



NOVA

IMS

Information
Management
School

MGI

Mestrado em Gestão de Informação

Master Program in Information Management

Construtor Automático de *Queries* SQL a partir de um Modelo Relacional

Afonso Antunes Bonifácio Vítor

Trabalho de Projeto apresentado como requisito parcial para
obtenção do grau de Mestre em Gestão de Informação

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

CONSTRUTOR AUTOMÁTICO DE *QUERIES* A PARTIR DE UM MODELO RELACIONAL

por

Afonso Antunes Bonifácio Vítor

Trabalho de Projeto apresentado como requisito parcial para a obtenção do grau de Mestre em
Gestão de Informação, Especialização em Gestão dos Sistemas e Tecnologias de Informação

Orientador/Coorientador: Vítor Manuel Pereira Duarte dos Santos, PhD

Novembro 2017

AGRADECIMENTOS

Ao Professor Vítor Santos, orientador desta dissertação, pela oportunidade de trabalhar neste projeto e pela suas críticas e sugestões que fez durante todo este trabalho.

À minha família, em especial aos meus pais, cujo suporte ajudou à conclusão deste projeto, para além de todo o apoio dado durante o meu percurso académico.

RESUMO

O presente trabalho consistiu em criar uma ferramenta desenvolvida em C# que, ligada a um qualquer modelo relacional dado, construa automaticamente e disponibilize as *queries* em linguagem SQL corretamente escritas aos utilizadores de acordo com o que estes procuram na base de dados, para além do seu equivalente em álgebra relacional. Adicionalmente, a ferramenta também pode gerar *queries* em linguagem LINQ.

Foram analisadas as matérias teóricas de suporte à geração de *queries* da linguagem SQL, desenhadas as arquiteturas de *software* e construído um protótipo. Finalmente, o protótipo foi sujeito à avaliação por diferentes utilizadores tendo passado com distinção nos testes.

PALAVRAS-CHAVE

Construtor de *Queries*, SQL, Modelo Relacional, Álgebra Relacional, LINQ

ABSTRACT

The present work consists in creating a tool developed in C# that linked to a given relational model, automatically build and make available SQL queries correctly written to the users according to what they look in a database, besides its equivalent in relational algebra. Additionally, it will also be possible for the tool to create the expression in LINQ language.

The theoretical material that supports the generation of queries in SQL language was analyzed, the software architectures were designed, and a prototype was build. Finally, the prototype was submitted to an evaluation by different users having passed with distinction throughout the tests.

KEYWORDS

Query Builder; SQL; Relational Model; Relational Algebra; LINQ

PUBLICAÇÕES RESULTANTES DA INVESTIGAÇÃO

Publicação de Poster na Conferência da Associação Portuguesa de Sistemas de Informação 2016 - CAPSI'2016 (Vitor & Santos, 2016)

ÍNDICE

1. Introdução	1
1.1. Contextualização	1
1.2. Justificação do Projeto	1
1.3. Objetivos do Projeto.....	2
1.3.1. Objetivo Principal	2
1.3.2. Objetivos Específicos	2
2. Plano de Trabalho.....	3
3. Enquadramento Teórico.....	6
3.1. Linguagem SQL	6
3.2. Álgebra Relacional	8
3.3. Linguagem LINQ.....	8
3.4. Ferramentas de Apoio ao Ensino de Informática.....	9
4. Análise e Desenho	10
4.1. Análise de Requisitos.....	10
4.1.1. Funcionais.....	10
4.1.2. Não Funcionais	10
4.2. Arquitetura de <i>Software</i>	11
4.2.1. Diagrama de <i>Use Case</i>	11
4.2.2. Diagrama de Atividades.....	12
4.2.3. Diagrama de Classes.....	13
4.3. Interface	14
4.4. Simulação de Funcionamento	15
5. Implementação do Protótipo	16
5.1. Arquitetura de <i>Software</i>	16
5.2. Arquitetura de Sistemas	17
5.3. Interfaces e Usabilidade	18
5.3.1. <i>Upload</i> do Ficheiro do Modelo Relacional	18
5.3.2. Geração de <i>Queries</i>	20
5.3.3. Descrição da Ferramenta	23
5.4. Testes.....	24
5.4.1. Testes Unitários	25
5.4.2. Testes de Integração	28
5.4.3. Testes de Sistema	29

5.4.4. Testes de Aceitação	36
6. Conclusões	38
6.1. Síntese do Trabalho Desenvolvido	38
6.2. Recomendações para Trabalhos Futuros	38
Bibliografia	39
Anexos	41

ÍNDICE DE FIGURAS

Figura 2.1 - Modelo do projeto (Santos, 2012)	4
Figura 4.1 - Diagrama de <i>Use Case</i>	11
Figura 4.2 - Diagrama de Atividades	12
Figura 4.3 - Diagrama de Classes.....	13
Figura 4.4 - Esboço da <i>interface</i> da ferramenta QG4B	14
Figura 5.1 - Arquitetura de <i>Software</i>	16
Figura 5.2 - Arquitetura de Sistemas.....	17
Figura 5.3 - Janela de <i>upload</i> do ficheiro	18
Figura 5.4 - Janela de escolha do ficheiro do modelo relacional.....	19
Figura 5.5 - Erro de estrutura incorreta do ficheiro inserido.....	19
Figura 5.6 - Janela para gerar a <i>query</i> pretendida	20
Figura 5.7 - Sinalização de erro numa tentativa de gerar uma <i>query</i>	21
Figura 5.8 - Exemplo de criação de uma <i>query</i>	22
Figura 5.9 - Janela de descrição da ferramenta	23

ÍNDICE DE TABELAS

Tabela 2.1 - Descrição das fases do modelo do projeto (Santos, 2012)	3
Tabela 2.2 - Cronograma do planeamento do projeto	5
Tabela 4.1 - Campos e dados da tabela Alunos	15
Tabela 4.2 - Dados recebidos após execução da query gerada	15
Tabela 5.1 - Plano de testes unitários	27
Tabela 5.2 - Plano de testes de integração	28
Tabela 5.3 - Plano de testes de sistema	35
Tabela 5.4 - Respostas às perguntas sobre a utilização da ferramenta.....	36
Tabela 5.5 - Respostas às perguntas sobre qualidade e usabilidade da ferramenta	37

LISTA DE SIGLAS E ABREVIATURAS

ANSI	American National Standards Institute
ISO	International Organization for Standardization
LINQ	Language Integrated Query
PDM	Physical Data Model
QG4B	Query Generator for Beginners
SQL	Structured Query Language

1. INTRODUÇÃO

1.1. CONTEXTUALIZAÇÃO

Hoje em dia, a linguagem SQL é a mais utilizada em base de dados (Russell & Cumming, 2004). No entanto, aprender esta linguagem é, por vezes, complicado. Um estudo elaborado por Antonija Mitrovic e Stellan Ohlsson mostra que o uso de uma ferramenta como auxiliar pedagógico pode ajudar um aluno a aprender e evoluir com os seus erros (Mitrovic & Ohlsson, 1999).

O problema na aprendizagem desta linguagem passa pelo acompanhamento dado aos alunos por parte dos Professores. Com a criação desta ferramenta, à qual foi dada o nome de *Query Generator for Beginners (QG4B)*, o utilizador desta aplicação tem um instrumento de ensino que possui uma ligação com o modelo relacional da base de dados apenas focado em si próprio, ajudando a que este melhore significativamente a maneira como aprende a linguagem SQL. Mais ainda, a ferramenta também gera expressões em álgebra relacional e em linguagem LINQ.

1.2. JUSTIFICAÇÃO DO PROJETO

Os utilizadores menos experientes em linguagens de base de dados são o principal foco deste projeto pois desta maneira podem criar facilmente e intuitivamente a *query* que necessitam de executar para procurar a informação que desejam, sem terem que conhecer profundamente a linguagem.

Por norma, utilizadores não-informáticos são capazes de fazer inúmeros erros na construção de uma *query*. Em bases de dados com vários milhares de registos, um erro como, por exemplo, correr uma expressão SQL em que esta irá fazer um *Table Scan* de duas tabelas pode afetar a performance do servidor de dados, e inclusive bloqueá-lo, o que pode prejudicar muito o negócio de uma empresa. A ferramenta permite ao utilizador reduzir essa possibilidade pois se a relação entre duas tabelas usadas já estiver definida no modelo relacional, então essa ligação é tida em conta durante a criação da *query*, evitando assim que haja uma sobrecarga desnecessária no servidor de dados. O uso da ferramenta também evita erros comuns durante a construção de expressões SQL como erros sintáticos e tipográficos.

A existência do *QG4B* proporciona aos estudantes iniciados na matéria de base de dados a capacidade de rapidamente validarem as suas respostas em exercícios através do modelo relacional, sendo por isso um potencial auxiliar pedagógico.

1.3. OBJETIVOS DO PROJETO

1.3.1. Objetivo Principal

O objetivo principal do projeto é o desenvolvimento de uma aplicação *desktop* capaz de gerar *queries* em três linguagens distintas de forma a constituir um auxiliar pedagógico a iniciantes de base de dados. Para tal, sendo dado um modelo relacional particular através de uma interface gráfica simples, o utilizador gera automaticamente *queries* em SQL e respetivo equivalente em álgebra relacional. Face à crescente utilização do LINQ, a ferramenta também dá a possibilidade de gerar expressões nesta linguagem.

Visto que a característica mais potente do SQL passa pela extração de dados existentes na base de dados (ClydeBank Technology, 2015), a ferramenta *QG4B* irá apenas suportar a consulta de dados, ou seja, o comando *SELECT*. Para além deste comando, a ferramenta suporta também as cláusulas *WHERE*, *GROUP BY*, *ORDER BY* e *JOIN*. Consequentemente, as outras duas linguagens também suportam apenas estas funcionalidades.

1.3.2. Objetivos Específicos

De forma a atingir o objetivo principal do projeto, foram definidos os seguintes objetivos:

- Estruturação da ferramenta de maneira a que esta seja de simples utilização para os seus utilizadores;
- Criação de um algoritmo que valide os valores inseridos pelo utilizador e posteriormente gere a *query* em linguagem SQL, álgebra relacional ou linguagem LINQ;
- Análise dos resultados aos testes realizados, de maneira a garantir a qualidade das respostas dadas pela ferramenta.

2. PLANO DE TRABALHO

O modelo *Waterfall* foi considerado como o que melhor se adapta ao projeto por se tratar de um modelo que oferece inúmeras vantagens no que toca ao desenvolvimento de *software* menos complexo e mais linear. O fato de cada etapa do seu ciclo de vida ter que estar completa antes do início da próxima permite um maior controlo sobre o desenvolvimento do projeto e também um aumento da qualidade do *software*. Por outro lado, também possibilita de uma maneira muito mais incisiva, determinar e corrigir possíveis falhas. Para além destes aspetos, com a utilização deste modelo, consegue-se avaliar o progresso obtido até ao momento no final de cada fase.

