repetida para todos os exemplos.

Há ainda a referir que para cada exemplo são apresentadas as tabelas dos cálculos estatísticos efetuados sobre os dados das tabelas unificadas. Apresenta-se um primeiro exemplo usando as tabelas geradas pelos sistemas de monitorização das cozinhas.

Exemplo 1:

É pretendido com estas tabelas fazer uma estimativa dos níveis médios do consumo de energia nas duas cozinhas durante um dia de aulas e de um dia de férias, no período de funcionamento do estabelecimento (das 8h às 17h). A medição nestes dois períodos diferentes é importante para saber se poderá haver reduções de gastos quando há e não há alunos, docentes e funcionários na escola.

ID_ CozinhaBarPriv	ID_ cozinhaRefSocial
HORA	Data/Hora
TEMP AR (°C)	Temp Ar (°F)
CO2 (Max 3%)	CO2 (ppm)
HUM REL (%)	Hum Rel (%)
FUNGOS INT (MAX 1500)	Fungos (Max 1500)
BAC INT (MAX 1500)	Bact Tot (Max 1500)
CONS ENER (€/KWH)	Cons Ener (\$/kwj)
VEL AR (KM/H)	-

Tabela 5.1: Cabeçalhos das tabelas das cozinhas do refeitório social e do bar privado

Média (€)	Valor Máximo (€)	Valor Mínimo (€)	Moda (€)	Soma Total (€)
6,85	11,10	3,03	9,08	130,15

Tabela 5.2: Cálculos do consumo de energia durante um dia no período de aulas

Média(€)	Valor Máximo (€)	Valor Mínimo (€)	Moda (€)	Soma Total (€)
3,29	7,06	1,01	3,03	62,59
			2,02	

Tabela 5.3: Cálculos do consumo de energia durante um dia no período de férias

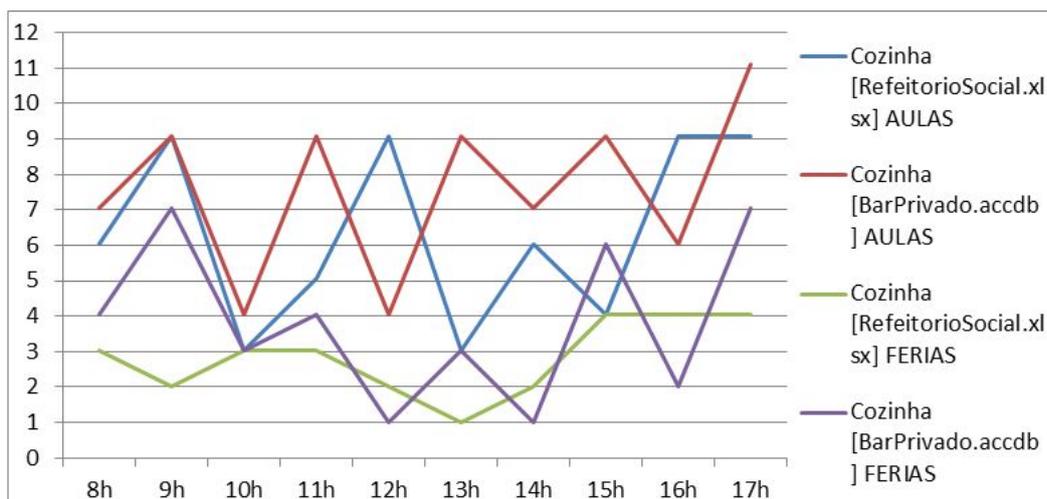


Figura 5.2: Custo instantâneo em euros da cozinha social e do bar privado durante um dia no período de aulas e férias

Conclusões:

Após a conversão das despesas para Euro/Kilowatt-hora e da seleção do período 8h-17h do dia 25-10-2012 (dia de aulas) pode agora observar-se a tabela C.14. Estudo idêntico foi feito para as mesmas cozinhas mas selecionando o dia 26-08-2013 (dia de férias) podendo os resultados ser observados na tabela C.16.

Analisando estas tabelas é possível estabelecer uma comparação com o que acontece nestas cozinhas em tempo de aulas e em tempo de férias e verificar que os custos geralmente são superiores em tempo de aulas, situação que já era de esperar devido à presença de mais pessoas e à preparação de um maior número de refeições.

Utilizando os dados das duas tabelas unificadas foi ainda possível criar o gráfico 5.2 onde se consegue comparar ao longo do dia, as mesmas cozinhas, em tempo de aulas e em tempo de férias. Daqui mais uma vez se conclui que os custos são no geral inferiores em tempo de férias. Olhando para o gráfico nota-se ainda que esta questão dos custos, especialmente em tempo de aulas, merece uma atenção especial uma vez que há picos em ambas as cozinhas. Há também o fato de que em tempo de aulas a soma dos custos das duas cozinhas ser mais do dobro do que em tempo de férias. Como em tempo de aulas as cozinhas funcionam mais, crê-se que algum tipo de aparelho possa estar a funcionar mal ou a necessitar de ser substituído pois pode estar a criar esses picos nos custos, fazendo com que no geral os custos subam muito também.

Exemplo 2:

Com as tabelas geradas pelo sistema pretende-se fazer uma estimativa dos níveis médios da temperatura do ar, em duas salas de aulas, na biblioteca e na secretaria-geral, durante uma semana no período de aulas (19-11-2012 até 25-11-2012) e durante uma semana no período de férias (22-07-2013 até 28-07-2013). A medição nestes dois períodos diferentes é importante para estabelecer uma possível melhoria do ambiente quando há e não há alunos, docentes e funcionários na escola.

Capítulo 5 Caso de Estudo

Id_SalaA2	id_salaB4	idSecretaria	Id_Biblioteca
Data	data/hora	Data_Hora	Data
TMR(°F)	tmr (°K)	TMR (°C)	TMR (°C)
Temp Ar (°F)	temp ar (°K)	Temp Ar (°C)	Temp Ar (°C)
Hum Rel (%)	hum rel (%)	Hum Rel (%)	Hum Rel (%)
Cons Energ (€/kwh)	cons ener (€/kwh)	Cons Ener \$/kwh	Cons Ener (€/kwh)
Vel Ar (m/s)	vel ar (km/h)	Vel Ar (Km/h)	Vel Ar (m/s)
CO2 (Max 3%)	Diox carbono (ppm)	CO2 (Max 3%)	CO2 (ppm)

Tabela 5.4: Cabeçalhos das tabelas das salas de aulas, secretaria-geral e biblioteca

Média (°C)	Valor Máximo (°C)	Valor Mínimo (°C)	Moda (°C)
19,64	31,02	9,08	15,99

Tabela 5.5: Cálculos da temperatura do ar nas salas (de aulas, bibliotecas e secretaria-geral) durante uma semana no período de aulas

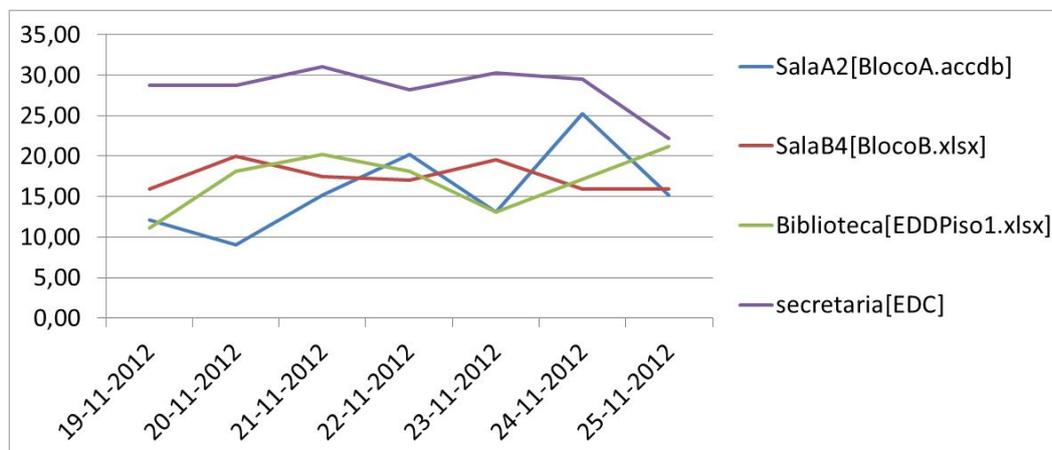


Figura 5.3: Temperatura do ar nas salas (de aulas, bibliotecas e secretaria-geral) durante uma semana no período de aulas

Média (°C)	Valor Máximo (°C)	Valor Mínimo (°C)	Moda (°C)
22,46	29,26	19,17	24,22

Tabela 5.6: Cálculos da temperatura do ar nas salas (de aulas, bibliotecas e secretaria-geral) durante uma semana no período de férias

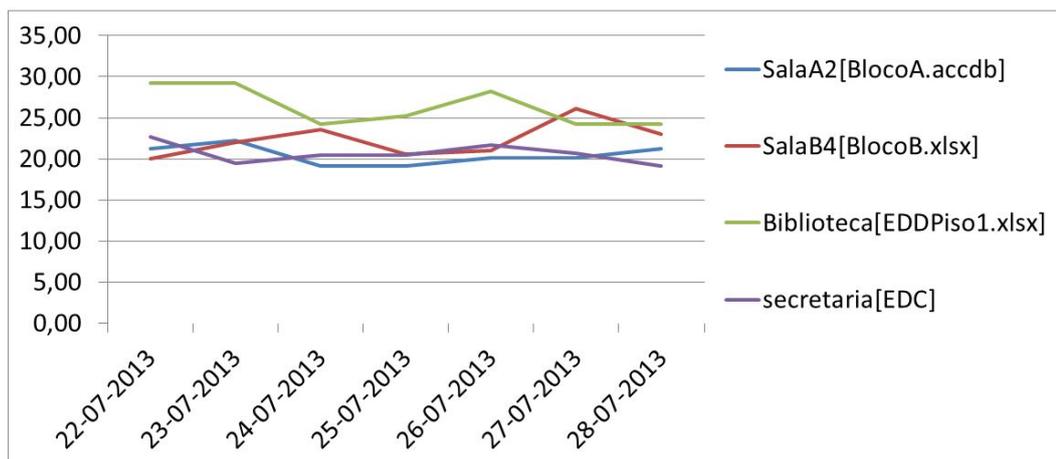


Figura 5.4: Temperatura do ar nas salas (de aulas, bibliotecas e secretaria-geral) durante uma semana no período de férias

Conclusões:

Após a conversão da temperatura para graus celsius e da seleção de uma semana no período de aulas pode agora observar-se a tabela C.18, com os dados da temperatura nas duas salas de aula, na biblioteca e na secretaria-geral. Estudo idêntico foi feito para os mesmos espaços, mas selecionando agora uma semana no período de férias. Os resultados podem ser observados na tabela C.20.

Utilizando os dados das duas tabelas unificadas foi ainda possível criar os gráficos 5.3 e 5.4 onde se consegue comparar ao longo da semana, os mesmos espaços, em tempo de aulas e em tempo de férias. Dos gráficos e das tabelas de cálculos respeitantes ao período de aulas (inverno) pode concluir-se que a temperatura é no geral mais baixa nas salas de aulas e na biblioteca e mais elevada na secretaria-geral. Dos gráficos e das tabelas de cálculos respeitantes ao período de férias (verão) pode concluir-se que a temperatura é no geral mais elevada nas salas de aulas e na biblioteca e mais baixa na secretária-geral.

Com estas observações pode concluir-se que a secretaria-geral é o único espaço devidamente climatizado e onde a temperatura quer seja verão ou inverno se encontra sempre em valores confortáveis.