Sendo assim, foram delineadas as seguintes fases para a elaboração deste projeto:

Nome da Fase	Descrição da Fase
Definição de Requisitos	Especificação dos requisitos da aplicação Revisão e melhoria dos capítulos Enquadramento Teórico e Metodologia
<i>Design</i>	Representação do desenho da aplicação tendo por base os requisitos especificados
Codificação	Criação de um algoritmo que lê os dados relativos ao Modelo Relacional Criação de um algoritmo que gere a <i>query</i> em SQL, ou o seu equivalente em expressão algébrica ou LINQ, através dos dados e filtros pedidos pelo utilizador
Testes	Realização de Testes Unitários Realização de Testes de Sistema Realização de Testes de Integração Realização de Testes de Aceitação
Manutenção	Correção e aperfeiçoamento da ferramenta através da avaliação do <i>feedback</i> dado pelos utilizadores

Tabela 2.1 - Descrição das fases do modelo do projeto (Santos, 2012)

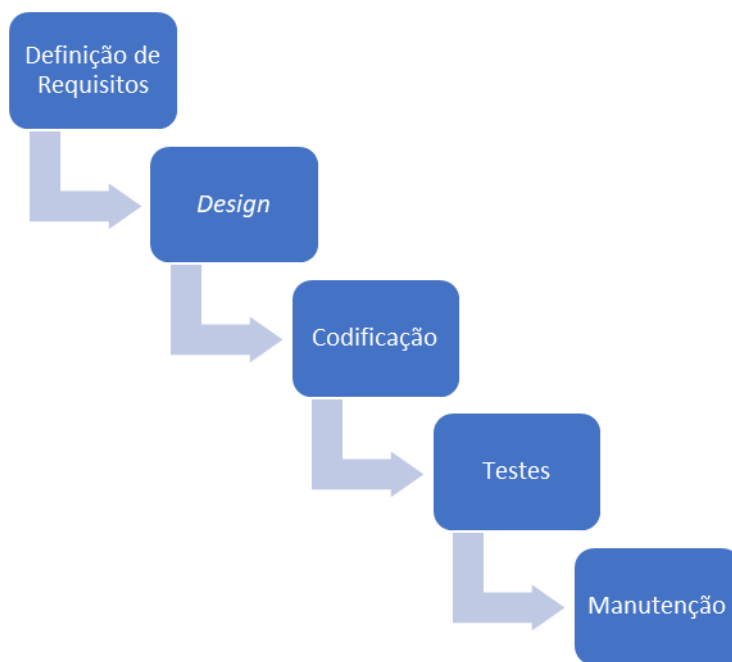


Figura 2.1 - Modelo do projeto (Santos, 2012)

A linguagem de programação escolhida para o desenvolvimento da ferramenta *Query Generator for Beginners* foi o C#, sendo que a aplicação *Microsoft Visual Studio*, da Microsoft, foi a escolhida para a realização desta ferramenta.

A ferramenta criada inicialmente pede ao utilizador que insira um modelo relacional de uma base de dados. A aplicação, ao receber o ficheiro introduzido pelo utilizador, irá ler e validar a informação presente no modelo relacional do ficheiro (metadados). Caso o ficheiro esteja no formato pretendido, a ferramenta abre uma nova janela onde é possível gerar *queries* com os dados presentes no modelo inserido pelo utilizador.

Na seguinte tabela (tabela 2.2) é apresentado um cronograma do plano proposto para a realização deste projeto.

Fases	Fase do Projeto	Atividades	2016					2017										
			AGO	SET	OUT	NOV	DEZ	JAN	FEV	MAR	ABR	MAI	JUN	JUL	AGO	SET	OUT	NOV
1	Preparação do Projeto	Definição de Requisitos	■															
		Revisão do Enquadramento Teórico		■														
		Revisão de Metodologia			■													
2	Criação da Ferramenta	Leitura de dados de um Modelo Relacional				■												
		Estruturação da Ferramenta					■	■										
		Criação do Algoritmo						■	■	■								
3	Testes à Ferramenta	Testes Unitários										■						
		Testes de Integração											■					
		Testes de Sistema												■	■			
		Testes de Aceitação																
4	Análise de Resultados e Conclusões												■					
5	Conclusão do Projeto	Escrita da Tese												■	■	■		
		Entrega da 1ª Fase da Tese													■			
		Correções Finais à Tese														■	■	
		Entrega Final da Tese																■

Tabela 2.2 - Cronograma do planeamento do projeto

3. ENQUADRAMENTO TEÓRICO

3.1. LINGUAGEM SQL

A linguagem SQL (*Structured Query Language*) foi inicialmente criada nos anos 70 pela IBM, com o intuito de simplificar a interação do utilizador com uma base de dados relacional, evitando assim escrever expressões em álgebra relacional (Gouveia, 2014). Esta linguagem consiste em expressões em Inglês, expressões estas que são usadas para inserir, modificar, apagar e consultar dados numa base de dados (Melton, 1998). A linguagem foi estandardizada ao longo dos anos pelas organizações ANSI (*American National Standards Institute*) e ISO (*International Organization for Standardization*) de maneira a que existisse um maior número de semelhanças entre os vários vendedores de sistemas de gestão de base de dados e as suas variantes (Kriegel, 2011).

As principais vantagens do uso de uma linguagem estandardizada comparativamente com o uso de expressões algébricas são as seguintes:

- Redução do custo de formação;
- Melhoria na produtividade;
- Maior portabilidade da aplicação, porque a aplicação que utilize linguagem SQL pode ser usada em qualquer computador desde que este utilize SQL;
- Maior longevidade da aplicação que use esta linguagem, pois o uso de uma linguagem estandardizada tende a manter-se igual durante muito tempo;
- Redução de dependência num simples vendedor (Sumathi & Esakkirajan, 2007).

Apesar da linguagem SQL não existir fora do modelo relacional, e não possuir algumas capacidades a nível de programação quando comparada com outras linguagens de programação, a verdade é que qualquer negócio moderno tem informações que precisam de ser guardadas numa base de dados, fazendo com que a aprendizagem desta linguagem seja essencial para qualquer interação com os seus dados (Stephens, Jones, & Plew, 2011).

A linguagem SQL pode ser dividida em 4 subconjuntos que representam as diversas funcionalidades presentes nesta linguagem, sendo estes os seguintes:

- DQL (*Data Query Language*) - Permite a consulta (*SELECT*) de registos na base de dados;
- DDL (*Data Definition Language*) - Permite a criação (*CREATE*), modificação (*ALTER*) e remoção (*DROP*) de tabelas e índices;
- DML (*Data Manipulation Language*) - Permite a modificação (*UPDATE*), inserção (*INSERT*) e remoção (*DELETE*) de registos na base de dados;
- DCL (*Data Control Language*) - Permite conceder (*GRANT*) ou revogar (*REVOKE*) permissões de um dado utilizador ou grupo à base de dados (ClydeBank Technology, 2015).

Os tipos de dados existentes na linguagem SQL podem ser categorizados em seis tipos principais de tipos de dados (Tale, 2016):

- Data e Tempo (*datetime*, *smalldatetime*, *date* e *time*);
- Numérico Exatos (*tinyint*, *bit*, *bigint*, *numeric*, *int*, *decimal*, *money*, *smallmoney* e *smallint*);
- Binário (*binary*, *varbinary* e *image*);
- Numérico Aproximado (*float* e *real*);
- Cadeia de Carateres Unicode (*ntext*, *nchar* e *nvarchar*);
- Cadeia de Carateres (*text*, *char* e *varchar*).

A principal característica da linguagem SQL é a sua habilidade para extrair dados existentes na base de dados, ou seja, o comando *SELECT*. No entanto, este comando pode utilizar múltiplas cláusulas, sendo que neste caso, as cláusulas de filtrar, de agrupar, de ordenar e de *joining* são também suportadas. A sintaxe de uma *query* SQL para consulta de dados presentes numa tabela tem o formato seguinte (ClydeBank Technology, 2015):

```
SELECT NomeTabela.NomeColuna
      FROM NomeTabela
      [INNER JOIN | LEFT OUTER JOIN | RIGHT OUTER JOIN | FULL OUTER JOIN NomeTabelaB ON
       NomeTabela.NomeColuna = NomeTabelaB.NomeColunaB]
      [WHERE Condição]
      [GROUP BY NomeTabela.NomeColuna]
      [ORDER BY NomeTabela.NomeColuna [ASCENDING | DESCENDING]]
```

As palavras em maiúsculas representam palavras reservadas pela linguagem, enquanto que o uso de parêntesis retos indica que a cláusula não é obrigatória. Para além de todas estas cláusulas, o comando *SELECT* também suporta o uso de funções de agregação, em que estão incluídas o *AVERAGE*, o *COUNT*, o *MAXIMUM*, o *MINIMUM* e o *SUM*. Caso uma coluna esteja a ser agrupada (cláusula *GROUP BY*) e tiver a ser usado um operador de agregação no comando *SELECT*, então também deve ter esse operador na cláusula *GROUP BY* (Sharma et al., 2010).

3.2. ÁLGEBRA RELACIONAL

Álgebra Relacional foi inicialmente definida em 1970 por Edgar Codd e consiste numa linguagem simples mas poderosa de construir novas relações a partir de determinadas relações (Garcia-Molina, Ullman, & Widom, 2008). As operações existentes nesta linguagem dividem-se em dois grupos: operações unárias, onde estão contemplados os operadores seleção, de projeção e de renomeação, e operações binárias que contêm os operadores de união, diferença, interseção, produto cartesiano, junção e divisão. No entanto, os operadores de Agrupar e Agregar não são suportados nesta linguagem (Elmasri & Navathe, 2010). Sendo assim, apenas os seguintes operadores são possíveis de ser utilizados em expressões algébricas:

- Seleção (σ);
- Projeção (π);
- Junção (\bowtie , \ltimes , \ltimes e \ltimes);
- Ordenação (τ);
- Produto Cartesiano (\times).

Apesar do uso desta linguagem ser muito reduzido atualmente, a Álgebra Relacional é parte integrante da maioria das bases de dados relacionais pois as *queries* em SQL quando são executadas no servidor de base de dados são traduzidas em operações de expressões algébricas (Elmasri & Navathe, 2010).

3.3. LINGUAGEM LINQ

A linguagem LINQ (*Language Integrated Query*) foi concebida em 2005 pela Microsoft para a linguagem de programação C# com o objetivo de fornecer ferramentas que possam melhorar a implementação de código a diferentes arquiteturas (Pialorsi & Russo, 2008). Devido ao crescente uso dos mais variados tipos de fontes de dados, o uso de LINQ mostra-se como uma forma cada vez mais importante e simples de manipular dados usando apenas uma sintaxe, sendo que os principais benefícios são os seguintes:

- Redução da quantidade de código;
- Melhoria na compreensão do código e no que este faz;
- Linguagem pode ser usada nos vários tipos de variantes existentes no LINQ (*LINQ to Objects, LINQ to SQL, etc.*);
- As *queries* podem ser compostas entre estas e construídas em etapas;
- As *queries* oferecem uma maior segurança para a identificação de erros pois estas tem que ser compiladas (Roberts, 2015).

Como foi mencionado anteriormente, o LINQ tem várias variantes, mas apenas a variante *LINQ to SQL* é utilizada pela ferramenta QG4B. Nesta variante, as *queries* criadas em LINQ para aceder ao conteúdo existente numa base de dados são convertidas para SQL através de um *provider*. Devido à sua maior legibilidade, o LINQ acaba por retirar a complexidade que antes existia quando se escreviam as *queries* em SQL no código C# (Abreu & Morgado, 2009).

3.4. FERRAMENTAS DE APOIO AO ENSINO DE INFORMÁTICA

A aprendizagem de uma linguagem de base de dados, nomeadamente SQL, através de uma ferramenta interativa é possível e, em alguns casos, recomendável. Estas ferramentas, apesar de auxiliares pedagógicos, estão apenas focadas num aluno singular. Ao contrário da habitual forma de ensino, ou seja, um professor ensina vários alunos, estas ferramentas focam-se apenas no utilizador, ajudando a que este tenha um ensino mais rápido e profundo da matéria (Russell & Cumming, 2004).

Em 2000, Antonija Mitrovic implementou um sistema de apoio aos alunos chamado *SQL-Tutor*. A autora refere no artigo “*Learning SQL with a computerized tutor*” (Mitrovic, 1998) que a orientação singular de um professor para apenas um aluno é mais vantajosa do que um professor para vários alunos, pois o aluno terá toda a atenção do professor. Neste caso, sendo o professor a ferramenta criada, esta ensina e corrige de forma interativa os alunos que estiverem a aprender SQL (Mitrovic & Martin, 2000). No entanto, é necessário ter especial atenção à estruturação da aplicação pois através de um estudo feito em 2011 a esta ferramenta, é possível verificar que nem todos os alunos conseguiram compreender o funcionamento desta aplicação (Guia et al., 2011).

Outra ferramenta de sucesso é a *SQLator* (Sadiq, Orłowska, Sadiq, & Lin, 2004), em que a principal “melhoria” relativamente à aplicação mencionada anteriormente é o facto de ser *web-based*, o que implica que está 24 horas *online* e qualquer utilizador pode usar em qualquer lugar, desde que este tenha acesso à Internet.

No artigo “*Learning SQL Programming with Interactive Tools: From Integration to Personalization*” (Brusilovsky et al., 2010) também são descritas as várias vantagens ao utilizar uma ferramenta de ensino na linguagem SQL. É possível verificar também que os alunos continuam a utilizar a ferramenta mesmo após a sua 1ª utilização, mostrando assim que gostam de confirmar através destas aplicações se as suas respostas estão corretas ou não.