Capítulo 6

Discussão, Conclusões e Recomendações

Findo este trabalho, é seguro dizer que o balanço do mesmo é bastante positivo. Muito foi aprendido e foram cumpridas as principais metas definidas no início. É possível ainda fazer um balanço mais aprofundado segundo várias perspectivas: ultrapassagem de dificuldades, cumprimento de metas e soluções propostas, e aprendizagem e possíveis melhoramentos no futuro.

6.1. Ultrapassagem de Dificuldades

Ao longo do desenvolvimento deste trabalho várias foram as dificuldades encontradas, em especial por ser uma área nova em vários aspetos, o que teve como consequência a necessidade de aprendizagem de raiz de alguns assuntos e o aprofundamento de outros conhecimentos.

Apesar da grande importância que tem para o desenvolvimento adequado de novas aplicações, a minha formação-base não contemplou a área da Engenharia de *Software*. Começando na análise de requisitos, passando pelos casos de uso, pelos testes e pelo *storyboard*, tudo isto fez parte de um processo de aprendizagem muito grande. Ultrapassada esta fase, avalio muito positivamente a necessidade desta aprendizagem considerando-me muito mais capacitada para encarar a minha vida profissional, em especial em tudo o que obrigue ao desenvolvimento de aplicações informáticas.

Seguidamente surgiram as naturais dificuldades relacionadas com a implementação. Apesar dos conhecimentos já adquiridos, tive que lidar com a complexidade própria da aplicação, nomeadamente no que diz respeito à manipulação de datas e a aspetos específicos da linguagem C#, que foi uma linguagem nova.

Finalmente, e como já esperava, houve necessidade de resolver vários problemas relacionados com a unificação de tabelas. Este desafio, um dos aspetos centrais do trabalho, teve a ver com o facto de as fontes de dados poderem ser diferentes e com diversos formatos de dados, o que obrigou a encontrar um ponto comum para que a

tabela unificada pudesse ser criada.

A criação de filtros foi menos complexa uma vez que a própria linguagem C# tem já definida em si um conjunto de estruturas e filtros que podem ser aplicados diretamente. As conversões e a aplicação de cálculos estáticos foram relativamente menos complicadas de implementar.

6.2. Cumprimento de Metas e Soluções Propostas

O objetivo inicialmente proposto de criar um protótipo capaz de mostrar uma forma de integrar várias fontes de informação, foi cumprido.

O protótipo desenvolvido é capaz de carregar tabelas de ficheiros ou de bases de dados, de entender datas que venham em formatos diferentes e, se assim se desejar, converte-las num mesmo formato. É ainda capaz de fazer conversões em campos que tenham a mesma descrição, mas venham em unidades diferentes, colocando-os na mesma unidade. Por fim, é ainda possível fazer alguns cálculos estatísticos sobre os dados unificados e guardar toda esta informação num novo ficheiro ou base de dados.

Como o protótipo desenvolvido é completamente transversal a qualquer área de negócio, tanto pode ser aplicado para medir as temperaturas de uma sala de aula, como pode ser usado para determinar os produtos mais vendidos num supermercado, num determinado período de tempo, etc.. Este resultado torna-o uma mais-valia para diferentes situações.

6.3. Aprendizagem e Possíveis Melhoramentos no Futuro

Em geral, e como já foi referido anteriormente, todo este processo de desenvolvimento da minha dissertação de mestrado foi de uma grande aprendizagem, por ter sido realizado numa área totalmente nova para mim. Desde a aplicação dos conceitos de engenharia de *software* no processo de desenvolvimento do trabalho até à implementação em C#, surgiram sucessivos desafios, que foram sendo superados.

Face às limitações temporais que tinha para a realização deste trabalho houve algumas funcionalidades que não puderam ser realizadas e faz todo o sentido que o venham a ser no futuro.

Uma funcionalidade identificada durante a fase de análise dos casos de estudo foi a possibilidade da criação e exportação de gráficos resultantes de cálculos estatísticos. Uma vez que se criam tabelas unificadas e se realizam cálculos estatísticos sobre elas, será muito interessante criar gráficos com esses dados e resultados, uma vez que a

visualização gráfica facilita a comparação e outros processos conclusivos.

Outro melhoramento que se pensa ser uma mais valia é o facto de a tabela unificada deixar de suportar apenas a escolha de um valor, mas passar a suportar a escolha de vários valores (por exemplo, ter na tabela unificada valores como temperatura, moeda, consumo de energia, etc.), podendo assim criar tabelas unificadas mais complexas, com um menor tempo de criação das mesmas.

Bibliografia

- [1] Abacuc: *Multi Database Tool - MySQL, SQLite, PostgreSQL*, 2013.
URL: <http://www.abacucapp.com/?lng=en#company> [Conferido em: 14 de Novembro de 2013].
- [2] ANSI/IEEE Std 830. *IEEE to Software Requirements Specifications*. U.S., 1984.
- [3] Booch Grady, Maksimchuk Robert, Engle Michael, Young Bobbi, Conallen Jim, Houston Kelli. *Object-Oriented Analysis and Design with Applications*. Third Edition, 2007.
- [4] Computerworld Portugal: *C# é linguagem de programação de 2012*, 2013.
URL: <http://www.computerworld.com.pt/2013/01/04/c-e-linguagem-de-programacao-de-2012/> [Conferido em: 26 de Novembro de 2013].
- [5] creativyst.com: *The Comma Separated Value (CSV) File Format*.
URL: <http://www.creativyst.com/Doc/Articles/CSV/CSV01.htm> [Conferido em: 28 de Janeiro de 2013].
- [6] Cunha Paulo. *Engenharia de Software 1, UML: Casos de Uso*. FCTUC, Coimbra 2005.
- [7] Damas Luís . *SQL - Structured Query Language* . FCA - Editora de Informática, Lisboa - Portugal, 2008.
- [8] Dbdeveloper: *Dbdeveloper - Database Management Tool*, 2010.
URL: <http://www.dbdeveloper.prominentus.com> [conferido em: 14 de Novembro de 2013].
- [9] DbVis Software AB: *About DbVisualizer™ “The Universal Database Tool”*, 2013.
URL: <http://www.dbvis.com> [conferido em: 14 de Novembro de 2013].

- [10] Dries Buytaert: *The history of MySQL AB*, 2013.
URL: <http://buytaert.net/the-history-of-mysql-ab> [conferido em: 30 de Dezembro de 2013].
- [11] Easysoft: *What is ODBC?*.
URL: http://www.easysoft.com/developer/interfaces/odbc/linux.html#what_is_odbc [Conferido em: 28 de Janeiro de 2013].
- [12] Geiger Kyle. *Insed ODBC*. Redmond: Microsoft Press, 1995.
- [13] Geiger Kyle. *International Software Testing Qualifications Board*. Certified Tester Foundation Level Syllabus. Section 1.1.2, 2011.
- [14] IBM Rational Unified Process (RUP): *Best practices for design, implementation and effective project management*, 2013.
URL: <http://www-01.ibm.com/software/rational/rup/> [conferido em: 30 de Dezembro de 2013].
- [15] Interaction Design Foundation: *Mock-ups*, 2013.
URL: <http://www.interaction-design.org> [conferido em: 17 de Dezembro de 2013].
- [16] ITL Education Solutions Limited. *Introduction to Computer Science*, 2011.
- [17] Kolawa Adam and Huizinga Dorota. *Automated Defect Prevention: Best Practices in Software Management*. Wiley-IEEE Computer Society Press, 2007, page 126.
- [18] Kotonya Gerald and Sommerville Ian. *Análise e Concepção de Sistema de Informação: Requisitos não funcionais*. Instituto Superior Técnico.
- [19] Marques Paulo, Pedrosa Hernâni, Figueira Ricardo. *C# 4.0*. FCA - Editora de Informática, Lisboa-Portugal, 2011.
- [20] *Microsoft: Access*, 2013.
URL: <http://office.microsoft.com> [conferido em: 26 de Novembro de 2013].
- [21] *MSDN*, 2012.
URL: <http://msdn.microsoft.com/en-us/library> [conferido em: 02 de Janeiro de 2013].

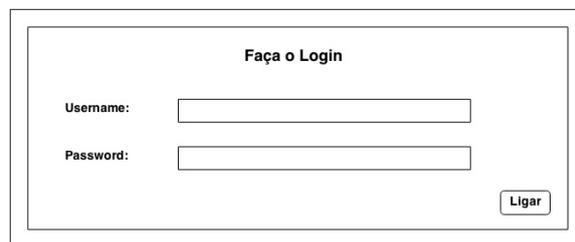
- [22] Navicat: *Products Info Navicat Premium*, 2013.
URL: <http://www.navicat.com> [conferido em: 14 de Novembro de 2013].
- [23] Neves Pedro, Ruas Rui. *Guia Prático do MySQL*. Lisboa-Portugal, 2005.
- [24] Open source initiative: *The Open source definition*, 2013.
URL: <http://opensource.org/> [conferido em: 14 de Dezembro de 2013].
- [25] *Oracle Provider for OLE DB Developer's Guide*. Release 1 (11.1.0.7.20) for Microsoft Windows, 2009.
- [26] Pinto Mário. *Microsoft Excel 2010*. Portugal, 2011.
- [27] Quaresma Pedro. *Base de Dados*. Apontamentos, FCTUC, Coimbra, 2010/2011.
- [28] Quaresma Pedro. *Programação Orientada a objetos*. Apontamentos, FCTUC, Coimbra, 2011/2012.
- [29] Ramalho José. *Extensible Markup Language (XML)*. Universidade do Minho, Portugal, 1999.
- [30] Rational Software Corporation 1987-2001: *Universal Data Access (UDA) Design*, 2012.
URL: http://www.wthreex.com/rup/process/modguide/md_udades.htm [conferido em: 29 de Dezembro de 2012].
- [31] Robertson Suzanne and Robertson James. *Mastering The Requirements Process*. Third Edition, 2013.
- [32] Silva Alberto e Videira Carlos. *UML, Metodologias e Ferramentas CASE*. Edições Centro Atlântico, Portugal, 2001.
- [33] Simba Technologies Inc.: *ODBC, 2012*.
URL: <http://www.simba.com/odbc.htm> [conferido em: 30 de Novembro de 2012].
- [34] Spínola Rodrigo, Araújo Marco e Spínola Eduardo. *Qualidade de software - Entenda os principais conceitos sobre testes e inspeção de software*. Edição especial, 2007.

- [35] Stellman Andrew and Greene Jennifer. *Applied Software Project Management*. O' Reilly, 2006.
- [36] Tolentino Ailton. *Interface para a Integração entre o OpenERP e o ERP Primavera*. Dissertação de Mestrado, Universidade de Aveiro, Portugal, 2011.
- [37] Walmsley Priscilla. *Definitive XML Schema*. New XML Schema 1.1, U.S., 2013.
- [38] Wiegers Karl E. *Software Requirements*. Second Edition, 2003.

Anexo A

A.1. Storyboard do Sistema

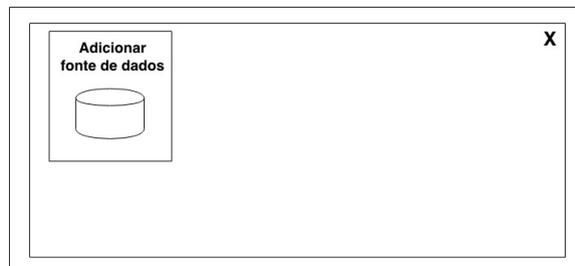
Ecrã 1: Login



The screenshot shows a login window with the title "Faça o Login". It features two text input fields: one labeled "Username:" and another labeled "Password:". A button labeled "Ligar" is positioned in the bottom right corner of the window.

Neste ecrã o utilizador efetua o *login* no sistema através de um *username* e de uma *password*.