Como é possível observar, o uso de aplicações para ensinar uma linguagem de base de dados não é recente. No entanto, nenhuma destas ferramentas faz a ligação com o modelo relacional, ou seja, estes aplicativos confirmam apenas se a *query* em SQL está correta sintacticamente, deixando de parte os erros tipográficos. Estes erros tipográficos acontecem normalmente quando um utilizador se engana ao escrever o nome de um campo.

4. ANÁLISE E DESENHO

Neste capítulo são definidos quais os requisitos face às necessidades dos seus utilizadores e qual a estrutura da ferramenta a ser desenvolvida, sendo que esta tem que estar em conformidade com os requisitos identificados nesta secção.

4.1. ANÁLISE DE REQUISITOS

4.1.1. Funcionais

- A ferramenta deve aceitar apenas ficheiros nos formatos *.pdm (Physical Data Model)* e *.sql (Structured Query Language)* e, posteriormente, verifica se a informação presente no ficheiro está na estrutura correta;
- A ferramenta tem a possibilidade de gerar *queries* nas linguagens SQL, Álgebra Relacional ou LINQ após a verificação dos campos preenchidos pelo utilizador;
- A aplicação não deve permitir que campos com um tipo de dados específico sejam preenchidos com um diferente tipo de dados, i.e., um campo do tipo inteiro não pode ser preenchido com caracteres alfabéticos;
- A ferramenta não deve permitir que o utilizador ultrapasse o tamanho limite em campos que tenham um número de caracteres definido.

4.1.2. Não Funcionais

- A ferramenta deve ser desenvolvida na linguagem C#;
- A aplicação deve ser de fácil instalação e utilização;
- A interface da aplicação deve ser *user-friendly* de maneira a proporcionar uma experiência fácil e agradável ao utilizador;
- A linguagem utilizada no *software* é a Inglesa de forma a possibilitar o uso de um maior número de pessoas.

4.2. ARQUITETURA DE SOFTWARE

4.2.1. Diagrama de Use Case

O seguinte diagrama de *Use Case* representa as várias possibilidades de como o utilizador poderá interagir com a ferramenta QG4B. Neste caso, o utilizador interage com a ferramenta ao fazer o *upload* do modelo relacional ou gerando uma *query*, sendo que nesta última interação estão incluídas as ações de escolha de campos para visualizar, filtrar ou criar um *join* entre as tabelas desses mesmos campos, para além da ação de escolha da linguagem pretendida para gerar a *query*.

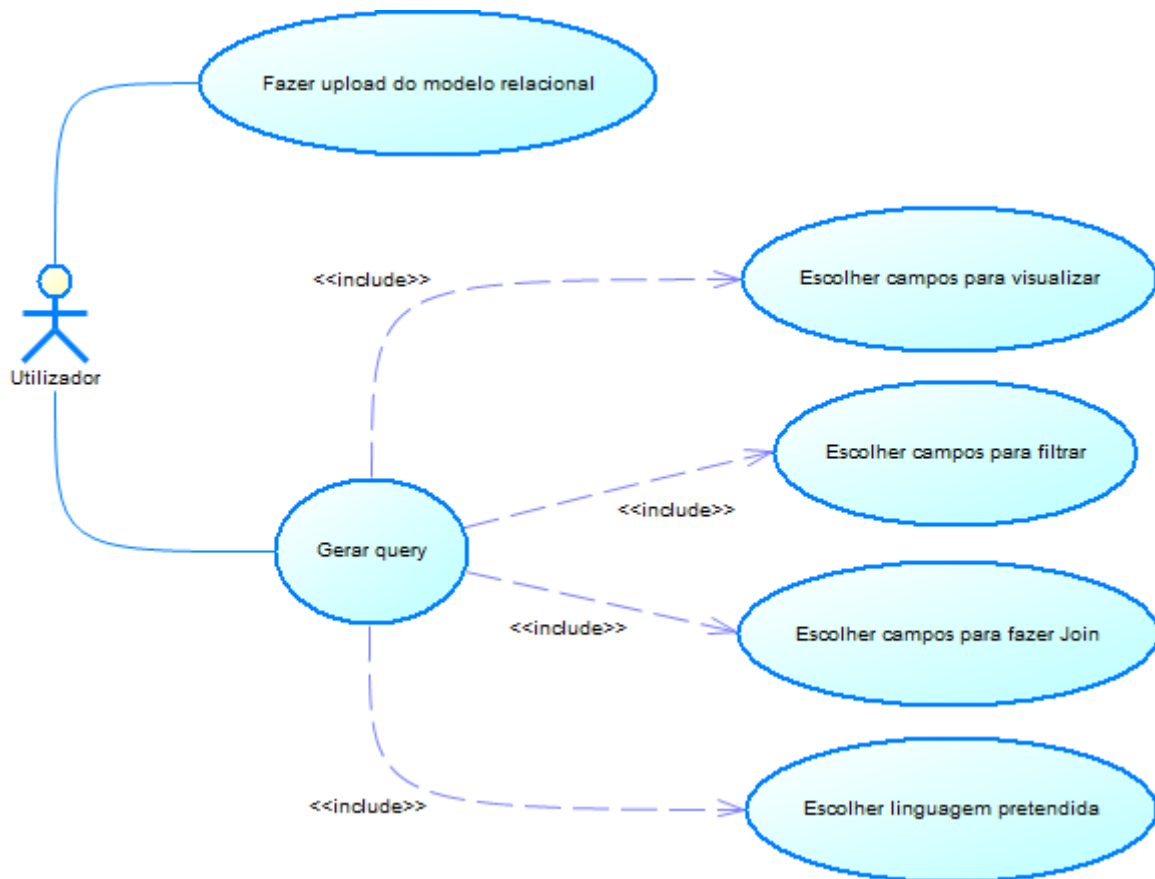


Figura 4.1 - Diagrama de Use Case

4.2.2. Diagrama de Atividades

O diagrama de atividades seguinte tem como finalidade mostrar o fluxo de atividades de todo o processo existente entre um utilizador e a ferramenta QG4B para a criação de uma *query*. Neste específico projeto, a atividade é começada pelo utilizador ao iniciar a ferramenta e, posteriormente, é feito o *upload* do modelo relacional. A ferramenta vai verificar o modelo relacional introduzido pelo utilizador e, se a estrutura deste não estiver correta, o fluxo regressa à atividade de fazer o *upload* do modelo relacional. Porém, se a estrutura estiver correta, o utilizador tem acesso a um novo conjunto de atividades que contemplam o preenchimento de campos de forma a obter a *query* pretendida. De seguida, o utilizador faz um pedido à ferramenta para gerar a *query* e esta vai verificar se existem erros no preenchimento dos campos. Caso existam erros, é necessário que o utilizador corrija esses mesmos erros sendo que o fluxo volta à atividade de preenchimento de campos. No entanto, caso não exista nenhum erro, a ferramenta gere a *query* e disponibiliza-a ao utilizador, sendo esta a última atividade registada neste diagrama.

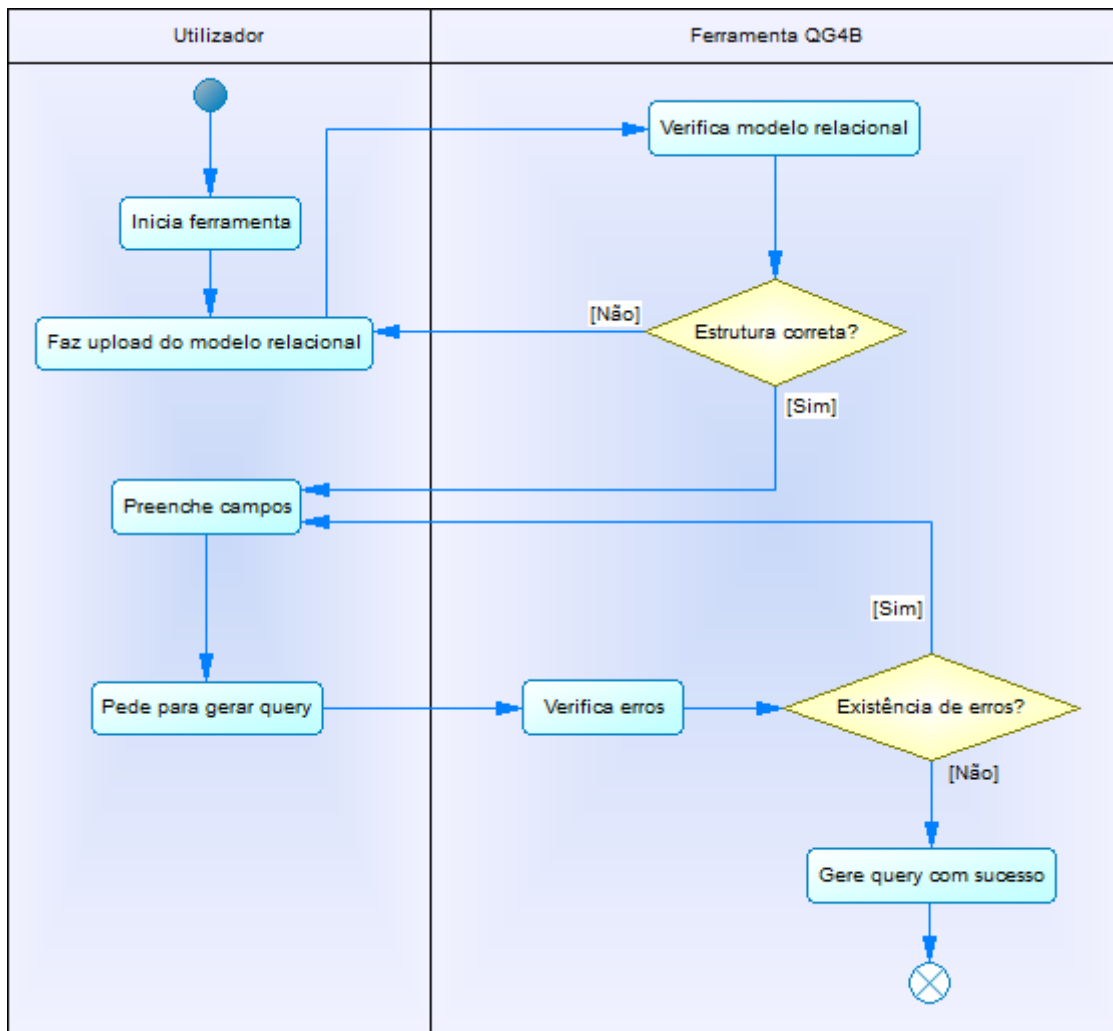


Figura 4.2 - Diagrama de Atividades

4.2.3. Diagrama de Classes

O seguinte diagrama de classes é uma representação da organização das classes e das suas relações existentes na ferramenta QG4B. O diagrama deste projeto é composto por seis classes, em que três delas são formulários da ferramenta (*frmUploadFile*, *frmGenerateQuery* e *frmAbout*) e as outras três foram classes criadas com o intuito de guardar dados relacionados com as tabelas e colunas presentes no modelo relacional (*TablesColumns*) e com os campos a mostrar e os *joins* definidos pelo utilizador (*SelectFields* e *SelectJoins*).

The image displays six class definition windows from a software development environment, showing the source code for various classes. Each window is organized into sections: Fields, Properties, and Methods.