Ecrã 2: Adicionar Dados



The screenshot shows a window titled "Adicionar fonte de dados" with a close button "X" in the top right corner. Inside the window, there is a button with a cylinder icon, which represents a data source.

Neste ecrã o utilizador poderá selecionar no botão “Adicionar fonte de dados”, onde irá abrir um outro ecrã de forma a escolher o tipo de fonte de dados que pretende adicionar ao sistema.

Também há a possibilidade de encerrar o sistema, bastando para isso, clicar no botão “Sair” que se encontra no canto superior direito.

Ecrã 3: Escolha o Tipo de Fonte de Dados

The screenshot shows a window titled "Escolha a fonte de dados". Inside the window, there are two buttons stacked vertically: "Ficheiro" on top and "Base de Dados" on the bottom.

Neste ecrã são apresentados dois botões ao utilizador, para que este escolha o tipo de fonte de dados à qual deseja ligar.

3.2 Acesso ao Ficheiro

The screenshot shows the "Escolha a fonte de dados" window. The "Ficheiro" button is selected. To its right, there is a text label "Caminho:" followed by an empty text input field and an "Abrir" button. Below the "Ficheiro" button is the "Base de Dados" button. At the bottom right of the window is a "Ligar" button.

O utilizador acede a este ecrã após ter clicado no botão "Ficheiro". Aqui consegue abrir o ficheiro onde está a informação que deseja aceder.

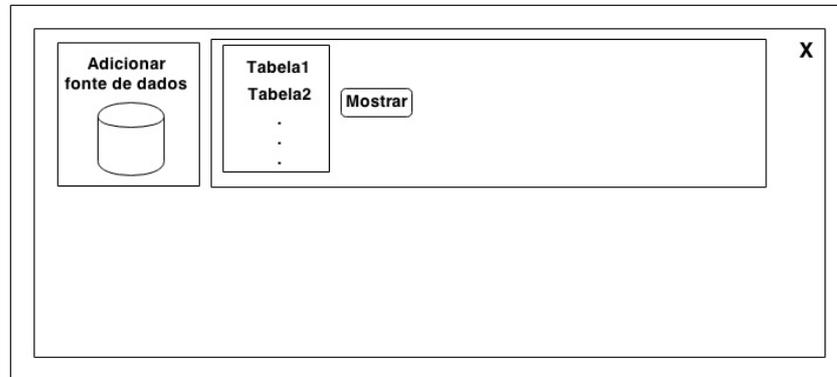
3.3 Ligação à Base de Dados

The screenshot shows the "Escolha a fonte de dados" window. The "Base de Dados" button is selected. To its right, there are four text input fields, each with a label: "Endereço:", "Nome da BD:", "User:", and "PassWord:". At the bottom right of the window is a "Ligar" button.

O utilizador acede a este ecrã após ter clicado no botão "Base de Dados". Aqui consegue introduzir os dados para se ligar à base de dados pretendida.

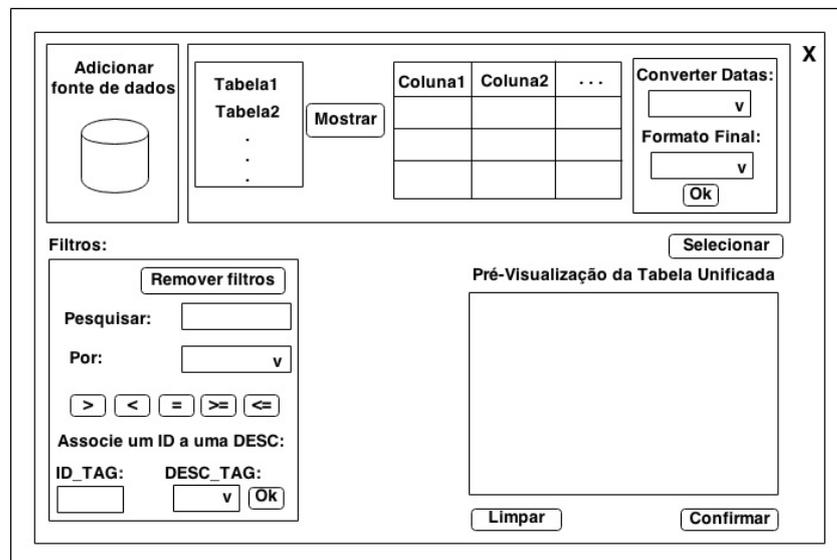
Ecrã 4: Selecionar Dados

4.1 Lista de Tabelas Disponíveis



Neste ecrã são apresentadas todas as tabelas disponíveis em todas fontes de dados às quais o utilizador se ligou. Aqui pode ser selecionada uma tabela de cada vez para posterior visualização do seu conteúdo, quando se carrega no botão “Mostrar”.

4.2 Associar Dados

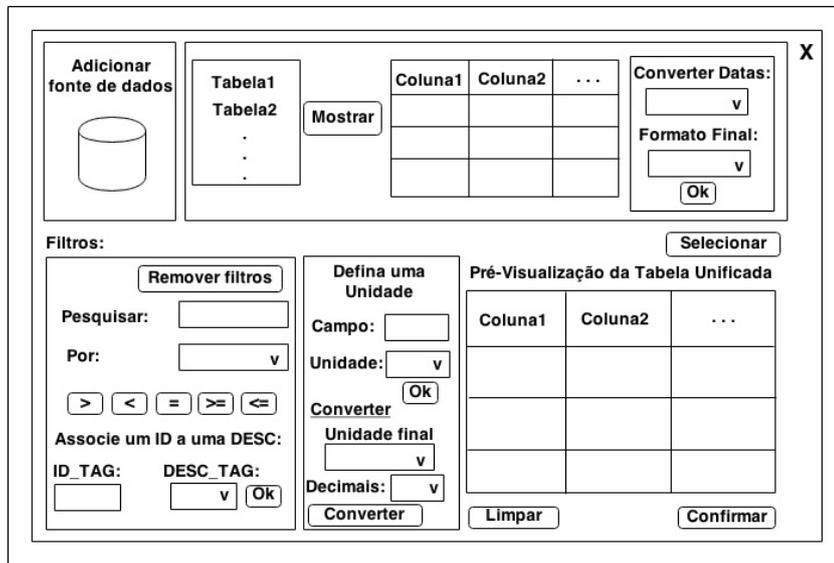


Neste ecrã é apresentado ao utilizador o conteúdo da tabela anteriormente escolhida.

Neste momento, caso seja necessário, o utilizador pode aplicar um filtro por ID ou por descrição da grandeza. Após esse filtro deve ser associado o ID da grandeza à descrição da mesma.

É ainda possível aplicar filtros por outros campos da tabela e ainda converter as datas para ficarem num dos formatos selecionados.

4.3 Definir uma Unidade e Selecionar os Dados



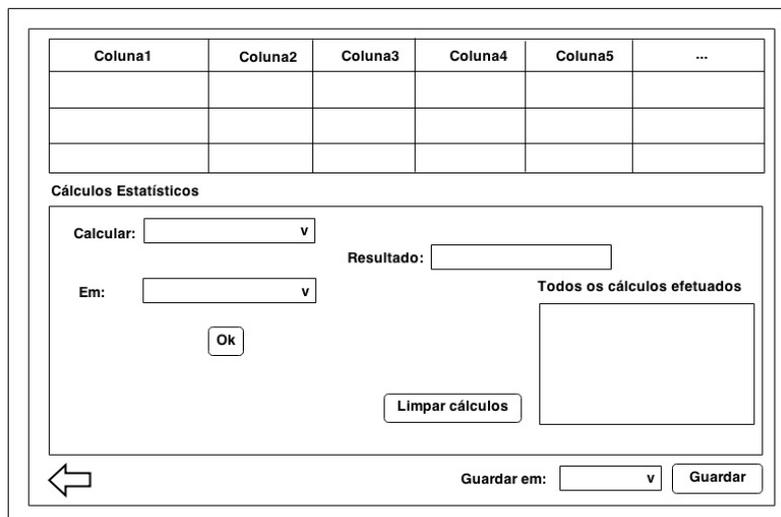
Neste ecrã o utilizador pode definir a unidade dos dados apresentados e pode ainda converter dessa unidade para outra selecionada.

Por fim, o utilizador deve carregar no botão “Selecionar” para que os dados apresentados na tabela superior sejam apresentados na tabela unificada.

Quando se obtiver a tabela unificada no formato pretendido, clicando no botão “Confirmar” é possível avançar para o ecrã seguinte.

Caso a tabela unificada não tenha ficado no formato pretendido, clicando no botão “Limpar” esta fica de novo vazia.

Ecrã 5: Aplicação de Funções de Processamento Estatístico



Neste ecrã o utilizador pode efetuar cálculos estatísticos sobre os dados previamente selecionados. Caso deseje, o utilizador poderá também voltar ao ecrã anterior, clicando na seta que se encontra no canto inferior esquerdo. Após efetuar todos os cálculos que deseje, o utilizador deve escolher como deseja guardar os dados e seguidamente deve clicar em “Guardar” para saltar para o próximo ecrã.

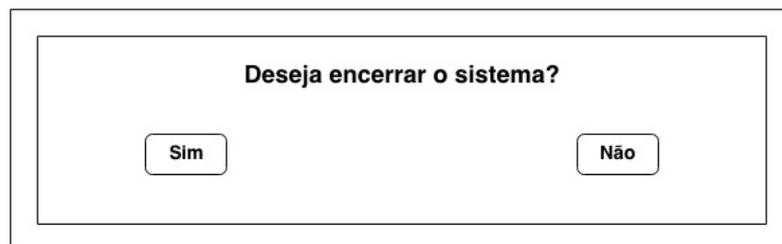
Também há a possibilidade de encerrar o sistema, bastando clicar no botão “Sair” que se encontra no canto superior direito.

Ecrã 6: Ecrã Informativo



Este ecrã aparece quando o utilizador clica no botão “Guardar”. É um ecrã unicamente informativo para que o utilizador perceba que todos os seus dados foram registados com sucesso, e clique no botão “Ok” de forma a confirmar.

Ecrã 7: Encerrar o Sistema



O utilizador chega a este ecrã após clicar no botão “sair”. Se clicar no botão do “Não” volta para o ecrã 2 ou 4 (caso esteja num destes ecrãs). Caso clique no “Sim” o sistema termina.