- frmUploadFile Class:**
 - Fields:** btnAbout, btnNext, btnQuit, btnUpload, components, fileUpload, lblFileName, lstDataTypes, strErrorMessage, strFileName.
 - Methods:** btnAbout_Click, btnNext_Click, btnQuit_Click, btnUpload_Click, convertStringToInt, Dispose, extractString, extractStringBetweenTags, extractTableContent, frmUploadFile, InitializeComponent.
- frmGenerateQuery Class:**
 - Fields:** btnAddField, btnAddFilter, btnAddJoin, btnBack, btnCleanQuery, btnCopyClipboard, btnGenerate, btnQuit, components, dictNumericDataTypes, errProvider, flipSelectFields, flipSelectFilters, flipSelectJoins, grpSelectFields, grpSelectFilters, grpSelectJoins, intCounterFields, intCounterJoins, intCounterOrderByClauses, intCounterOrderByClauses, intCounterParentheses, intPreviousValueOrderBy, intWidthAggregateFunctions, intWidthCheckBoxGroupBy, intWidthCheckBoxOrderBy, intWidthClosingParentheses, intWidthComparisonOperator, intWidthFieldOne, intWidthFieldToFilter, intWidthFieldToShow, intWidthFieldTwo, intWidthJoinType, intWidthLabelGroupBy, intWidthLabelOrderBy, intWidthLabelOperator, intWidthOrderByOrder, intWidthOrderByType, intWidthStartingParentheses, intWidthValue, isLastWhereClause, isWhereClauseCorrect, lstJoins, lstTablesColumns, rdBLink, rdBRelationalAlgebra, rdBSQL, rdBFinalQuery, strErrorMessage.
 - Methods:** btnAddField_Click, btnAddFilter_Click, btnAddJoin_Click, btnBack_Click, btnCleanQuery_Click, btnCopyClipboard_Click, btnGenerate_Click, btnQuit_Click, btnRemoveField_Click, btnRemoveFilter_Click, btnRemoveJoin_Click, chbGroupBy_CheckedChanged, chbOrderBy_CheckedChanged, checkFilters, cmbAggregateFunctionChangeItems, cmbComparisonOperator_SelectedIndexChanged, cmbEqualFields_SelectedIndexChanged, cmbFieldToFilter_SelectedIndexChanged, cmbFieldToShow_SelectedIndexChanged, cmbOrderBy_DropDown, cmbOrderBy_SelectionChangeCommitted, convertStringToInt, decrementOrderBy, Dispose, frmGenerateQuery, generateQuery, getFieldDataType, getFieldLength, getTableName, InitializeComponent, pullUpDynamicObjects, refreshOrderBy(Dst).
- frmAbout Class:**
 - Fields:** btnOK, components, flipDescription, lblDescription, lblName, lblEmail, picAbout, PictureBox.
 - Methods:** btnOK_Click, Dispose, frmAbout, InitializeComponent, lblEmail_LinkClicked.
- TablesColumns Class:**
 - Properties:** ColumnDataType, ColumnLength, ColumnPrecision, TableColumn, TableColumnReference.
- SelectFields Class:**
 - Properties:** AggregateFunction, FieldToShow, GroupBy, OrderBy, OrderByOrder, OrderByType.
- SelectJoins Class:**
 - Properties:** FieldOne, FieldTwo, JoinType, TableJoined.

Figura 4.3 - Diagrama de Classes

4.3. INTERFACE

A figura seguinte ilustra um esboço da *interface* desenvolvida e o exemplo prático definido após a inserção do modelo relacional descrito no capítulo 4.4.

The interface is titled "Campos a mostrar (Passo 1)" and contains four columns of dropdown menus for selecting fields: "Nome", "Número", "Idade", and "E-mail". Below this is the "Filtros (Passo 2)" section, which is a table with four columns: "Campo", "Operador de Comparação", "Valor", and "Operador Lógico". The first row shows "Nome" selected in the "Campo" column, "Começa com" in "Operador de Comparação", "Gustavo" in "Valor", and "AND" in "Operador Lógico". The second row shows "Idade" in "Campo", "Maior que" in "Operador de Comparação", and "17" in "Valor". The third and fourth rows are empty. Below the filter table is a "Gerar QUERY (Passo 3)" button. At the bottom, a text box displays the generated SQL query: "Select Nome, Número, Idade, E-mail From Alunos Where Nome Like 'Gustavo%' And Idade > 17".

Campos a mostrar (Passo 1)			
Nome ▾	Número ▾	Idade ▾	E-mail ▾
▾	▾	▾	▾

Filtros (Passo 2)			
Campo	Operador de Comparação	Valor	Operador Lógico
Nome ▾	Começa com ▾	Gustavo	AND ▾
Idade ▾	Maior que ▾	17	▾
▾	▾		▾
▾	▾		▾

Gerar QUERY (Passo 3)

```
Select Nome, Número, Idade, E-mail
From Alunos
Where Nome Like "Gustavo%" And Idade > 17
```

Figura 4.4 - Esboço da *interface* da ferramenta QG4B

4.4. SIMULAÇÃO DE FUNCIONAMENTO

Para melhor explicação, em seguida apresenta-se uma simulação básica da utilização da ferramenta, em que é fornecido um modelo relacional com uma base de dados que contém apenas uma tabela chamada Alunos (que inclui informações sobre o Nome, Número, Idade, Morada e *E-mail* dos alunos). O sistema irá posteriormente permitir que o utilizador escolha quais os campos que deseja visualizar a informação e que campos deseja filtrar. Neste exemplo, a base de dados Alunos contém os seguintes dados:

Nome	Número	Idade	Morada	<i>E-mail</i>
Gustavo Reis	1	20	Rua das Silvas	gustavo.reis@exemplo.com
Maria Esteves	2	17	Rua das Cenouras	maria.esteves@exemplo.com
Ana Luísa	3	23	Rua das Couves	ana.luisa@exemplo.com
Gustavo Sousa	4	16	Rua das Cebolas	gustavo.sousa@exemplo.com
Filipe Pereira	5	21	Rua dos Agriões	filipe.pereira@exemplo.com
Gustavo Henriques	6	22	Rua das Sementes	gustavo.henriques@exemplo.com

Tabela 4.1 - Campos e dados da tabela Alunos

Neste exemplo específico, o utilizador deseja ver quais são os Nomes, Números, Idades e *E-mails* de alunos (a coluna Morada foi intencionalmente ignorada) que comecem com o nome de Gustavo e que tenham idade superior a 17 anos. O utilizador necessita de fazer os seguintes 3 passos na aplicação:

1. Preencher os campos que deseja visualizar, o que neste caso específico são o Nome, o Número, a Idade e o *E-mail* do aluno;
2. Escolher quais os campos a filtrar, ou seja, o nome do aluno tem que começar por Gustavo e a idade deste tem de ser maior do que 17;
3. Por último, o utilizador carrega no botão *Generate* e posteriormente, caso não existam erros, o sistema disponibiliza a *query* na caixa de texto.

Depois do utilizador correr na base de dados a *query* gerada pela ferramenta QG4B, os dados visualizados são estes:

Nome	Número	Idade	<i>E-mail</i>
Gustavo Reis	1	20	gustavo.reis@exemplo.com
Gustavo Henriques	6	22	gustavo.henriques@exemplo.com

Tabela 4.2 - Dados recebidos após execução da *query* gerada

5. IMPLEMENTAÇÃO DO PROTÓTIPO

Neste capítulo é descrito o processo de desenvolvimento da ferramenta. O protótipo foi desenvolvido na linguagem C# através da aplicação *Visual Studio*, sendo que durante a realização dos testes, foram criados vários modelos relacionais através da aplicação *PowerDesigner*. Foi utilizado também durante a fase de testes o modelo relacional de um exemplo de uma base de dados chamada *Northwind*, a qual é fornecida pela gratuitamente *Microsoft*.

5.1. ARQUITETURA DE SOFTWARE

A Arquitetura de um *Software* ilustra a estrutura do *software* de acordo com os requisitos previamente definidos, sendo que a imagem seguinte apresenta a arquitetura de *software* referente a este projeto. É possível verificar que a arquitetura é composta por cinco componentes: O utilizador, através da *interface* da ferramenta QG4B, faz o *upload* de um modelo relacional e, por conseguinte, essa mesma interface recebe a informação existente no ficheiro e disponibiliza-a ao utilizador. Posteriormente, o utilizador completa os campos de maneira a criar a *query* pretendida, e é aí que o gerador de *queries* entra e gera a mesma.

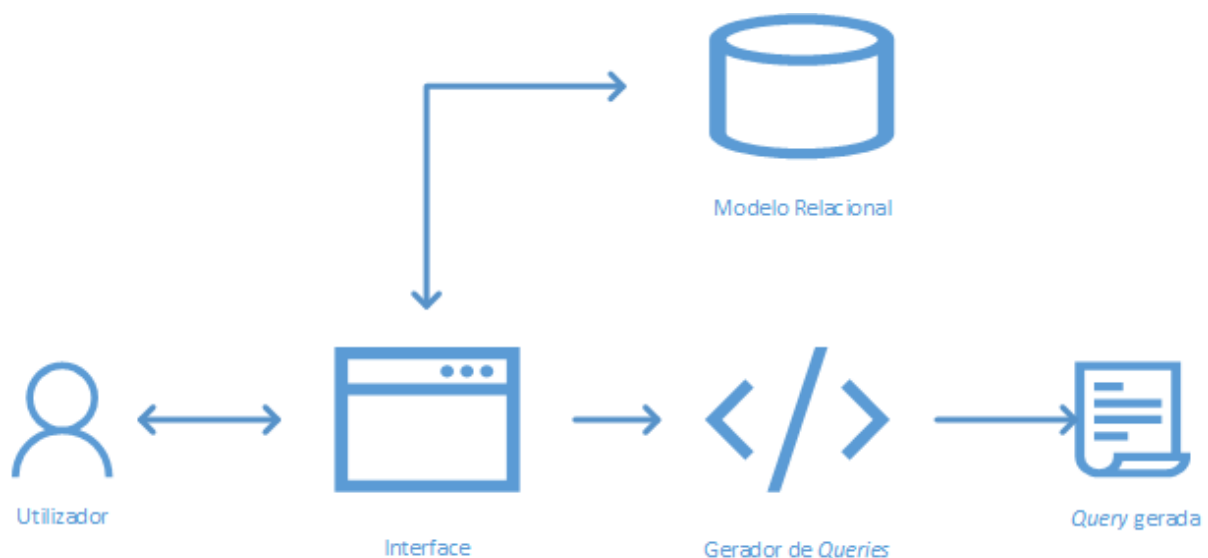


Figura 5.1 - Arquitetura de *Software*

5.2. ARQUITETURA DE SISTEMAS

A Arquitetura de Sistemas tem como objetivo a descrição e análise de um sistema através de um modelo que detalhe os vários componentes do sistema e como estes se relacionam entre eles. A imagem seguinte ilustra a arquitetura de sistemas relativa a este projeto, em que esta é dividida em quatro níveis. O primeiro nível representa a ferramenta QG4B e é através desta que o utilizador comunica com o sistema. O segundo nível é composto pelos componentes SQL e PDM que são os tipos de dados relacionais aceites pelo sistema. De seguida, vem a linguagem de programação C#, pois trata-se da linguagem usada para a realização da ferramenta e, por fim, a *framework* .NET que se mostra essencial para a execução da ferramenta QG4B.

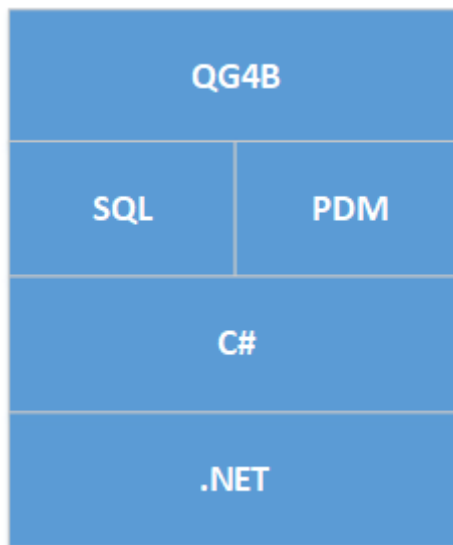


Figura 5.2 - Arquitetura de Sistemas

5.3. INTERFACES E USABILIDADE

A interface da ferramenta QG4B foi dividida em três janelas distintas: na primeira janela, é dada ao utilizador a possibilidade de fazer o *upload* do ficheiro do modelo relacional. Na segunda janela é possível gerar *queries* de acordo com as necessidades do utilizador e, por fim, na terceira janela existe uma pequena descrição da ferramenta.