Anexo B

B.1. Plano de Testes

Nº de teste	T1
Objetivo	Validar utilizador
Condições de execução	1. Estar no ecrã de login
Entradas	1. Preencher apenas uma ou nenhuma das <i>textBox</i> 2. Clicar no botão de validar
Saídas Esperadas	1. Mensagem de erro: "Preencha todos os campos!"
Ref. ao protótipo	Ecrã "Faça o Login"> <i>textBox</i> " <i>Username</i> " e " <i>PassWord</i> "> botão "Validar"
Resultado	Validado

Nº de teste	T2
Objetivo	Validar utilizador
Condições de execução	1. Estar no ecrã de login
Entradas	1. Preencher as <i>textBox</i> com <i>username</i> e/ou <i>password</i> incorretos 2. Clicar no botão de validar
Saídas Esperadas	1. Mensagem de erro: "Erro ao ligar-se ao sistema. Tente novamente!"
Ref. ao protótipo	Ecrã "Faça o Login"> <i>textBox</i> " <i>Username</i> " e " <i>PassWord</i> "> botão "Validar"
Resultado	Validado

Nº de teste	T3
Objetivo	Validar utilizador
Condições de execução	1. Estar no ecrã de login
Entradas	1. Preencher as <i>textBox</i> com <i>username</i> e <i>password</i> corretos

Anexo B

	2. Clicar no botão de validar
Saídas Esperadas	1. Avança para o ecrã “Adicionar fonte de dados”
Ref. ao protótipo	Ecrã “Faça o Login”> textBox “ <i>Username</i> ” e “ <i>PassWord</i> ”> botão “Validar”
Resultado	Validado

Nº de teste	T4
Objetivo	Adicionar fonte de dados ao sistema
Condições de execução	1. Estar no ecrã “Adicionar fonte de dados”
Entradas	1. Clicar no botão “Adicionar fonte de dados”
Saídas Esperadas	1. Aparece um <i>pop-up</i> onde se escolhe a fonte de dados
Ref. ao protótipo	Ecrã “Adicionar fonte de dados”> botão “Adicionar fonte de dados”
Resultado	Validado

Nº de teste	T5
Objetivo	Visualizar a textBox onde aparece o caminho para aceder ao ficheiro e o botão para abrir esse caminho de acesso
Condições de execução	1. Estar no <i>pop-up</i> “Escolha a fonte de dados”
Entradas	1. Clicar no botão “Ficheiro”
Saídas Esperadas	1. Aparecimento da textBox onde é mostrado o caminho para aceder ao ficheiro 2. Aparecimento do botão para abrir o caminho de acesso ao ficheiro
Ref. ao protótipo	<i>Pop-up</i> “Escolha a fonte de dados”> botão “Ficheiro”
Resultado	Validado

Nº de teste	T6
Objetivo	Aceder ao ficheiro
Condições de execução	1. Estar no <i>pop-up</i> para escolher a fonte de dados e ter visível a textBox onde aparece o caminho do ficheiro e o botão para abrir o caminho de acesso ao ficheiro
Entradas	1. Não clicar no botão para abrir o caminho de acesso

	2. Clicar no botão “Ligar”
Saídas Esperadas	1. Mensagem de erro: “Clique sobre a pasta para escolher o ficheiro!”
Ref. ao protótipo	<i>Pop-up</i> “Escolha a fonte de dados”> botão “Ligar”
Resultado	Validado

Nº de teste	T7
Objetivo	Aceder ao ficheiro
Condições de execução	1. Estar no <i>pop-up</i> para escolher a fonte de dados e ter visível a <i>textBox</i> onde aparece o caminho do ficheiro e o botão para abrir o caminho de acesso ao ficheiro
Entradas	1. Clicar no botão para abrir o caminho de acesso 2. Escolher um ficheiro não compatível com o sistema 3. Clicar no botão “Ligar”
Saídas Esperadas	1. Mensagem de erro: "Existem tabelas que não estão de acordo com o sistema. Ficheiro não carregado."
Ref. ao protótipo	<i>Pop-up</i> “Escolha a fonte de dados”> <i>textBox</i> “Caminho”> botão para aceder ao caminho do ficheiro> botão “Ligar”
Resultado	Validado

Nº de teste	T8
Objetivo	Aceder ao ficheiro
Condições de execução	1. Estar no ecrã para escolher a fonte de dados e ter visível a <i>textBox</i> onde aparece o caminho do ficheiro e o botão para abrir o caminho de acesso ao ficheiro
Entradas	1. Clicar sobre o botão para abrir o caminho de acesso ao ficheiro 2. Escolher um ficheiro compatível com o sistema 3. Clicar no botão “Ligar”
Saídas Esperadas	1. O <i>pop-up</i> para escolher a fonte de dados desaparece. Volta ao ecrã onde podem agora ser visualizadas as tabelas disponíveis para seleção de dados

Anexo B

Ref. ao protótipo	<i>Pop-up</i> “Escolha a fonte de dados”> textBox “Caminho”> botão para aceder ao caminho do ficheiro> botão “Ligar”
Resultado	Validado

Nº de teste	T9
Objetivo	Visualizar as textBox onde se colocam os dados de acesso à base de dados
Condições de execução	1. Estar no <i>pop-up</i> para escolher a fonte de dados
Entradas	1. Clicar no botão “Base de Dados”
Saídas Esperadas	1. Aparecimento das textBox para colocar os dados de acesso à base de dados
Ref. ao protótipo	<i>Pop-up</i> “Escolha a fonte de dados”> botão “Base de Dados”
Resultado	Validado

Nº de teste	T10
Objetivo	Ligar à base de dados
Condições de execução	1. Estar no <i>pop-up</i> para escolher a fonte de dados e ter visível as textBox para se ligar à base de dados
Entradas	1. Não preencher uma ou todas as textBox “Endereço”, “Nome da base de dados”, “User” e “PassWord” 2. Clicar no botão “Ligar”
Saídas Esperadas	1. Mensagem de Erro: “Preencha todos os campos!”
Ref. ao protótipo	<i>Pop-up</i> “Escolha a fonte de dados”> textBox “Endereço”, “Nome da base de dados”, “User” e “PassWord”> botão “Ligar”
Resultado	Validado

Nº de teste	T11
Objetivo	Ligar à base de dados
Condições de execução	1. Estar no <i>pop-up</i> para escolher a fonte de dados e ter visível as textBox para se ligar à base de dados
Entradas	1. Preencher uma ou todas as textBox “Endereço”, “Nome da base de dados”, “User” e “PassWord” com dados incorretos

B.1 Plano de Testes

	2. Clicar no botão “Ligar”
Saídas Esperadas	1. Mensagem de Erro: “Erro ao ligar-se à base de dados. Tente Novamente!”
Ref. ao protótipo	<i>Pop-up</i> “Escolha a fonte de dados”> textBox “Endereço”, “Nome da base de dados”, “User” e “PassWord”> botão “Ligar”
Resultado	Validado

Nº de teste	T12
Objetivo	Ligar à base de dados
Condições de execução	1. Estar no <i>pop-up</i> para escolher a fonte de dados e ter visível as textBox para se ligar à base de dados
Entradas	1. Preencher todas as textBox “Endereço”, “Nome da base de dados”, “User” e “PassWord” com dados corretos 2. Escrever na textBox “Nome da base de dados” uma base de dados não compatível com o sistema 3. Clicar no botão “Ligar”
Saídas Esperadas	1. Mensagem de erro: "Existem tabelas que não estão de acordo com o sistema. Base de dados não carregada."
Ref. ao protótipo	<i>Pop-up</i> “Escolha a fonte de dados”> textBox “Endereço”, “Nome da base de dados”, “User” e “PassWord”> botão “Ligar”
Resultado	Validado

Nº de teste	T13
Objetivo	Ligar à base de dados
Condições de execução	1. Estar no <i>pop-up</i> para escolher a fonte de dados e ter visível as textBox para se ligar à base de dados
Entradas	1. Preencher todas as textBox “Endereço”, “Nome da base de dados”, “User” e “PassWord” com dados corretos 2. Escrever na textBox “Nome da base de dados” uma base de dados compatível com o sistema 3. Clicar no botão “Ligar”

Saídas Esperadas	1. O <i>pop-up</i> para escolher a fonte de dados desaparece. Volta ao ecrã onde podem agora ser visualizadas as tabelas disponíveis para seleção de dados
Ref. ao protótipo	Pop-up “Escolha a fonte de dados”> textBox “Endereço”, “Nome da base de dados”, “User” e “PassWord”> botão “Ligar”
Resultado	Validado

Nº de teste	T14
Objetivo	Mostrar os detalhes de uma tabela selecionada
Condições de execução	1. Estar no ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados 2. Ter tabelas na lista de tabelas disponíveis
Entradas	1. Selecionar múltiplas tabelas na lista de tabelas disponíveis
Saídas Esperadas	1. O sistema não permite múltipla seleção
Ref. ao protótipo	Ecrã “Lista de tabelas disponíveis”
Resultado	Validado

Nº de teste	T15
Objetivo	Mostrar os detalhes de uma tabela selecionada
Condições de execução	1. Estar no ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados 2. Ter tabelas na lista de tabelas disponíveis
Entradas	1. Selecionar uma tabela na lista de tabelas disponíveis 2. Clicar no botão “Mostrar”
Saídas Esperadas	1. O sistema mostra o conteúdo da tabela selecionada
Ref. ao protótipo	Ecrã “Lista de tabelas disponíveis”> botão “Mostrar”
Resultado	Validado

Nº de teste	T16
Objetivo	Converter datas na tabela apresentada
Condições de execução	1. Estar no ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados

	<ol style="list-style-type: none"> 2. Ter visível o conteúdo de uma tabela 3. Ter na textBox “Campo”, o nome da coluna sobre a qual é possível fazer a conversão
Entradas	<ol style="list-style-type: none"> 1. Selecionar o formato para o qual se deseja converter na comboBox “Formato Final” 2. Clicar no botão “Ok”
Saídas Esperadas	<ol style="list-style-type: none"> 1. O sistema faz a conversão pretendida e mostra o resultado na coluna respetiva da tabela apresentada
Ref. ao protótipo	Ecrã “Lista de tabelas disponíveis”> visualizar os dados da tabela selecionada> textBox “Campo”> comboBox “Formato Final”> botão “Ok”
Resultado	Validado

Nº de teste	T17
Objetivo	Filtros
Condições de execução	<ol style="list-style-type: none"> 1. Estar no ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados
Entradas	<ol style="list-style-type: none"> 1. Não escolher nenhum campo na comboBox “Pesquisar” ou preencher esta comboBox com caracteres 2. Escrever caracteres na textBox “Por”
Saídas Esperadas	<ol style="list-style-type: none"> 1. Mensagem de erro: "Selecionar o campo pretendido!"
Ref. ao protótipo	Ecrã “Lista de tabelas disponíveis”> comboBox “Pesquisar”> textBox “Por”
Resultado	Validado

Nº de teste	T18
Objetivo	Filtros
Condições de execução	<ol style="list-style-type: none"> 1. Estar no ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados
Entradas	<ol style="list-style-type: none"> 1. Escolher um campo de tipo numérico na comboBox “Pesquisar”
Saídas Esperadas	<ol style="list-style-type: none"> 1. Aparecem os botões de “>”, “<”, “=”, “>=”, “<=”
Ref. ao protótipo	Ecrã “Lista de tabelas disponíveis”> comboBox “Pesquisar”

Resultado	Validado
------------------	----------

Nº de teste	T19
Objetivo	Filtros
Condições de execução	1. Estar no ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados
Entradas	1. Escolher um campo do tipo numérico na comboBox “Pesquisar” 2. Colocar caracteres na textBox “Por” 3. Clicar sobre um dos botões “>”, “<”, “=”, “>=”, “<=”
Saídas Esperadas	1. Não acontece nada
Ref. ao protótipo	Ecrã “Lista de tabelas disponíveis”> comboBox “Pesquisar”> textBox “Por”> botões “>”, “<”, “=”, “>=”, “<=”
Resultado	Validado

Nº de teste	T20
Objetivo	Filtros
Condições de execução	1. Estar no ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados
Entradas	1. Escolher um campo do tipo numérico na comboBox “Pesquisar” 2. Escrever números com vírgulas na textBox “Por” 3. Clicar sobre os botões “>”, “<”, “=”, “>=”, “<=”
Saídas Esperadas	1. Mensagem de erro: "Os valores decimais são com ponto e não virgula!"
Ref. ao protótipo	Ecrã “Lista de tabelas disponíveis”> comboBox “Pesquisar”> textBox “Por”> botões “>”, “<”, “=”, “>=”, “<=”
Resultado	Validado

Nº de teste	T21
Objetivo	Filtros
Condições de execução	1. Estar no ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados

Entradas	<ol style="list-style-type: none"> 1. Escolher um campo do tipo numérico na comboBox “Pesquisar” 2. Colocar números na textBox “Por” 3. Clicar sobre um dos botões “>”, “<”, “=”, “>=”, “<=”
Saídas Esperadas	1. O resultado da pesquisa efetuada deve aparecer na tabela apresentada
Ref. ao protótipo	Ecrã “Lista de tabelas disponíveis”> comboBox “Pesquisar”> textBox “Por”> botões “>”, “<”, “=”, “>=”, “<=”
Resultado	Validado