5.3.1. *Upload* do Ficheiro do Modelo Relacional

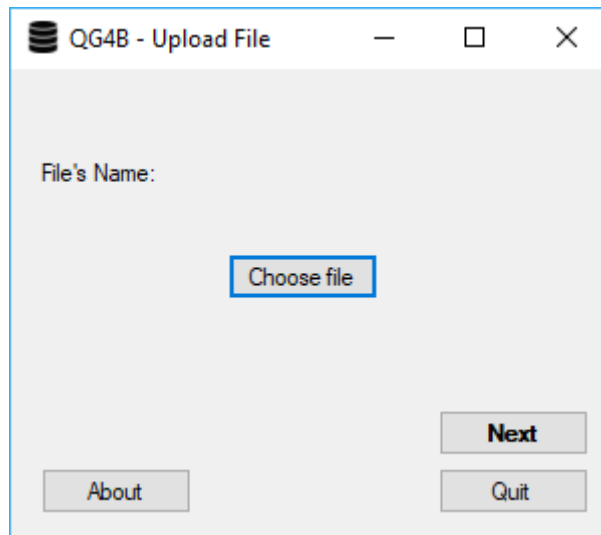


Figura 5.3 - Janela de *upload* do ficheiro

A primeira secção é composta por um formulário simples que contém uma *label*, a qual apresenta o nome do ficheiro inserido, e quatro botões (figura 5.3). O botão *About* abre uma nova janela (a denominada terceira janela), onde haverá uma pequena descrição sobre a ferramenta QG4B, enquanto que o botão *Quit* fechará a ferramenta e qualquer interação feita pelo utilizador nesta janela será perdida. O botão *Choose file* abre uma nova janela onde é pedido ao utilizador que selecione o modelo de dados que deseja usar para criar as suas *queries* na aplicação, como se pode ver na figura 5.4. Após o modelo relacional ser escolhido, é necessário carregar no botão *Next* para que a ferramenta verifique se os dados presentes no ficheiro contêm erros ou não.

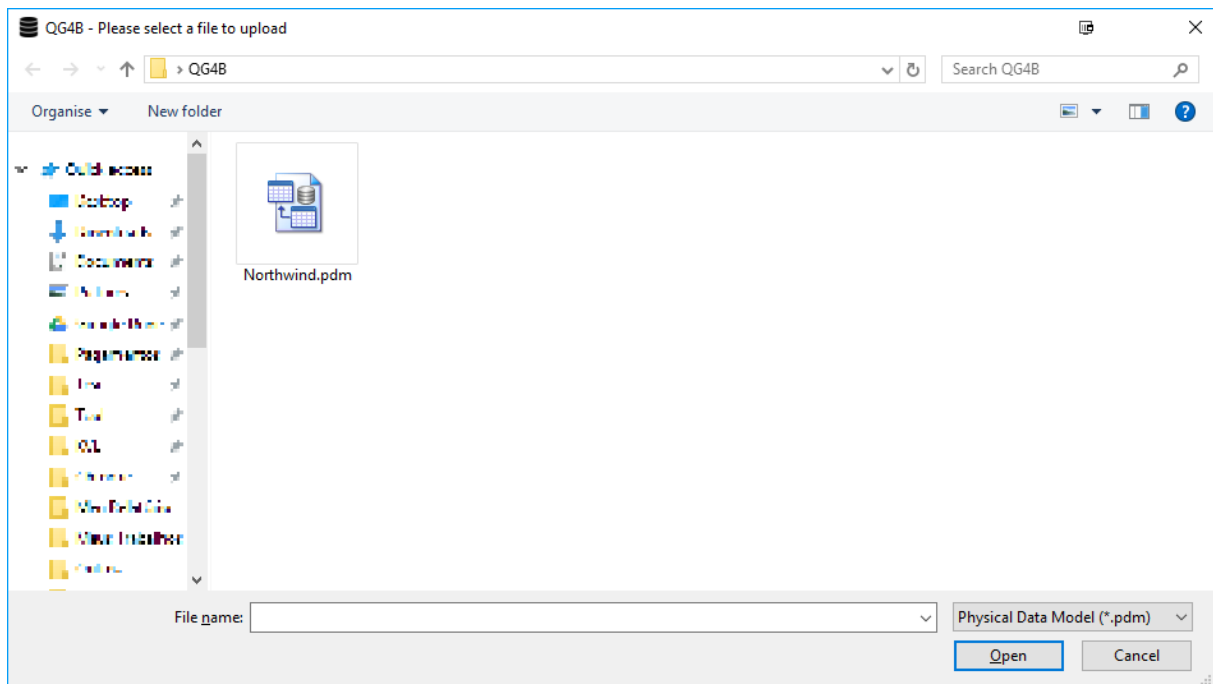


Figura 5.4 - Janela de escolha do ficheiro do modelo relacional

Em caso de erro, será mostrada uma janela de erro com o pedido para tentar de novo, como ilustrado na figura 5.5 e, caso não exista nenhum problema, é aberta uma nova janela (janela de Geração de *Queries*) onde o utilizador pode começar a construir as suas próprias *queries*.

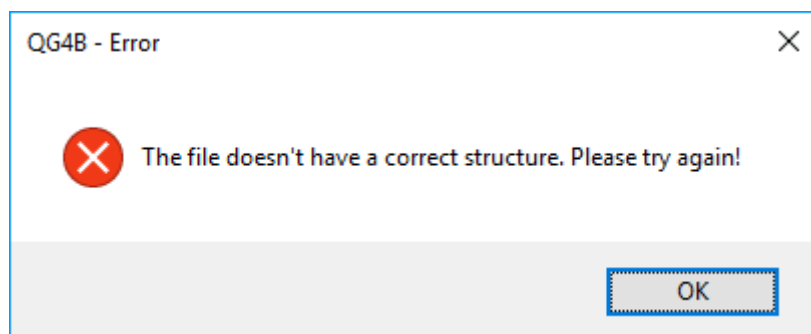


Figura 5.5 - Erro de estrutura incorreta do ficheiro inserido

5.3.2. Geração de *Queries*

The screenshot shows a software window titled "QG4B - Generate Query - Physical Data _4.pdm". The window is organized into three main sections for building a query:

- Select Fields:** A table with columns: Field To Show, Group By, Order By, Order By's Order, Order By's Type, and Aggregate Function. An "Add Field" button is on the right.
- Select Filters:** A table with columns: Field to Filter, Comparison Operator, Filter Value, and Logic Operator. An "Add Filter" button is on the right.
- Select Joins:** A table with columns: Field One, Field Two, and Join Type. An "Add Join" button is on the right.

At the bottom of the window, there is a "Generate" button, a "Clean Query" button, and a "Copy to Clipboard" button. On the far right, there are "About", "Back", and "Quit" buttons. At the bottom center, there are radio buttons for selecting the query language: SQL (selected), Relational Algebra, and LINGQ.

Figura 5.6 - Janela para gerar a *query* pretendida

Nesta nova janela (figura 5.6), o utilizador pode gerar as suas *queries* escolhendo quais os campos a mostrar e filtrar, para além de poder criar *joins* entre tabelas, caso não exista nenhuma conexão previamente definida no modelo relacional entre essas mesmas tabelas. No nome da janela (parte superior na janela) é apresentado o nome do modelo relacional inserido e o formulário é composto por três painéis (*Select Fields*, *Select Filters* e *Select Joins*) onde o utilizador pode criar a sua *query* da forma que pretende. Cada uma destas secções tem um botão para adicionar, respetivamente, um campo, filtro ou *join*. É então criada dinamicamente uma série de objetos, nomeadamente, botões, caixas de seleção (*check boxes*), caixas de combinações (*combo boxes*) e caixa de textos no respetivo painel, dando a possibilidade ao utilizador de construir a sua própria *query*.

Para além destes três painéis, existe também o botão *Generate* que verifica a existência de erros na *query* construída pelo utilizador e caso não exista nenhum, esta é disponibilizada numa caixa de texto situada na zona inferior da janela. Por baixo dessa caixa de texto, é dada a possibilidade ao utilizador de escolher a linguagem pretendida para a criação da *query* através de três botões de opções (*radio buttons*). Também existe à disposição o botão *Copy to Clipboard* que através de um clique simplifica o processo de copiar a *query* gerada, o botão *Clean Query* que limpa todas as opções

escolhidas até aquele momento pelo utilizador e, por fim, também é possível aceder à janela descritiva da ferramenta através do botão *About*. Para além disso, também foi criado um botão que fecha a ferramenta após o seu clique (botão *Quit*).

Durante a criação de uma *query*, a ferramenta ajuda o utilizador limitando as opções disponíveis em certos campos consoante o tipo de dados da coluna escolhida pelo utilizador. Por exemplo, após ter escolhido uma coluna do tipo alfanumérico, se o utilizador quiser criar uma função de agregação para essa coluna, só vai ter à sua disposição os operadores *COUNT*, *MAX* e *MIN*. Como os outros dois operadores (*AVG* e *SUM*) não são elegíveis para este tipo de dados, é então limitada essa escolha ao utilizador de forma a reduzir um erro.

Para além disso, quando o utilizador tentar gerar uma *query* e por alguma razão tiver um erro, a ferramenta mostra uma mensagem de erro com detalhes sobre este, e também indica através de um sinal de aviso onde esse mesmo erro poderá estar, como é possível verificar na figura 5.7 (assinalado com uma seta vermelha).

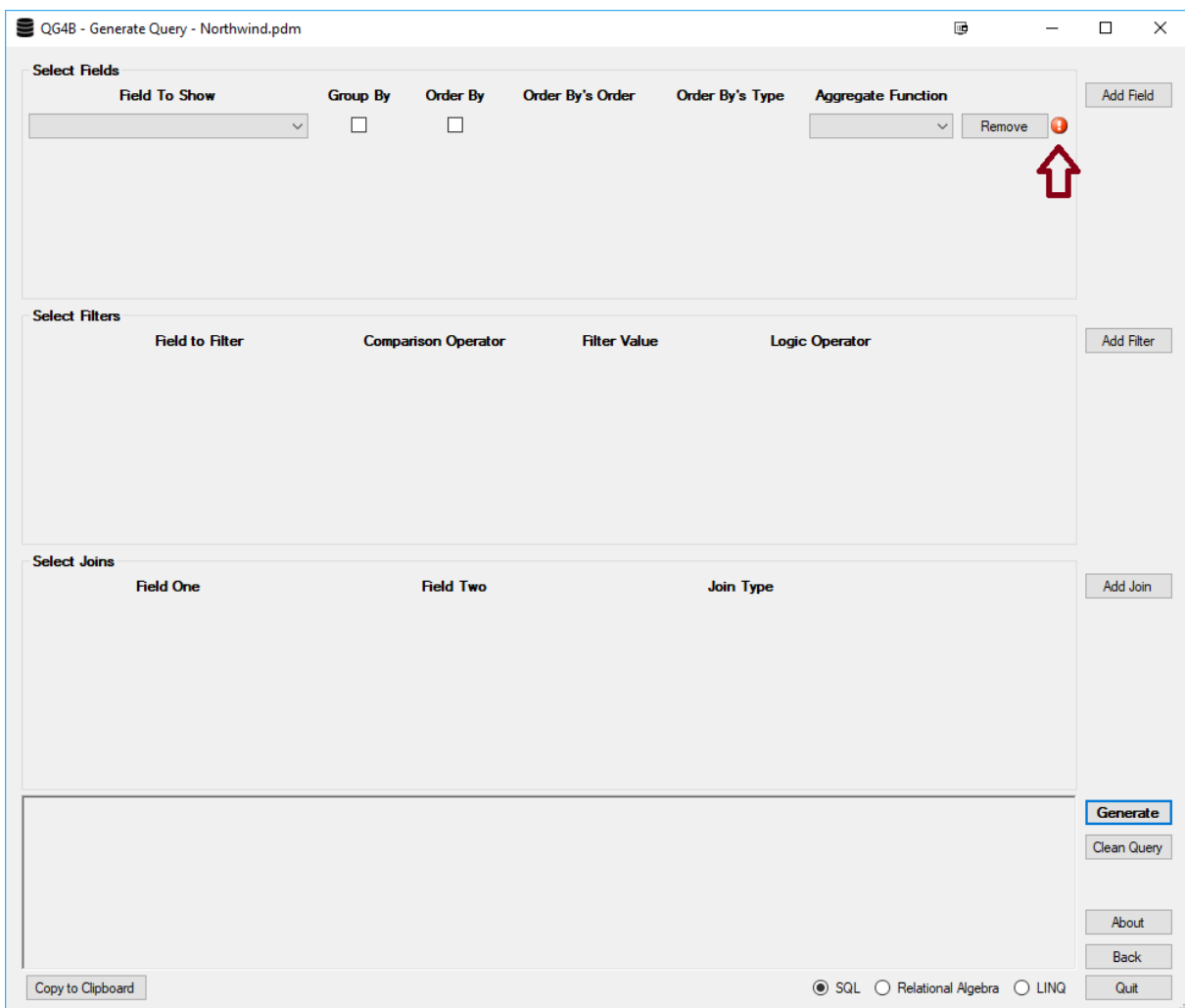


Figura 5.7 - Sinalização de erro numa tentativa de gerar uma *query*

Adicionalmente, o utilizador pode passar o cursor por cima do sinal de sinalização onde o erro está inserido, e uma nova caixa de texto aparece com detalhes deste.