Nº de teste	T22
Objetivo	Filtros
Condições de execução	1. Estar no ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados
Entradas	1. Escolher um campo do tipo não numérico na comboBox “Pesquisar”
Saídas Esperadas	1. Não aparecem (ou desaparecem) os botões de “>”, “<”, “=”, “>=”, “<=”, datePicker “Nova data” e a comboBox “De” e a “Até” e o botão “Ok”
Ref. ao protótipo	Ecrã “Lista de tabelas disponíveis”> comboBox “Pesquisar”
Resultado	Validado

Nº de teste	T23
Objetivo	Filtros
Condições de execução	1. Estar no ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados
Entradas	<ol style="list-style-type: none"> 1. Escolher um campo do tipo não numérico na comboBox “Pesquisar” 2. Colocar caracteres na textBox “Por”
Saídas Esperadas	1. O resultado da pesquisa efetuada deve aparecer na tabela apresentada
Ref. ao protótipo	Ecrã “Lista de tabelas disponíveis”> comboBox “Pesquisar”> textBox “Por”

Anexo B

Resultado	Validado
------------------	----------

Nº de teste	T24
Objetivo	Filtros
Condições de execução	1. Estar no ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados
Entradas	1. Escolher um campo do tipo DateTime só com hora na comboBox “Pesquisar”
Saídas Esperadas	1. Aparecem os botões de “>”, “<”, “=”, “>=”, “<=” e o dateTimePicker “Nova data”
Ref. ao protótipo	Ecrã “Lista de tabelas disponíveis”> comboBox “Pesquisar”
Resultado	Validado

Nº de teste	T25
Objetivo	Filtros
Condições de execução	1. Estar no ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados
Entradas	1. Escolher um campo do tipo DateTime só com hora na comboBox “Pesquisar” 2. Não escolher nenhum campo na comboBox “Horas” ou escrever caracteres na comboBox 3. Clicar sobre um dos botões “>”, “<”, “=”, “>=”, “<=”
Saídas Esperadas	1. Mensagem de erro “Selecione uma hora na comboBox!”
Ref. ao protótipo	Ecrã “Lista de tabelas disponíveis”> comboBox “Pesquisar”> ComboBox “Por”> botões “>”, “<”, “=”, “>=”, “<=”
Resultado	Validado

Nº de teste	T26
Objetivo	Filtros
Condições de execução	1. Estar no ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados
Entradas	1. Escolher um campo do tipo DateTime só com hora na comboBox “Pesquisar”

B.1 Plano de Testes

	<ol style="list-style-type: none"> Escolher uma hora na comboBox “Horas” Clicar sobre um dos botões “>”, “<”, “=”, “> =”, “< =” Não escolher uma nova data no dateTimePicker “Nova Data”
Saídas Esperadas	1. O resultado da pesquisa efetuada deve aparecer na pré-visualização da tabela unificada com a data por defeito do DateTimePicker (data de hoje)
Ref. ao protótipo	Ecrã “Lista de tabelas disponíveis”> comboBox “Pesquisar”> ComboBox “Horas”> botões “>”, “<”, “=”, “> =”, “< =”
Resultado	Validado

Nº de teste	T27
Objetivo	Filtros
Condições de execução	1. Estar no ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados
Entradas	<ol style="list-style-type: none"> Escolher um campo do tipo DateTime só com hora na comboBox “Pesquisar” Escolher uma hora na comboBox “Horas” Clicar sobre um dos botões “>”, “<”, “=”, “> =”, “< =” Escolher uma nova data no dateTimePicker “Nova Data”
Saídas Esperadas	1. O resultado da pesquisa efetuada deve aparecer na pré-visualização da tabela unificada com a nova data escolhida
Ref. ao protótipo	Ecrã “Lista de tabelas disponíveis”> comboBox “Pesquisar”> dateTimePicker “Por”> botões “>”, “<”, “=”, “> =”, “< =”> dateTimePicker “Nova data”
Resultado	Validado

Nº de teste	T28
Objetivo	Filtros
Condições de execução	1. Estar no ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados
Entradas	1. Escolher um campo do tipo DateTime com data na comboBox “Pesquisar”

Saídas Esperadas	<ol style="list-style-type: none"> 1. Não aparece (ou desaparece) o botão de datePicker “Nova data” 2. Não aparecem (ou desaparecem) os botões “>”, “<”, “=”, “> =”, “< =” 3. Aparece a comboBox “De”, a comboBox “Até” e o botão “Ok”
Ref. ao protótipo	Ecrã “Lista de tabelas disponíveis”> comboBox “Pesquisar”
Resultado	Validado

Nº de teste	T29
Objetivo	Filtros
Condições de execução	1. Estar no ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados
Entradas	<ol style="list-style-type: none"> 1. Escolher um campo do tipo DateTime com data e hora, na comboBox “Pesquisar” 2. Colocar caracteres (ou deixar em branco) as comboBox “De” e “Ate” 3. Carregar no botão “Ok”
Saídas Esperadas	1. Mensagem de erro: "Verifique se selecionou nas comboBox uma data mínima e uma máxima!"
Ref. ao protótipo	Ecrã “Lista de tabelas disponíveis”> comboBox “Pesquisar”> comboBox “De” e “Ate”> botão “Ok”
Resultado	Validado

Nº de teste	T30
Objetivo	Filtros
Condições de execução	1. Estar no ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados
Entradas	<ol style="list-style-type: none"> 1. Escolher um campo do tipo DateTime com data e hora, na comboBox “Pesquisar” 2. Escolher a data e hora nas comboBox “De” e “Ate” 3. Clicar sobre o botão “Ok”

Saídas Esperadas	1. O resultado da pesquisa efetuada deve aparecer na tabela apresentada
Ref. ao protótipo	Ecrã “Lista de tabelas disponíveis”> comboBox “Pesquisar”> comboBox “De” e “Ate”> botão “Ok”
Resultado	Validado

Nº de teste	T31
Objetivo	Filtros
Condições de execução	1. Estar no ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados
Entradas	1. Escolher o campo do tipo numérico correspondente ao ID_TAG, na comboBox “Pesquisar”
Saídas Esperadas	1. Aparecem os botões de “>”, “<”, “=”, “>=”, “<=”, mas apenas o botão “=” está ativo
Ref. ao protótipo	Ecrã “Lista de tabelas disponíveis”> comboBox “Pesquisar”
Resultado	Validado

Nº de teste	T32
Objetivo	Filtros
Condições de execução	1. Estar no ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados
Entradas	1. Escolher o campo do tipo numérico correspondente ao ID_TAG, na comboBox “Pesquisar” 2. Colocar um número na textBox “Por” 3. Clicar no botão “=”
Saídas Esperadas	1. O resultado da pesquisa efetuada deve aparecer na tabela apresentada 2. No campo ID_TAG aparece o número pelo qual pesquisamos.
Ref. ao protótipo	Ecrã “Lista de tabelas disponíveis”> comboBox “Pesquisar”> textBox “Por”> botão “=”
Resultado	Validado

Anexo B

Nº de teste	T33
Objetivo	Remover filtros
Condições de execução	1. Estar no ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados
Entradas	1. Ter efetuado algum filtro 2. Clicar no botão “Remover Filtros”
Saídas Esperadas	1. É repostos o original da tabela apresentada
Ref. ao protótipo	Ecrã “Lista de tabelas disponíveis”> botão “Remover Filtros”
Resultado	Validado

Nº de teste	T34
Objetivo	Associar dados
Condições de execução	1. Estar no ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados
Entradas	1. Não preencher a textBox “ID_TAG” ou não selecionar ou escrever a descrição pretendida na comboBox “DESC_TAG” 2. Clicar no botão “Ok”
Saídas Esperadas	1. Mensagem de erro: "Verifique se o ID_TAG ou DES_TAG estão vazios!"
Ref. ao protótipo	Ecrã “Lista de tabelas disponíveis”> textBox “ID_TAG”> comboBox “DESC_TAG”> botão “Ok”
Resultado	Validado

Nº de teste	T35
Objetivo	Associar dados
Condições de execução	1. Estar no ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados
Entradas	1. Ter visível uma tabela que tenha o campo correspondente ao ID_TAG 2. Ter aplicado um filtro ao ID_TAG por um número específico 3. Ter o ID_TAG visível na textBox “ID_TAG”

B.1 Plano de Testes

	<ol style="list-style-type: none"> 4. Selecionar a descrição pretendida na comboBox “DESC_TAG” 5. Clicar no botão “Ok”
Saídas Esperadas	1. Aparecem os campos para definir a unidade pretendida
Ref. ao protótipo	Ecrã “Lista de tabelas disponíveis”> textBox “ID_TAG”> comboBox “DESC_TAG”> botão “Ok”
Resultado	Validado

Nº de teste	T36
Objetivo	Associar dados
Condições de execução	1. Estar no ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados
Entradas	<ol style="list-style-type: none"> 1. Ter visível uma tabela que tenha o campo correspondente ao ID_TAG 2. Ter aplicado um filtro ao ID_TAG por um número específico 3. Ter o ID_TAG visível na textBox “ID_TAG” 4. Escrever a descrição pretendida na comboBox “DESC_TAG” 5. Clicar no botão “Ok”
Saídas Esperadas	<ol style="list-style-type: none"> 1. Aparece adicionado na comboBox “DESC_TAG” a descrição que foi escrita 2. Aparecem os campos para definir a unidade pretendida
Ref. ao protótipo	Ecrã “Lista de tabelas disponíveis”> textBox “ID_TAG”> comboBox “DESC_TAG”> botão “Ok”
Resultado	Validado

Nº de teste	T37
Objetivo	Associar dados
Condições de execução	1. Estar no ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados
Entradas	1. Ter visível uma tabela que tenha o campo correspondente à DESC_TAG

Anexo B

	2. Ter aplicado um filtro ao DESC_TAG por uma descrição específica
Saídas Esperadas	1. A textBox “ID_TAG”, a comboBox “DESC_TAG” e o botão “OK” ficam ativos
Ref. ao protótipo	Ecrã “Lista de tabelas disponíveis”> textBox “Por”
Resultado	Validado

Nº de teste	T38
Objetivo	Associar dados
Condições de execução	1. Estar no ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados
Entradas	1. Ter visível uma tabela que tenha o campo correspondente à DESC_TAG 2. Ter aplicado um filtro ao DESC_TAG por uma descrição específica 3. Preencher a textBox “ID_TAG” e selecionar a descrição pretendida na comboBox “DESC_TAG” 4. Clicar no botão “Ok”
Saídas Esperadas	1. Aparecem os campos para definir a unidade pretendida
Ref. ao protótipo	Ecrã “Lista de tabelas disponíveis”> textBox “ID_TAG”> comboBox “DESC_TAG”> botão “Ok”
Resultado	Validado

Nº de teste	T39
Objetivo	Associar dados
Condições de execução	1. Estar no ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados
Entradas	1. Ter visível uma tabela que tenha o campo correspondente à DESC_TAG 2. Ter aplicado um filtro ao DESC_TAG por uma descrição específica 3. Preencher a textBox “ID_TAG” e escrever a descrição pretendida na comboBox “DESC_TAG”

B.1 Plano de Testes

	4. Clicar no botão “Ok”
Saídas Esperadas	1. Aparece adicionado na comboBox “DESC_TAG” a descrição que foi escrita 2. Aparecem os campos para definir a unidade pretendida
Ref. ao protótipo	Ecrã “Lista de tabelas disponíveis”> textBox “ID_TAG”> comboBox “DESC_TAG”> botão “Ok”
Resultado	Validado

Nº de teste	T40
Objetivo	Associar dados
Condições de execução	1. Estar no ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados
Entradas	1. Ter visível uma tabela que não tenha o campo correspondente ao ID_TAG nem o campo correspondente à DESC_TAG
Saídas Esperadas	1. A textBox “ID_TAG”, a comboBox “DESC_TAG” e o botão “OK” ficam ativos
Ref. ao protótipo	Ecrã “Lista de tabelas disponíveis”
Resultado	Validado

Nº de teste	T41
Objetivo	Associar dados
Condições de execução	1. Estar no ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados
Entradas	1. Ter visível uma tabela que não tenha o campo correspondente ao ID_TAG nem o campo correspondente à DESC_TAG 2. Preencher a textBox “ID_TAG” e selecionar a descrição pretendida na comboBox “DESC_TAG” 3. Clicar no botão “Ok”
Saídas Esperadas	1. Aparecem os campos para definir a unidade pretendida
Ref. ao protótipo	Ecrã “Lista de tabelas disponíveis”> textBox “ID_TAG”> comboBox “DESC_TAG”> botão “Ok”

Resultado	Validado
------------------	----------

Nº de teste	T42
Objetivo	Associar dados
Condições de execução	1. Estar no ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados
Entradas	1. Ter visível uma tabela que não tenha o campo correspondente ao ID_TAG nem o campo correspondente à DESC_TAG 2. Preencher a textBox "ID_TAG" e escrever a descrição pretendida na comboBox "DESC_TAG" 3. Clicar no botão "Ok"
Saídas Esperadas	1. Aparece adicionado na comboBox "DESC_TAG" a descrição que foi escrita 2. Aparecem os campos para definir a unidade pretendida
Ref. ao protótipo	Ecrã "Lista de tabelas disponíveis"> textBox "ID_TAG"> comboBox "DESC_TAG"> botão "Ok"
Resultado	Validado

Nº de teste	T43
Objetivo	Associar dados
Condições de execução	1. Estar no ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados
Entradas	1. Ter ativo o campo de ID_TAG e o campo DESC_TAG 2. Preencher a textBox "ID_TAG" com caracteres e seleccionar ou escrever a descrição pretendida na comboBox "DESC_TAG" 3. Clicar no botão "Ok"
Saídas Esperadas	1. Mensagem de erro: "O ID_TAG deve ser um número!"
Ref. ao protótipo	Ecrã "Lista de tabelas disponíveis"> textBox "ID_TAG"> comboBox "DESC_TAG"> botão "Ok"
Resultado	Validado

Nº de teste	T44
Objetivo	Definir unidade
Condições de execução	1. Estar no ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados
Entradas	1. Ter visíveis as comboBox “Campo” e “Unidade” e ter o botão “Ok” desativado. 2. Preencher a comboBox “Campo” e “Unidade”
Saídas Esperadas	1. Botão “Ok” fica ativo
Ref. ao protótipo	Ecrã “Lista de tabelas disponíveis”> “Campo” e “Unidade”
Resultado	Validado

Nº de teste	T45
Objetivo	Definir unidade
Condições de execução	1. Estar no ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados
Entradas	1. Preencher a comboBox “Campo” e “Unidade” 2. Clicar no botão “Ok”
Saídas Esperadas	1. Botão “Ok” muda de cor 2. O botão “Selecionar” fica ativo
Ref. ao protótipo	Ecrã “Lista de tabelas disponíveis”> “Campo” e “Unidade”> botão “Ok”
Resultado	Validado

Nº de teste	T46
Objetivo	Converter dados
Condições de execução	1. Estar no ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados
Entradas	1. Ter definido a unidade na comboBox “Unidade” 2. Clicar no link “Converter”
Saídas Esperadas	1. O <i>link</i> “Converter” muda de cor 2. Aparecem as comboBox “Unidade Final” e “Decimal” ativas e o botão “Converter” desativado

Anexo B

Ref. ao protótipo	Ecrã “Lista de tabelas disponíveis”> comboBox “Campo” e “Unidade”> botão “Ok”> <i>link</i> “Converter”
Resultado	Validado

Nº de teste	T47
Objetivo	Converter dados
Condições de execução	1. Estar no ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados
Entradas	1. Não ter definido a unidade na comboBox “Unidade” 2. Clicar no <i>link</i> “Converter”
Saídas Esperadas	1. O link “Converter” muda de cor 2. Aparecem as comboBox “Unidade Final” e “Decimal” e o botão “Converter” desativados
Ref. ao protótipo	Ecrã “Lista de tabelas disponíveis”> <i>link</i> “Converter”
Resultado	Validado

Nº de teste	T48
Objetivo	Converter dados
Condições de execução	1. Estar no ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados 2. Ter definida a unidade inicial
Entradas	1. Clicar no link “Converter” 2. Preencher a comboBox “Unidade final” com a mesma ou com unidade incompatíveis relativamente a unidade inicial 3. Preencher a comboBox “Decimais” 4. Clicar no botão “Converter”
Saídas Esperadas	1. Não acontece nada
Ref. ao protótipo	Ecrã “Lista de tabelas disponíveis”> link “Converter”> comboBox “Unidade final”> comboBox “Decimais”> botão “Converter”
Resultado	Validado

Nº de teste	T49
--------------------	------------

Objetivo	Converter dados
Condições de execução	<ol style="list-style-type: none"> 1. Estar no ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados 2. Ter definida a unidade inicial
Entradas	<ol style="list-style-type: none"> 1. Clicar no link “Converter” 2. Preencher a comboBox “Unidade final” com uma unidade diferente da inicial 3. Preencher a comboBox “Decimais” 4. Clicar no botão “Converter”
Saídas Esperadas	1. O sistema faz a conversão pretendida e mostra o resultado na coluna respetiva da tabela apresentada
Ref. ao protótipo	Ecrã “Lista de tabelas disponíveis”> link “Converter”> comboBox “Unidade final”> comboBox “Decimais”> botão “Converter”
Resultado	Validado

Nº de teste	T50
Objetivo	Selecionar dados da tabela apresentada
Condições de execução	<ol style="list-style-type: none"> 1. Estar no ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados 2. Ter associado o ID_TAG e DESC_TAG 3. Ter definido uma unidade 4. Ter ou não feito conversão dos dados
Entradas	1. Clicar no botão “Selecionar”
Saídas Esperadas	<ol style="list-style-type: none"> 1. O sistema mostra os dados selecionados na tabela unificada 2. Os botões “Limpar” e “Confirmar” ficam ativos
Ref. ao protótipo	Ecrã “Lista de tabelas disponíveis”> botão “Selecionar”
Resultado	Validado

Nº de teste	T51
Objetivo	1. Repetir a seleção de dados da tabela apresentada

Condições de execução	<ol style="list-style-type: none"> 1. Estar no ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados 2. Ter associado o ID_TAG e DESC_TAG 3. Ter definido uma unidade de um campo diferente do anterior 4. Ter ou não feito conversão dos dados
Entradas	1. Clicar no botão “Selecionar”
Saídas Esperadas	Mensagem de Erro: "Antes de selecionar os dados limpe a seleção efetuada anteriormente!"
Ref. ao protótipo	Ecrã “Lista de tabelas disponíveis”> botão “Selecionar”
Resultado	Validado

Nº de teste	T52
Objetivo	Limpar os dados da tabela unificada
Condições de execução	<ol style="list-style-type: none"> 1. Estar no ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados 2. Ter dados visíveis na tabela unificada
Entradas	1. Clicar no botão “Limpar”
Saídas Esperadas	<ol style="list-style-type: none"> 1. O sistema limpa toda informação a tabela unificada 2. O botão “Confirmar” fica desativado 3. O sistema limpa a seleção efetuada à tabela apresentada
Ref. ao protótipo	Ecrã “Lista de tabelas disponíveis”> visualizar os dados da tabela unificada> botão “Limpar”
Resultado	Validado

Nº de teste	T53
Objetivo	Confirmar a seleção de dados efetuados
Condições de execução	<ol style="list-style-type: none"> 1. Estar no ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados 2. Ter dados na tabela unificada
Entradas	1. Clicar no botão “Confirmar”
Saídas Esperadas	1. Avança para onde se efetuam os processamentos

Ref. ao protótipo	Ecrã “Lista de tabelas disponíveis”> visualizar os dados da tabela unificada> botão “Confirmar”
Resultado	Validado

Nº de teste	T54
Objetivo	1. Efetuar cálculos estatísticos
Condições de execução	Estar no ecrã onde se efetuam os cálculos estatísticos
Entradas	1. Não colocar nada em uma ou ambas as comboBox “Calcular” e “Em” 2. Clicar no botão “Ok”
Saídas Esperadas	1. Não acontece nada
Ref. ao protótipo	Ecrã “Processar dados”> botão “Ok”
Resultado	Validado

Nº de teste	T55
Objetivo	1. Efetuar cálculos estatísticos
Condições de execução	Estar no ecrã onde se efetuam os cálculos estatísticos
Entradas	1. Escolher um campo na comboBox “Calcular” 2. Escolher um campo na comboBox “Em” 3. Clicar no botão “Ok”
Saídas Esperadas	1. Aparece na textBox “Resultado” o resultado obtido do cálculo efetuado 2. Aparece na listBox “Todos os cálculos efetuados” o cálculo agora efetuado, a seguir aos já lá existentes
Ref. ao protótipo	Ecrã “Processar dados”> comboBox “Calcular”> comboBox “Em”> botão “Ok”
Resultado	Validado

Nº de teste	T56
Objetivo	1. Efetuar cálculos estatísticos
Condições de execução	Estar no ecrã onde se efetuam os cálculos estatísticos
Entradas	1. Escolher na comboBox “Calcular” um cálculo já efetuado e adicionado à listBox “Todos os cálculos efetuados”

Anexo B

	<ol style="list-style-type: none"> Escolher na comboBox “Em” o mesmo campo sobre a qual se efetuou o cálculo Clicar no botão “Ok”
Saídas Esperadas	1. Não acontece nada
Ref. ao protótipo	Ecrã “Processar dados”> comboBox “Calcular”> comboBox “Em”> botão “Ok”
Resultado	Validado

Nº de teste	T57
Objetivo	1. Efetuar cálculos estatísticos
Condições de execução	Estar no ecrã onde se efetuam os cálculos estatísticos
Entradas	<ol style="list-style-type: none"> Escrever caracteres na comboBox “Calcular” e/ou na comboBox “Em” Clicar no botão “Ok”
Saídas Esperadas	1. Mensagem de erro: "Selecionar o campo pretendido nas comboBox!"
Ref. ao protótipo	Ecrã “Processar dados”> comboBox “Calcular” e “Em”> botão “Ok”
Resultado	Validado

Nº de teste	T58
Objetivo	1. Limpar os resultados dos cálculos estatísticos
Condições de execução	Estar no ecrã onde se efetuam os cálculos estatísticos
Entradas	1. Clicar no botão “Limpar Cálculos”
Saídas Esperadas	1. Os cálculos presentes na listView “Todos os cálculos efetuados” e textBox “Resultado” são limpos.
Ref. ao protótipo	Ecrã “Processar dados”> botão “Limpar cálculos”
Resultado	Validado

Nº de teste	T59
Objetivo	Voltar ao ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados
Condições de execução	1. Estar no ecrã onde se efetuam os cálculos estatísticos

Entradas	1. Clicar no botão “Voltar”
Saídas Esperadas	1. O sistema retorna ao ecrã anterior e mantém a tabela unificada preenchida
Ref. ao protótipo	Ecrã “Processar dados” > botão “Voltar”
Resultado	Validado