Finalmente, e para demonstrar a utilização e funcionamento da ferramenta para a criação de uma *query* em linguagem SQL, foi utilizada como base a simulação previamente delineada no capítulo 4.4, como é possível verificar na figura 5.8.

The screenshot shows the 'QG4B - Generate Query - Physical Data _4.pdm' application window. It is divided into several sections for configuring a query:

- Select Fields:** A table with columns: Field To Show, Group By, Order By, Order By's Order, Order By's Type, and Aggregate Function. It lists 'Alunos.Nome', 'Alunos.Número', 'Alunos.Idade', and 'Alunos.E-mail'.
- Select Filters:** A table with columns: Field to Filter, Comparison Operator, Filter Value, and Logic Operator. It shows filters for 'Alunos.Nome' (Starts with 'Gustavo') and 'Alunos.Idade' (Greater than 17).
- Select Joins:** A table with columns: Field One, Field Two, and Join Type. It is currently empty.
- SQL Query:** A text area displaying the generated SQL query:

```
SELECT Alunos.Nome, Alunos.Número, Alunos.Idade, Alunos.E-mail
FROM Alunos
WHERE Alunos.Nome LIKE 'Gustavo%' AND Alunos.Idade > 17
```
- Buttons:** 'Add Field', 'Add Filter', 'Add Join', 'Generate', 'Clean Query', 'About', 'Back', 'Quit', and 'Copy to Clipboard'.
- Language Selection:** Radio buttons for 'SQL' (selected), 'Relational Algebra', and 'LING'.

Figura 5.8 - Exemplo de criação de uma *query*

5.3.3. Descrição da Ferramenta

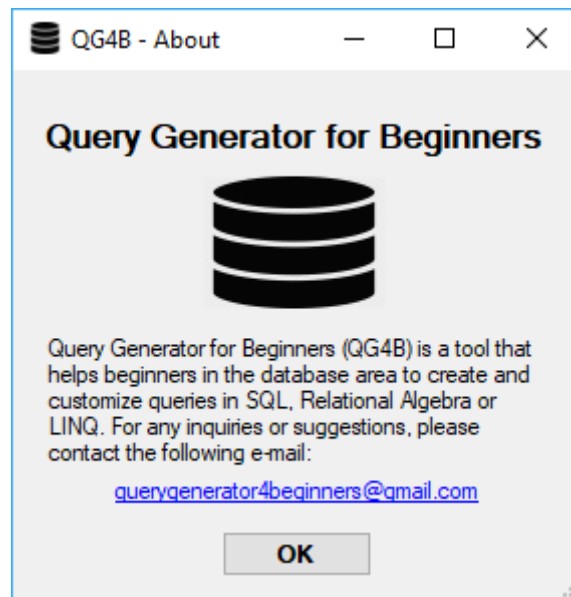


Figura 5.9 - Janela de descrição da ferramenta

Na janela descritiva do programa (figura 5.9), existe uma *label* que contém uma pequena descrição da ferramenta e um e-mail de contato que pode ser usado para quaisquer perguntas e sugestões que possam existir por parte dos utilizadores. Para além disso, esta secção também inclui um botão para voltar para a janela antiga.

5.4. TESTES

O plano de testes pretende detalhar quais os aspetos da ferramenta que irão ser testados, como é que esses testes serão feitos, os resultados esperados e os resultados obtidos. É importante explicar os tipos de testes que foram feitos ao longo do desenvolvimento do projeto, assim como as suas principais características, pois estes permitem avaliar se a ferramenta contempla as funcionalidades previamente definidas, garantindo assim a qualidade do *software* desenvolvido.

- Testes Unitários - Testes executados a cada um dos vários componentes da ferramenta de maneira a determinar se cada um destes componentes executa aquilo a que foi previamente definido;
- Testes de Integração - Testes realizados entre os vários subsistemas do programa de forma a avaliar as interfaces e a integração destes;
- Testes de Sistema - Testes que são feitos após o sistema estar completamente integrado para verificar se o projeto foi corretamente desenvolvido tendo em conta os requisitos. Estes testes não requerem conhecimento da estrutura interna da aplicação, pois são realizados somente com a preocupação dos aspetos gerais do sistema. Deste modo, os testes de sistema não se limitam a testar apenas os requisitos funcionais, mas também requisitos não funcionais;
- Testes de Aceitação - Testes realizados antes de sua disponibilização, por um conjunto de indivíduos com conhecimentos básicos em base de dados. Estes testes incidem sobre a utilidade e funcionalidade do sistema, de modo a provar a relevância com maior probabilidade de conter erros e têm como objetivo verificar o sistema em relação aos seus requisitos originais e às necessidades atuais do utilizador.

Nas seguintes secções estão descritos os vários planos de testes, assim como as ações seguidas durante a realização dos mesmos, o resultado esperado e o resultado obtido. Visto a ferramenta aceitar dois tipos de modelos relacionais, foram usados dois tipos de ficheiros: Para o tipo de ficheiro .sql, foi usado o modelo relacional disponível na base de dados *Northwind*, que é uma base de dados base fornecida pela *Microsoft* para o *SQL Server*, sendo que para o segundo tipo de ficheiro aceitável (.pdm), foi criado um modelo relacional idêntico à base de dados *Northwind*.

5.4.1. Testes Unitários

Descrição	Condição	Ação	Resultado Esperado	Resultado Obtido
Upload do modelo relacional		Na janela <i>QG4B - Upload File</i> , carregar no botão <i>Choose file</i> . De seguida, é necessário selecionar o modelo relacional pretendido e carregar no botão <i>Open</i>	Nome do ficheiro na janela <i>QG4B - Upload File</i> é visível	Nome do ficheiro visível com sucesso
Saída do programa		Pressionar o botão <i>Quit</i> da janela <i>QG4B - Upload File</i>	Ferramenta é encerrada	Ferramenta encerrada com sucesso
Adição de um campo		Pressionar o botão <i>Add Field</i> da janela <i>QG4B - Generate Query</i>	Campo é adicionado no painel <i>Select Fields</i>	Campo adicionado com sucesso
Adição de um filtro		Pressionar o botão <i>Add Filter</i> da janela <i>QG4B - Generate Query</i>	Filtro é adicionado no painel <i>Select Filters</i>	Filtro adicionado com sucesso
Adição de um <i>join</i>		Pressionar o botão <i>Add Join</i> da janela <i>QG4B - Generate Query</i>	<i>Join</i> é adicionado no painel <i>Select Joins</i>	<i>Join</i> adicionado com sucesso
Ativação/Desativação da marca na caixa de seleção <i>Group By</i>	Necessário a criação de um campo	Ativar/Desativar marca na caixa de seleção <i>Group By</i> do painel <i>Select Fields</i> da janela <i>QG4B - Generate Query</i>	Marca na caixa de seleção <i>Group By</i> é ativada/desativada	Marca na caixa de seleção <i>Group By</i> ativada/desativada com sucesso
Ativação da marca na caixa de seleção <i>Order By</i>	Necessário a criação de um campo	Ativar marca na caixa de seleção <i>Order By</i> do painel <i>Select Fields</i>	Marca na caixa de seleção <i>Order By</i> é ativada	Marca na caixa de seleção <i>Order By</i> ativada com sucesso
			Duas caixas de combinações que contém a ordem e o tipo de ordem são criadas	Duas caixas de combinações criadas com sucesso

Descrição	Condição	Ação	Resultado Esperado	Resultado Obtido
Modificação da ordem na caixa de combinação <i>Order By's Order</i>	Marca na caixa de seleção <i>Order By's</i> necessita estar ativa	Alterar ordem na caixa de combinação <i>Order By's Order</i>	Troca de ordem entre a caixa de combinação pretendida e a caixa de combinação onde o número da ordem desejada estava antigamente é efetuada	Troca de ordem bem-sucedida
Desativação da marca na caixa de seleção <i>Order By</i>	Marca na caixa de seleção <i>Order By's</i> necessita estar ativa	Desativar marca na caixa de seleção <i>Order By</i> do painel <i>Select Fields</i> da janela <i>QG4B - Generate Query</i>	Marca na caixa de seleção <i>Order By</i> é desativada	Marca na caixa de seleção <i>Order By</i> desativada com sucesso
			Duas caixas de combinações que contém a ordem e o tipo de ordem são eliminadas	Duas caixas de combinações eliminadas com sucesso
Mudança dos valores disponíveis para escolha no campo <i>Orientation Operator</i> após ser inserido um valor no campo <i>Field to Filter</i>	Necessário a criação de um filtro	No painel <i>Select Filters</i> da janela <i>QG4B - Generate Query</i> , introduzir um valor no campo <i>Field to Filter</i> e verificar se os valores disponíveis para selecionar no campo <i>Orientation Operator</i> mudaram consoante o tipo de campo selecionado (i.e. texto, numérico, etc.)	Mudança de valores disponíveis no campo <i>Orientation Operator</i> feita após ser selecionado um valor no campo <i>Field to Filter</i>	Mudança de valores disponíveis no campo <i>Orientation Operator</i> feita com sucesso
Proibição de um <i>join</i> de dois campos da mesma tabela	Necessário a criação de um <i>join</i>	No painel <i>Select Joins</i> da janela <i>QG4B - Generate Query</i> , introduzir a mesma tabela, mas colunas diferentes nos campos <i>Field One</i> e <i>Field Two</i> de um <i>join</i>	Aparecimento de uma caixa de erro	Aparecimento de uma caixa de erro com sucesso
Proibição de valores repetidos nos campos presentes num <i>join</i>	Necessário a criação de um <i>join</i>	No painel <i>Select Joins</i> da janela <i>QG4B - Generate Query</i> , introduzir o mesmo valor nos campos <i>Field One</i> e <i>Field Two</i> de um <i>join</i>	Aparecimento de uma caixa de erro	Aparecimento de uma caixa de erro com sucesso

Descrição	Condição	Ação	Resultado Esperado	Resultado Obtido
Remoção de um campo	Necessário a criação de um campo	Pressionar o botão <i>Remove</i> no painel <i>Select Fields</i> da janela <i>QG4B - Generate Query</i>	Campo é removido no painel <i>Select Fields</i>	Campo removido no painel <i>Select Fields</i> com sucesso
Remoção de um filtro	Necessário a criação de um filtro	Pressionar o botão <i>Remove</i> no painel <i>Select Filters</i> da janela <i>QG4B - Generate Query</i>	Filtro é removido do painel <i>Select Filters</i>	Filtro removido do painel <i>Select Filters</i> com sucesso
Remoção de um <i>join</i>	Necessário a criação de um <i>join</i>	Pressionar o botão <i>Remove</i> no painel <i>Select Joins</i> da janela <i>QG4B - Generate Query</i>	<i>Join</i> é removido do painel <i>Select Joins</i>	<i>Join</i> removido do painel <i>Select Joins</i> com sucesso
Sair do programa		Pressionar o botão <i>Quit</i> da janela <i>QG4B - Generate Query</i>	Ferramenta é encerrada	Ferramenta encerrada com sucesso