Nº de teste	T60
Objetivo	Guardar dados
Condições de execução	1. Estar no ecrã onde se efetuam os cálculos estatísticos
Entradas	1. Não selecionar como se quer guardar os dados na combo-Box “Guardar em” 2. Clicar no botão “Guardar”
Saídas Esperadas	1. Mensagem de erro: "Escolha como quer guardar os dados!"
Ref. ao protótipo	Ecrã “Processar dados” > botão “Guardar”
Resultado	Validado

Nº de teste	T61
Objetivo	Guardar dados
Condições de execução	1. Estar no ecrã onde se efetuam os cálculos estatísticos
Entradas	1. Selecionar, na comboBox “Guardar em”, que se quer guardar os dados em formato por cima de um ficheiro já existente. 2. Ter o ficheiro onde se está a tentar guardar aberto
Saídas Esperadas	1. Mensagem de erro: "Antes de guardar, verifique se o ficheiro destino está aberto!"
Ref. ao protótipo	Ecrã “Processar dados” > botão “Guardar”
Resultado	Validado

Nº de teste	T62
Objetivo	Guardar dados
Condições de execução	1. Estar no ecrã onde se efetuam os cálculos estatísticos
Entradas	1. Selecionar como se quer guardar os dados na comboBox “Guardar em”

Anexo B

	2. Clicar no botão “Guardar”
Saídas Esperadas	1. O sistema guarda os dados da tabela unificada e dos cálculos estatísticos 2. Aparece o <i>pop-up</i> que informa que os dados foram registados com sucesso
Ref. ao protótipo	Ecrã “Processar dados”> comboBox “Guardar em”> botão “Guardar”
Resultado	Validado

Nº de teste	T63
Objetivo	Painel informativo
Condições de execução	1. Estar no <i>pop-up</i> Informativo “Dados registados com sucesso”
Entradas	1. Clicar no botão “Ok”
Saídas Esperadas	1. O sistema retorna ao ecrã “Processar dados” 2. Os cálculos presentes na listView “Todos os cálculos efetuados” e textBox “Resultado” são limpos
Ref. ao protótipo	Ecrã “Processar dados”> botão “Ok”
Resultado	Validado

Nº de teste	T64
Objetivo	Saber se o utilizador pretende encerrar o sistema a partir do ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados ou do ecrã “Processar dados”
Condições de execução	1. Estar no ecrã onde se efetuam a seleção dos dados ou no ecrã “Processar dados”
Entradas	1. Clicar sobre o botão “Encerrar o sistema”
Saídas Esperadas	1. Aparece o <i>pop-up</i> a perguntar se “deseja encerrar o sistema”
Ref. ao protótipo	Ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados ou ecrã “Processar dados”> botão “Encerrar o sistema”
Resultado	Validado

Nº de teste	T65
Objetivo	Saber se o utilizador pretende encerrar o sistema a partir do ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados ou do ecrã “Processar dados”
Condições de execução	1. Estar no <i>pop-up</i> onde se pergunta se deseja encerrar o sistema?
Entradas	1. Clicar no botão “Não”
Saídas Esperadas	1. Retorna ao ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados ou ao ecrã “Processar dados”
Ref. ao protótipo	Ecrã “Deseja encerrar o sistema?”> botão “Não”
Resultado	Validado

Nº de teste	T66
Objetivo	Saber se o utilizador pretende encerrar o sistema a partir do ecrã onde podem ser visualizadas as tabelas disponíveis para seleção de dados ou do ecrã “Processar dados”
Condições de execução	1. Estar no <i>pop-up</i> onde se pergunta se deseja encerrar o sistema?
Entradas	1. Clicar no botão “Sim”
Saídas Esperadas	1. O sistema termina
Ref. ao protótipo	Ecrã “Deseja encerrar o sistema?”> botão “Sim”
Resultado	Validado

Anexo C

C.1. Tabelas Originais e Unificadas

ID_ cozinha- RefSocial	Data/Hora	Temp Ar (°F)	CO2 (ppm)	Hum Rel(%)	Fungos (Max 1500)	Bact Tot (Max 1500)	CONS ENER (\$/kwj)
1	25/10/12 0:00	84,67	14515,33	28,25	174,56	167,00	9,53
2	25/10/12 1:00	82,85	20495,23	37,33	479,28	165,00	5,45
3	25/10/12 2:00	90,12	3690,03	36,32	328,93	267,00	10,89
4	25/10/12 3:00	90,12	2680,87	25,23	301,69	205,00	9,53
5	25/10/12 4:00	93,76	29340,42	34,31	163,46	536,00	4,09

Tabela C.2: Tabela original da cozinha do refeitório social

ID_ Cozinha BarPriv	HORA	Temp Ar (°C)	CO2 (Max 3%)	Hum Rel (%)	Fungos Int (Max 1500)	Bact Int (Max 1500)	CONS ENER (€/kwh)	Vel Ar (Km/h)
1	0:00:00	20,00	1,00	52,47	365,26	378,00	1,01	0,18
2	1:00:00	17,00	0,00	39,35	24,22	153,00	3,03	1,21
3	2:00:00	19,00	3,00	29,26	40,36	246,00	7,06	1,08
4	3:00:00	21,00	2,00	35,32	10,09	119,00	7,06	0,52
5	4:00:00	21,00	0,00	29,26	14,13	225,00	3,03	1,40

Tabela C.4: Tabela original da cozinha do bar privado

Id_ SalaA2	Data	Temp Ar (°F)	TMR (°F)	Vel Ar (m/s)	Cons Energ (€/kwh)	CO2 (%)	Hum Rel (%)
1	quinta-feira, novembro 01, 2012	77,414	77,405	1,04	50,45	0,997	33,297
2	sexta-feira, novembro 02, 2012	70,142	79,221	0,86	19,171	2,354	36,324
3	sábado, novembro 03, 2012	71,96	75,589	0,16	69,621	0,652	28,252
4	domingo, novembro 04, 2012	68,324	75,589	0,33	36,324	1,390	33,297
5	segunda-feira, novembro 05, 2012	71,96	75,589	0,7	46,414	2,262	32,288

Tabela C.6: Tabela original da SalaA2

id_ salaB4	data/hora	temp ar (°K)	hum rel (%)	tmr (°K)	vel ar (km/h)	cons ener (€/kwh)	Diox carbono (ppm)
1	19/11/12 0:00	291,16	15,14	301,25	0,13	46,41	23266,76
2	19/11/12 12:00	287,13	17,15	298,23	0,31	53,48	5525,51
3	20/11/12 0:00	293,18	15,14	300,24	0,23	59,53	26203,26
4	20/11/12 12:00	293,18	14,13	298,23	0,15	49,44	1026,91
5	21/11/12 0:00	292,17	19,17	301,25	0,35	44,40	3594,29

Tabela C.8: Tabela original da SalaB4

Id_ Biblioteca	Data	Temp Ar (°C)	TMR (°C)	Hum Rel (%)	Vel Ar (m/s)	Cons Ener (€/kwh)	CO2 (ppm)
1	1/nov	28,25	27,24	23,21	0,67	59,53	23805,38
2	2/nov	28,25	27,24	20,18	0,89	43,39	23237,28
4	4/nov	29,26	28,25	27,24	0,09	42,38	26615,33
5	5/nov	28,25	28,25	24,22	1,29	50,45	17208,10

Tabela C.10: Tabela original da biblioteca

C.1 Tabelas Originais e Unificadas

idSecr	DataHora	Temp Ar (°C)	TMR (°C)	Hum Rel (%)	Vel Ar (Km/h)	Cons Ener (\$/kwh)	CO2(%)
1	2012-11-19 00:00:00	28.25	31.28	19.17	0.83	32.29	1.73
2	2012-11-19 06:00:00	31.28	29.26	12.11	0.48	34.30	0.379
3	2012-11-19 12:00:00	28.25	32.29	14.13	0.40	48.43	0.38
4	2012-11-19 18:00:00	27.24	29.26	11.10	0.83	41.37	0.00
5	2012-11-20 00:00:00	27.24	31.28	17.15	1.23	39.35	1.65

Tabela C.12: Tabela original da secretaria-geral

Chave Unica	Ref_ Sistema	ID	Data/Hora	ID_ TAG	DESC_ TAG	Valor	Unidade
1	Cozinha[Refeitorio Social.xlsx]	9	25/10/2012 08:00:00	1	Consumo de Energia	6,054	Euro/KiloWatt- Hora
2	Cozinha[Refeitorio Social.xlsx]	10	25/10/2012 09:00:00	1	Consumo de Energia	9,081	Euro/KiloWatt- Hora
3	Cozinha[Refeitorio Social.xlsx]	11	25/10/2012 10:00:00	1	Consumo de Energia	3,027	Euro/KiloWatt- Hora
4	Cozinha[Refeitorio Social.xlsx]	12	25/10/2012 11:00:00	1	Consumo de Energia	5,045	Euro/KiloWatt- Hora
5	Cozinha[Refeitorio Social.xlsx]	13	25/10/2012 12:00:00	1	Consumo de Energia	9,081	Euro/KiloWatt- Hora
6	Cozinha[Refeitorio Social.xlsx]	14	25/10/2012 13:00:00	1	Consumo de Energia	3,027	Euro/KiloWatt- Hora
7	Cozinha[Refeitorio Social.xlsx]	15	25/10/2012 14:00:00	1	Consumo de Energia	6,054	Euro/KiloWatt- Hora
8	Cozinha[Refeitorio Social.xlsx]	16	25/10/2012 15:00:00	1	Consumo de Energia	4,036	Euro/KiloWatt- Hora

9	Cozinha[Refeitório Social.xlsx]	17	25/10/2012 16:00:00	1	Consumo de Energia	9,081	Euro/KiloWatt-Hora
10	Cozinha[Refeitório Social.xlsx]	18	25/10/2012 17:00:00	1	Consumo de Energia	9,081	Euro/KiloWatt-Hora
11	cozinha[BarPrivado .accdb]	9	25/10/2012 08:00:00	2	Consumo de Energia	7,06	Euro/KiloWatt-Hora
12	cozinha[BarPrivado .accdb]	10	25/10/2012 09:00:00	2	Consumo de Energia	9,08	Euro/KiloWatt-Hora
13	cozinha[BarPrivado .accdb]	11	25/10/2012 10:00:00	2	Consumo de Energia	4,04	Euro/KiloWatt-Hora
14	cozinha[BarPrivado .accdb]	12	25/10/2012 11:00:00	2	Consumo de Energia	9,08	Euro/KiloWatt-Hora
15	cozinha[BarPrivado .accdb]	13	25/10/2012 12:00:00	2	Consumo de Energia	4,04	Euro/KiloWatt-Hora
16	cozinha[BarPrivado .accdb]	14	25/10/2012 13:00:00	2	Consumo de Energia	9,08	Euro/KiloWatt-Hora
17	cozinha[BarPrivado .accdb]	15	25/10/2012 14:00:00	2	Consumo de Energia	7,06	Euro/KiloWatt-Hora
18	cozinha[BarPrivado .accdb]	16	25/10/2012 15:00:00	2	Consumo de Energia	9,08	Euro/KiloWatt-Hora
19	cozinha[BarPrivado .accdb]	17	25/10/2012 16:00:00	2	Consumo de Energia	6,05	Euro/KiloWatt-Hora
20	cozinha[BarPrivado .accdb]	18	25/10/2012 17:00:00	2	Consumo de Energia	11,1	Euro/KiloWatt-Hora

Tabela C.14: Tabela unificada sobre o consumo de energia durante um dia no período de aulas