Tabela 5.1 - Plano de testes unitários

5.4.2. Testes de Integração

Descrição	Condição	Ação	Resultado Esperado	Resultado Obtido
Visualização da janela de descrição da ferramenta		Na janela <i>QG4B - Upload File</i> , pressionar o botão <i>About</i>	Nova janela aberta com informação relativa à ferramenta	Janela aberta com sucesso
		Na janela <i>QG4B - Generate Query</i> , pressionar o botão <i>About</i>	Nova janela aberta com informação relativa à ferramenta	Janela aberta com sucesso
Saída da janela de descrição da ferramenta		Na janela <i>QG4B - About</i> , pressionar o botão <i>OK</i>	A janela atual é fechada e a antiga janela é aberta	Janela atual fechada e antiga janela aberta com sucesso
Verificação do modelo relacional e da sua estrutura	Necessário que o <i>upload</i> do modelo relacional tenha sido realizado	Na janela <i>QG4B - Upload File</i> , após fazer o <i>upload</i> do modelo relacional, carregar no botão <i>Next</i> . Na janela seguinte (<i>QG4B - Generate Query</i>), adicionar um campo e verificar através da caixa de combinações se existem campos que possam ser escolhidos	Existência de campos disponíveis para escolha na caixa de combinações <i>Field To Show</i> na janela <i>QG4B - Generate Query</i>	Campos para escolha na caixa de combinações <i>Field To Show</i> disponíveis com sucesso
Retorno à janela inicial de <i>upload</i> do modelo relacional	Necessário que a verificação do modelo relacional tenha a estrutura correta	Na janela <i>QG4B - Generate Query</i> , pressionar no botão <i>Back</i>	A janela atual é fechada e a janela inicial <i>QG4B - Upload File</i> é aberta	Janela atual fechada e janela inicial aberta com sucesso

Tabela 5.2 - Plano de testes de integração

5.4.3. Testes de Sistema

Descrição	Número do Passo	Ação	Resultado Esperado	Resultado Obtido
<i>Upload</i> do modelo relacional do tipo .pdm/.sql e verificação da estrutura deste		Na janela <i>QG4B - Upload File</i> , carregar no botão <i>Choose file</i> . De seguida, é necessário seleccionar o modelo relacional, carregar no botão <i>Open</i> e posteriormente no botão <i>Next</i>	Uma nova janela chamada <i>QG4B - Generate Query</i> é aberta	Nova janela aberta com sucesso
Criação de uma <i>query</i> com um campo, dois filtros e um <i>join</i> na linguagem SQL/Álgebra Relacional/LINQ	1º	Na janela <i>QG4B - Generate Query</i> , adicionar um campo pressionando o botão <i>Add Field</i>	Campo é adicionado no painel <i>Select Fields</i>	Campo adicionado com sucesso
	2º	No painel <i>Select Fields</i> , seleccionar um campo na caixa de combinações <i>Field To Show</i>	Valor na caixa de combinações <i>Field To Show</i> é alterado	Valor na caixa de combinações <i>Field To Show</i> alterado com sucesso
	3º	Pressionar o botão <i>Generate</i>	A <i>query</i> é gerada na linguagem pretendida e visível na caixa de texto da janela <i>QG4B - Generate Query</i>	<i>Query</i> gerada com sucesso
	4º	No painel <i>Select Fields</i> , ativar a marca na caixa de seleção <i>Group By</i>	Marca na caixa de seleção <i>Group By</i> é ativada	Marca na caixa de seleção <i>Group By</i> ativada com sucesso
	5º	Pressionar o botão <i>Generate</i>	A <i>query</i> é gerada na linguagem pretendida e visível na caixa de texto da janela <i>QG4B - Generate Query</i>	<i>Query</i> gerada com sucesso

Descrição	Número do Passo	Ação	Resultado Esperado	Resultado Obtido
	6º	No painel <i>Select Fields</i> , ativar a marca na caixa de seleção <i>Order By</i> do painel <i>Select Fields</i>	Marca na caixa de seleção <i>Order By</i> é ativada e duas caixas de combinações que contém a ordem e o tipo de ordem são criadas	Marca na caixa de seleção <i>Order By</i> ativada e duas caixas de combinações criadas com sucesso
	7º	Pressionar o botão <i>Generate</i>	A <i>query</i> é gerada na linguagem pretendida e visível na caixa de texto da janela <i>QG4B - Generate Query</i>	<i>Query</i> gerada com sucesso
	8º	No painel <i>Select Fields</i> , selecionar um campo na caixa de combinações <i>Aggregate Function</i>	Valor na caixa de combinações <i>Aggregate Function</i> é alterado	Valor na caixa de combinações <i>Aggregate Function</i> alterado com sucesso
	9º	Pressionar o botão <i>Generate</i>	A <i>query</i> é gerada na linguagem pretendida e visível na caixa de texto da janela <i>QG4B - Generate Query</i>	<i>Query</i> gerada com sucesso
	10º	Na janela <i>QG4B - Generate Query</i> , adicionar dois filtros pressionando o botão <i>Add Filter</i> duas vezes	Filtros são adicionados no painel <i>Select Filters</i>	Filtros adicionados com sucesso
	11º	No primeiro filtro do painel <i>Select Filters</i> , selecionar um filtro na caixa de combinações <i>Field To Filter</i>	Valor na caixa de combinações <i>Field To Filter</i> é alterado	Valor na caixa de combinações <i>Field To Filter</i> alterado com sucesso

Descrição	Número do Passo	Ação	Resultado Esperado	Resultado Obtido
	12º	No primeiro filtro do painel <i>Select Filters</i> , selecionar um valor na caixa de combinações <i>Orientation Operator</i>	Valor na caixa de combinações <i>Orientation Operator</i> é alterado	Valor na caixa de combinações <i>Orientation Operator</i> alterado com sucesso
	13º	No primeiro filtro do painel <i>Select Filters</i> , introduzir um valor na caixa de texto <i>Filter Value</i>	Valor na caixa de texto <i>Filter Value</i> é introduzido	Valor na caixa de texto <i>Filter Value</i> introduzido com sucesso
	14º	No primeiro filtro do painel <i>Select Filters</i> , selecionar um valor na caixa de combinações <i>Logic Operator</i>	Valor na caixa de combinações <i>Logic Operator</i> é alterado	Valor na caixa de combinações <i>Logic Operator</i> alterado com sucesso
	15º	No segundo filtro do painel <i>Select Filters</i> , selecionar um filtro na caixa de combinações <i>Field To Filter</i>	Valor na caixa de combinações <i>Field To Filter</i> é alterado	Valor na caixa de combinações <i>Field To Filter</i> alterado com sucesso
	16º	No segundo filtro do painel <i>Select Filters</i> , selecionar um valor na caixa de combinações <i>Orientation Operator</i>	Valor na caixa de combinações <i>Orientation Operator</i> é alterado	Valor na caixa de combinações <i>Orientation Operator</i> alterado com sucesso
	17º	No segundo filtro do painel <i>Select Filters</i> , introduzir um valor na caixa de texto <i>Filter Value</i>	Valor na caixa de texto <i>Filter Value</i> é introduzido	Valor na caixa de texto <i>Filter Value</i> introduzido com sucesso

Descrição	Número do Passo	Ação	Resultado Esperado	Resultado Obtido
	18º	Pressionar o botão <i>Generate</i>	A <i>query</i> é gerada na linguagem pretendida e visível na caixa de texto da janela <i>QG4B - Generate Query</i>	<i>Query</i> gerada com sucesso
	19º	Pressionar os botões de parêntesis de maneira a que hajam sempre dois parêntesis (um parêntesis inicial e outro final)	Os botões de parêntesis são ativados e assumem uma cor laranja	Os botões de parêntesis são ativados e assumem uma cor laranja com sucesso
	20º	Pressionar o botão <i>Generate</i>	A <i>query</i> é gerada na linguagem pretendida e visível na caixa de texto da janela <i>QG4B - Generate Query</i>	<i>Query</i> gerada com sucesso
	21º	Na janela <i>QG4B - Generate Query</i> , adicionar um <i>join</i> pressionando o botão <i>Add Join</i>	<i>Join</i> adicionado no painel <i>Select Joins</i>	<i>Join</i> adicionado com sucesso
	22º	No painel <i>Select Joins</i> , selecionar um <i>join</i> na caixa de combinações <i>Field One</i>	Valor na caixa de combinações <i>Field One</i> é alterado	Valor na caixa de combinações <i>Field One</i> alterado com sucesso
	23º	No painel <i>Select Joins</i> , selecionar um <i>join</i> na caixa de combinações <i>Field Two</i>	Valor na caixa de combinações <i>Field Two</i> é alterado	Valor na caixa de combinações <i>Field Two</i> alterado com sucesso
	24º	No painel <i>Select Joins</i> , selecionar o tipo de <i>join</i> desejado na caixa de combinações <i>Join Type</i>	Valor na caixa de combinações <i>Join Type</i> é alterado	Valor na caixa de combinações <i>Join Type</i> alterado com sucesso

Descrição	Número do Passo	Ação	Resultado Esperado	Resultado Obtido
	25º	Pressionar o botão <i>Generate</i>	A <i>query</i> é gerada na linguagem pretendida e visível na caixa de texto da janela <i>QG4B - Generate Query</i>	<i>Query</i> gerada com sucesso
Remoção de um campo com um <i>Order By</i> definido e verificação da recomposição das ordens dos restantes campos	1º	Na janela <i>QG4B - Generate Query</i> , adicionar dois campos pressionando o botão <i>Add Field</i> duas vezes	Campos são adicionados no painel <i>Select Fields</i>	Campos adicionados com sucesso
	2º	No painel <i>Select Fields</i> , ativar a marca nas duas caixas de seleção <i>Order By</i> do painel <i>Select Fields</i>	Marcas nas caixas de seleção <i>Order By</i> são ativadas e duas caixas de combinações que contém a ordem e o tipo de ordem são criadas	Marcas nas caixas de seleção <i>Order By</i> ativadas e duas caixas de combinações criadas com sucesso
	3º	No painel <i>Select Fields</i> , pressionar o botão <i>Remove</i> do campo da qual a marca da caixa de seleção foi primeiramente ativada	Campo removido no painel <i>Select Fields</i> e a caixa de seleção <i>Order By</i> do campo ainda existente é alterada	Campo removido no painel <i>Select Fields</i> e caixa de seleção <i>Order By</i> do campo ainda existente é alterado com sucesso
Verificação da ferramenta para que esta não permita que o utilizador preencha o valor de um filtro com um tamanho superior ao definido no modelo relacional	1º	Na janela <i>QG4B - Generate Query</i> , adicionar um campo pressionando o botão <i>Add Field</i>	Campo é adicionado no painel <i>Select Fields</i>	Campo adicionado com sucesso
	2º	No painel <i>Select Fields</i> , seleccionar um campo na caixa de combinações <i>Field To Show</i>	Valor na caixa de combinações <i>Field To Show</i> é alterado	Valor na caixa de combinações <i>Field To Show</i> alterado com sucesso

Descrição	Número do Passo	Ação	Resultado Esperado	Resultado Obtido
	3º	Na janela <i>QG4B - Generate Query</i> , adicionar um filtro pressionando o botão <i>Add Filter</i>	Filtro é adicionado no painel <i>Select Filters</i>	Filtro adicionado com sucesso
	4º	No painel <i>Select Filters</i> , seleccionar um filtro com um tamanho limite na caixa de combinações <i>Field To Filter</i>	Valor na caixa de combinações <i>Field To Filter</i> é alterado	Valor na caixa de combinações <i>Field To Filter</i> alterado com sucesso
	5º	No painel <i>Select Filters</i> , seleccionar um valor na caixa de combinações <i>Orientation Operator</i>	Valor na caixa de combinações <i>Orientation Operator</i> é alterado	Valor na caixa de combinações <i>Orientation Operator</i> alterado com sucesso
	6º	No painel <i>Select Filters</i> , introduzir um valor na caixa de texto <i>Filter Value</i> com um tamanho superior ao limite previamente definido no modelo relacional	Valor na caixa de texto <i>Filter Value</i> é introduzido	Valor na caixa de texto <i>Filter Value</i> introduzido com sucesso
	7º	Pressionar o botão <i>Generate</i>	Surge uma mensagem de erro referindo que o valor introduzido é superior ao tamanho limite definido para o campo	Mensagem de erro visível
Verificação da ferramenta para que esta não permita que o utilizador preencha o valor de um filtro com um diferente tipo de dados, i.e., um campo do tipo inteiro não pode ser preenchido com caracteres alfabéticos	1º	Na janela <i>QG4B - Generate Query</i> , adicionar um campo pressionando o botão <i>Add Field</i>	Campo é adicionado no painel <i>Select Fields</i>	Campo adicionado com sucesso
	2º	No painel <i>Select Fields</i> , seleccionar um campo na caixa de combinações <i>Field To Show</i>	Valor na caixa de combinações <i>Field To Show</i> é alterado	Valor na caixa de combinações <i>Field To Show</i> alterado com sucesso