Chave Unica	Ref_Sistema	ID	Data/Hora	ID_TAG	DESC_TAG	Valor	Unidade
1	Cozinha[Refeitório Social.xlsx]	9	26/08/2013 08:00:00	1	Consumo de Energia	3,027	Euro/KiloWatt-Hora
2	Cozinha[Refeitório Social.xlsx]	10	26/08/2013 09:00:00	1	Consumo de Energia	2,018	Euro/KiloWatt-Hora

C.1 Tabelas Originais e Unificadas

3	Cozinha[Refeitório Social.xlsx]	11	26/08/2013 10:00:00	1	Consumo de Energia	3,027	Euro/KiloWatt-Hora
4	Cozinha[Refeitório Social.xlsx]	12	26/08/2013 11:00:00	1	Consumo de Energia	3,027	Euro/KiloWatt-Hora
5	Cozinha[Refeitório Social.xlsx]	13	26/08/2013 12:00:00	1	Consumo de Energia	2,018	Euro/KiloWatt-Hora
6	Cozinha[Refeitório Social.xlsx]	14	26/08/2013 13:00:00	1	Consumo de Energia	1,009	Euro/KiloWatt-Hora
7	Cozinha[Refeitório Social.xlsx]	15	26/08/2013 14:00:00	1	Consumo de Energia	2,018	Euro/KiloWatt-Hora
8	Cozinha[Refeitório Social.xlsx]	16	26/08/2013 15:00:00	1	Consumo de Energia	4,036	Euro/KiloWatt-Hora
9	Cozinha[Refeitório Social.xlsx]	17	26/08/2013 16:00:00	1	Consumo de Energia	4,036	Euro/KiloWatt-Hora
10	Cozinha[Refeitório Social.xlsx]	18	26/08/2013 17:00:00	1	Consumo de Energia	4,036	Euro/KiloWatt-Hora
11	Cozinha[BarPrivado .accdb]	9	26/08/2013 08:00:00	2	Consumo de Energia	4,04	Euro/KiloWatt-Hora
12	Cozinha[BarPrivado .accdb]	10	26/08/2013 09:00:00	2	Consumo de Energia	7,06	Euro/KiloWatt-Hora
13	Cozinha[BarPrivado .accdb]	11	26/08/2013 10:00:00	2	Consumo de Energia	3,03	Euro/KiloWatt-Hora
14	Cozinha[BarPrivado .accdb]	12	26/08/2013 11:00:00	2	Consumo de Energia	4,04	Euro/KiloWatt-Hora
15	Cozinha[BarPrivado .accdb]	13	26/08/2013 12:00:00	2	Consumo de Energia	1,01	Euro/KiloWatt-Hora
16	Cozinha[BarPrivado .accdb]	14	26/08/2013 13:00:00	2	Consumo de Energia	3,03	Euro/KiloWatt-Hora
17	Cozinha[BarPrivado .accdb]	15	26/08/2013 14:00:00	2	Consumo de Energia	1,01	Euro/KiloWatt-Hora
18	Cozinha[BarPrivado .accdb]	16	26/08/2013 15:00:00	2	Consumo de Energia	6,05	Euro/KiloWatt-Hora

Anexo C

19	Cozinha[BarPrivado .accdb]	17	26/08/2013 16:00:00	2	Consumo de Energia	2,02	Euro/KiloWatt- Hora
20	Cozinha[BarPrivado .accdb]	18	26/08/2013 17:00:00	2	Consumo de Energia	7,06	Euro/KiloWatt- Hora

Tabela C.16: Tabela unificada sobre o consumo de energia durante um dia no período de férias

Chave Unica	Ref_Sistema	ID	Data/Hora	ID_TAG	DESC_TAG	Valor	Unidade
1	SalaA2[BlocoA.accdb]	19	19/11/2012 00:00:00	3	Temperatura Ar	12,11	Graus Celsius
2	SalaA2[BlocoA.accdb]	20	20/11/2012 00:00:00	3	Temperatura Ar	9,08	Graus Celsius
3	SalaA2[BlocoA.accdb]	21	21/11/2012 00:00:00	3	Temperatura Ar	15,14	Graus Celsius
4	SalaA2[BlocoA.accdb]	22	22/11/2012 00:00:00	3	Temperatura Ar	20,18	Graus Celsius
5	SalaA2[BlocoA.accdb]	23	23/11/2012 00:00:00	3	Temperatura Ar	13,12	Graus Celsius
6	SalaA2[BlocoA.accdb]	24	24/11/2012 00:00:00	3	Temperatura Ar	25,23	Graus Celsius
7	SalaA2[BlocoA.accdb]	25	25/11/2012 00:00:00	3	Temperatura Ar	15,14	Graus Celsius
8	SalaB4[BlocoB.xlsx]	1	19/11/2012 00:00:00	4	Temperatura Ar	15,99	Graus Celsius
9	SalaB4[BlocoB.xlsx]	3	20/11/2012 00:00:00	4	Temperatura Ar	20,03	Graus Celsius
10	SalaB4[BlocoB.xlsx]	5	21/11/2012 00:00:00	4	Temperatura Ar	17,50	Graus Celsius
11	SalaB4[BlocoB.xlsx]	7	22/11/2012 00:00:00	4	Temperatura Ar	17,00	Graus Celsius
12	SalaB4[BlocoB.xlsx]	9	23/11/2012 00:00:00	4	Temperatura Ar	19,53	Graus Celsius

C.1 Tabelas Originais e Unificadas

13	SalaB4[BlocoB.xlsx]	11	24/11/2012 00:00:00	4	Temperatura Ar	15,99	Graus Celsius
14	SalaB4[BlocoB.xlsx]	13	25/11/2012 00:00:00	4	Temperatura Ar	15,99	Graus Celsius
15	Biblioteca[EDDPiso1.xlsx]	19	19/11/2012 00:00:00	5	Temperatura Ar	11,10	Graus Celsius
16	Biblioteca[EDDPiso1.xlsx]	20	20/11/2012 00:00:00	5	Temperatura Ar	18,16	Graus Celsius
17	Biblioteca[EDDPiso1.xlsx]	21	21/11/2012 00:00:00	5	Temperatura Ar	20,18	Graus Celsius
18	Biblioteca[EDDPiso1.xlsx]	22	22/11/2012 00:00:00	5	Temperatura Ar	18,16	Graus Celsius
19	Biblioteca[EDDPiso1.xlsx]	23	23/11/2012 00:00:00	5	Temperatura Ar	13,12	Graus Celsius
20	Biblioteca[EDDPiso1.xlsx]	24	24/11/2012 00:00:00	5	Temperatura Ar	17,15	Graus Celsius
21	Biblioteca[EDDPiso1.xlsx]	25	25/11/2012 00:00:00	5	Temperatura Ar	21,19	Graus Celsius
22	secretaria[edc]	1	19/11/2012 00:00:00	6	Temperatura Ar	28,76	Graus Celsius
23	secretaria[edc]	5	20/11/2012 00:00:00	6	Temperatura Ar	28,76	Graus Celsius
24	secretaria[edc]	9	21/11/2012 00:00:00	6	Temperatura Ar	31,02	Graus Celsius
25	secretaria[edc]	13	22/11/2012 00:00:00	6	Temperatura Ar	28,25	Graus Celsius
26	secretaria[edc]	17	23/11/2012 00:00:00	6	Temperatura Ar	30,27	Graus Celsius
27	secretaria[edc]	21	24/11/2012 00:00:00	6	Temperatura Ar	29,51	Graus Celsius
28	secretaria[edc]	25	25/11/2012 00:00:00	6	Temperatura Ar	22,20	Graus Celsius

Anexo C

Tabela C.18: Tabela unificada da temperatura do ar nas salas (de aulas, bibliotecas e secretaria-geral) durante uma semana no período de aulas

Chave Unica	Ref_Sistema	ID	Data/Hora	ID_TAG	DESC_TAG	Valor	Unidade
1	SalaA2[BlocoA.accdb]	53	22/07/2013 00:00:00	3	Temperatura Ar	21,19	Graus Celsius
2	SalaA2[BlocoA.accdb]	54	23/07/2013 00:00:00	3	Temperatura Ar	22,20	Graus Celsius
3	SalaA2[BlocoA.accdb]	55	24/07/2013 00:00:00	3	Temperatura Ar	19,17	Graus Celsius
4	SalaA2[BlocoA.accdb]	56	25/07/2013 00:00:00	3	Temperatura Ar	19,17	Graus Celsius
5	SalaA2[BlocoA.accdb]	57	26/07/2013 00:00:00	3	Temperatura Ar	20,18	Graus Celsius
6	SalaA2[BlocoA.accdb]	58	27/07/2013 00:00:00	3	Temperatura Ar	20,18	Graus Celsius
7	SalaA2[BlocoA.accdb]	59	28/07/2013 00:00:00	3	Temperatura Ar	21,19	Graus Celsius
8	SalaB4[BlocoB.xlsx]	15	22/07/2013 00:00:00	4	Temperatura Ar	20,03	Graus Celsius
9	SalaB4[BlocoB.xlsx]	17	23/07/2013 00:00:00	4	Temperatura Ar	22,05	Graus Celsius
10	SalaB4[BlocoB.xlsx]	19	24/07/2013 00:00:00	4	Temperatura Ar	23,56	Graus Celsius
11	SalaB4[BlocoB.xlsx]	21	25/07/2013 00:00:00	4	Temperatura Ar	20,53	Graus Celsius
12	SalaB4[BlocoB.xlsx]	23	26/07/2013 00:00:00	4	Temperatura Ar	21,04	Graus Celsius
13	SalaB4[BlocoB.xlsx]	25	27/07/2013 00:00:00	4	Temperatura Ar	26,09	Graus Celsius
14	SalaB4[BlocoB.xlsx]	27	28/07/2013 00:00:00	4	Temperatura Ar	23,06	Graus Celsius

15	Biblioteca[EDDPiso1.xlsx]	52	22/07/2013 00:00:00	5	Temperatura Ar	29,26	Graus Celsius
16	Biblioteca[EDDPiso1.xlsx]	53	23/07/2013 00:00:00	5	Temperatura Ar	29,26	Graus Celsius
17	Biblioteca[EDDPiso1.xlsx]	54	24/07/2013 00:00:00	5	Temperatura Ar	24,22	Graus Celsius
18	Biblioteca[EDDPiso1.xlsx]	55	25/07/2013 00:00:00	5	Temperatura Ar	25,23	Graus Celsius
19	Biblioteca[EDDPiso1.xlsx]	56	26/07/2013 00:00:00	5	Temperatura Ar	28,25	Graus Celsius
20	Biblioteca[EDDPiso1.xlsx]	57	27/07/2013 00:00:00	5	Temperatura Ar	24,22	Graus Celsius
21	Biblioteca[EDDPiso1.xlsx]	58	28/07/2013 00:00:00	5	Temperatura Ar	24,22	Graus Celsius
22	secretaria[edc]	29	22/07/2013 00:00:00	6	Temperatura Ar	22,71	Graus Celsius
23	secretaria[edc]	33	23/07/2013 00:00:00	6	Temperatura Ar	19,42	Graus Celsius
24	secretaria[edc]	37	24/07/2013 00:00:00	6	Temperatura Ar	20,43	Graus Celsius
25	secretaria[edc]	41	25/07/2013 00:00:00	6	Temperatura Ar	20,43	Graus Celsius
26	secretaria[edc]	45	26/07/2013 00:00:00	6	Temperatura Ar	21,70	Graus Celsius
27	secretaria[edc]	49	27/07/2013 00:00:00	6	Temperatura Ar	20,68	Graus Celsius
28	secretaria[edc]	53	28/07/2013 00:00:00	6	Temperatura Ar	19,17	Graus Celsius

Tabela C.20: Tabela unificada da temperatura do ar nas salas (de aulas, bibliotecas e secretaria-geral) durante uma semana no período de férias