Descrição	Número do Passo	Ação	Resultado Esperado	Resultado Obtido
	3º	Na janela <i>QG4B - Generate Query</i> , adicionar um filtro pressionando o botão <i>Add Filter</i>	Filtro é adicionado no painel <i>Select Filters</i>	Filtro adicionado com sucesso
	4º	No painel <i>Select Filters</i> , selecionar um filtro na caixa de combinações <i>Field To Filter</i>	Valor na caixa de combinações <i>Field To Filter</i> é alterado	Valor na caixa de combinações <i>Field To Filter</i> alterado com sucesso
	5º	No painel <i>Select Filters</i> , selecionar um valor na caixa de combinações <i>Orientation Operator</i>	Valor na caixa de combinações <i>Orientation Operator</i> é alterado	Valor na caixa de combinações <i>Orientation Operator</i> alterado com sucesso
	6º	No painel <i>Select Filters</i> , introduzir um valor na caixa de texto <i>Filter Value</i> de um tipo de dados diferente do filtro que foi escolhido no 4º passo	Valor na caixa de texto <i>Filter Value</i> é introduzido	Valor na caixa de texto <i>Filter Value</i> introduzido com sucesso
	7º	Pressionar o botão <i>Generate</i>	Surge uma mensagem de erro referindo que o valor introduzido é de um tipo diferente ao do filtro escolhido	Mensagem de erro visível

Tabela 5.3 - Plano de testes de sistema

5.4.4. Testes de Aceitação

Para realizar os testes de aceitação foi pedido a seis pessoas que respondessem a um questionário (Anexo B) no qual foi pedido ao utilizador para utilizar a ferramenta QG4B para responder com *queries* geradas pela aplicação a seis perguntas em que todas as funcionalidades para se gerar uma *query* foram requisitas. Por fim, foram realizadas outras oito perguntas sobre a qualidade e usabilidade do *software* que serviram para avaliar a experiência de utilização da ferramenta e identificar quais os problemas que esta tem em auxiliar da melhor maneira os seus utilizadores. Para além disso, foi também fornecida a ferramenta QG4B, o guia de utilização (Anexo A) e dois modelos relacionais iguais, mas de diferentes tipos (.pdm e .sql).

Na tabela 5.4 estão descritos os resultados das respostas às perguntas em que era necessário usar a ferramenta para responder às mesmas. É possível verificar que em 36 das respostas possíveis, 33 foram respondidas corretamente, o que equivale a uma taxa de sucesso de 91,7%.

Pergunta	Número de Respostas	
	Corretas	Erradas
Crie uma <i>query</i> com o preço por unidade mais elevado de um produto (<i>Products.UnitPrice</i>)	6	0
Crie uma <i>query</i> com os todos os nomes dos clientes (<i>Customers.Name</i>) da cidade de London (<i>Customers.City</i>) com um número de fax associado (<i>Customers.Fax</i>)	5	1
Crie uma <i>query</i> com todos os nomes dos produtos (<i>Products.ProductName</i>) e os seus preços por unidade (<i>Products.UnitPrice</i>) e que sejam fornecidos por fornecedor <i>Exotic Liquids</i> (<i>Suppliers.CompanyName</i>)	6	0
Crie uma <i>query</i> com todas as identificações das encomendas (<i>Orders.OrderID</i>), qual o primeiro e último nome do empregado associado a cada uma dessas encomendas (<i>Employees.FirstName</i> e <i>Employees.LastName</i>) e que o nome do expedidor seja <i>Federal Shipping</i> (<i>Shippers.CompanyName</i>)	6	0
Crie uma <i>query</i> com todos os nomes dos clientes (<i>Customers.CompanyName</i>) agrupados por cidade (<i>Customers.City</i>), ordenados descendentemente por região (<i>Customers.Region</i>) e que o país de despacho da encomenda (<i>Orders.ShipCountry</i>) seja <i>Portugal</i>	6	0
Crie uma <i>query</i> com todos os nomes dos fornecedores (<i>Suppliers.CompanyName</i>) e em que o nome da categoria do produto seja <i>Beverages</i> (<i>Categories.CategoryName</i>)	4	2

Tabela 5.4 - Respostas às perguntas sobre a utilização da ferramenta

Na tabela 5.5 é possível verificar o número de respostas às perguntas de qualidade e usabilidade da ferramenta, sendo que nesta segunda tabela, a escala vai de um a cinco, sendo 1 a nota mais reduzida e 5 a mais elevada. Através desta segunda parte do inquérito é possível apurar que apesar da ferramenta ter sido de fácil instalação para todos, vários utilizadores não acharam que a ferramenta lhes possa ser útil no futuro. Este número pode ser explicado devido ao alto conhecimento na área de base de dados demonstrado por estes utilizadores. Por fim, é mais uma vez de salientar que a maioria dos utilizadores ficou satisfeita ao utilizar a ferramenta, sendo que 66,7% dos utilizadores ficaram muito satisfeitos.

Pergunta	Número de Respostas				
	1	2	3	4	5
Achou a ferramenta de fácil instalação?					6
Achou a aplicação de fácil utilização?				1	5
Achou a interface <i>user-friendly</i> ?			2	2	2
Achou a aplicação intuitiva?				2	4
Caso tenha tido algum erro ao gerar uma <i>query</i> , conseguiu perceber com clareza o que devia corrigir?				3	1
Esta ferramenta poderá ser-lhe útil em alguma situação futura?	1		1	1	3
O manual de utilizador foi importante durante o uso da ferramenta?			2	1	3
Qual o seu grau de satisfação após usar esta ferramenta?		1	1		4

Tabela 5.5 - Respostas às perguntas sobre qualidade e usabilidade da ferramenta

6. CONCLUSÕES

6.1. SÍNTESE DO TRABALHO DESENVOLVIDO

O desenvolvimento deste projeto permitiu a criação de uma ferramenta capaz de gerar *queries* de consulta de dados para iniciação à prática de base de dados em três linguagens distintas (SQL, Álgebra Relacional e LINQ), podendo ser usada por utilizadores com pouco ou nenhum conhecimento nesta área. A validação das respostas por parte da ferramenta, verificando tanto o tipo de dados do campo como o seu tamanho limite, é um ponto chave na criação de *queries* para um iniciante na matéria pois diminui a probabilidade de erro, reduzindo assim o número de tentativas executadas por uma pessoa no servidor de base de dados.

O projeto foi iniciado com uma análise a ferramentas utilizadas no ensino de base de dados, sendo elaborado um estudo de quais as vantagens e desvantagens da utilização deste tipo de aplicações usadas no passado. De seguida, foram identificados os requisitos necessários para a boa execução da ferramenta e foram desenhados diagramas que exemplificam o funcionamento desta. Finalmente, foi desenvolvido o protótipo através de vários tipos de arquiteturas referentes à ferramenta e foram executados e documentados os testes à aplicação para verificar e controlar a qualidade desta.

Apesar de não ter sido testada em larga escala, foi possível verificar, através dos testes de aceitação realizados, que o uso deste tipo de aplicações para a aprendizagem de base de dados para um utilizador iniciante é intuitivo e eficaz. Também foi possível confirmar que, apesar das várias vantagens que o uso desta ferramenta pode ter no ensino na área de base de dados, há espaço para evoluir, nomeadamente na maneira como a ferramenta pode interagir com o utilizador durante a criação de uma *query*, minimizando assim o uso do guia de utilização e melhorando a experiência do utilizador com a ferramenta QG4B. Para além disso, a criação de um sistema de levantamento de *feedback* por parte da própria aplicação pode aumentar o número de opiniões sobre os pontos fortes e fracos que a ferramenta oferece, sendo mais fácil e mais exato definir o que necessita ser mantido ou melhorado.

6.2. RECOMENDAÇÕES PARA TRABALHOS FUTUROS

No futuro, a ferramenta poderá ser melhorada na perspetiva de reduzir a dependência do guia de utilização, ou seja, a ferramenta poderá explicar passo-a-passo quais podem ser as ações do utilizador. Por último, e de forma a que esta ferramenta possa ajudar o maior número de utilizadores possíveis, esta poderá pedir um *feedback* aos seus utilizadores no final do seu uso, sendo que depois seria realizada uma avaliação do mesmo e, caso fosse possível, melhorar a aplicação.

BIBLIOGRAFIA

- Abreu, L., & Morgado, P. (2009). *LINQ com C#*. Lousã: FCA - Editora de Informática, Lda.
- Brusilovsky, P., Sosnovsky, S., Uydelson, M. V., Lee, Danielle, H., Zadorozhny, V., & Zhou, X. (2010). Learning SQL Programming with Interactive Tools: From Integration to Personalization. *ACM Transactions on Computing Education*, 9(4), 367–376. <https://doi.org/10.1145.1656255.1656257>
- ClydeBank Technology. (2015). *SQL: QuickStart Guide - The Simplified Beginner's Guide To SQL* (1st ed.). ClydeBank Media LLC.
- Elmasri, R., & Navathe, S. B. (2010). *Fundamentals of Database Systems* (6th ed.). Pearson Education, Inc.
- Garcia-Molina, H., Ullman, J. D., & Widom, J. (2008). *Database Systems: The Complete Book. Education* (2nd ed.). New Jersey: Pearson Prentice Hall. <https://doi.org/10.1145/253262.253287>
- Gouveia, F. (2014). *Fundamentos de Bases de Dados*. Lisboa: FCA - Editora de Informática, Lda.
- Guia, T. F. G., Rodrigo, M. M. T., Dagami, M. M. C., Sugay, J. O., Macam, F. J. F., & Mitrovic, A. (2011). Modeling the Affective States of Students using SQL- Tutor, 2–3.
- Kriegel, A. (2011). *Discovering SQL: A Hands-On Guide for Beginners* (1st ed.). Wrox.
- Melton, J. (1998). Database Language SQL. In *Handbook on Architectures of Information Systems* (pp. 103–128). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-03526-9_5
- Mitrovic, A. (1998). Learning SQL with a computerized tutor. *ACM SIGCSE Bulletin*. <https://doi.org/10.1145/274790.274318>
- Mitrovic, A., & Martin, B. (2000). Evaluating the effectiveness of feedback in SQL-Tutor. In *Proceedings - International Workshop on Advanced Learning Technologies: Advanced Learning Technology: Design and Development Issues, IWALT 2000* (pp. 143–144). <https://doi.org/10.1109/IWALT.2000.890591>
- Mitrovic, A., & Ohlsson, S. (1999). Evaluation of a constraint-based tutor for a database language. *International Journal of Artificial Intelligence in Education*, 10, 238–256. Retrieved from <http://www.uic.edu/depts/psch/ohlson-1.html>
- Pialorsi, P., & Russo, M. (2008). *Programming Microsoft LINQ*. Redmond, Washington: Microsoft Press.
- Roberts, J. (2015). *LINQ Succinctly*. Morrisville: Syncfusion Inc.
- Russell, G., & Cumming, A. (2004). Improving the Student Learning Experience for SQL using Automatic Marking. In *Cognition and Exploratory Learning in Digital Age* (pp. 281–288). Retrieved from <http://researchrepository.napier.ac.uk/1774/1/Russell.pdf>
- Sadiq, S. W., Orłowska, M. E., Sadiq, W., & Lin, J. (2004). SQLator: An Online SQL Learning Workbench. *Proceedings of the Ninth Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, 223–227. <https://doi.org/10.1145/1007996.1008055>

Santos, V. (2012). *System Analysis*. Lisboa.

Sharma, N., Perniu, L., Chong, R. F., Iyer, A., Nandan, C., Mitea, A., ... Danubianu, M. (2010). *Database Fundamentals. IBM Corporation 2010* (1st ed.). Markham.

Stephens, R., Jones, A. D., & Plew, R. (2011). *Sams Teach Yourself SQL in 24 Hours* (5th ed.). Sams.

Sumathi, S., & Esakkirajan, S. (2007). *Fundamentals of Relational Database Management Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg.

Tale, S. (2016). *SQL: The Ultimate Beginners Guide: Learn SQL Today*.

Vítor, A., & Santos, V. (2016). Construtor Automático de Queries a partir de um Modelo Relacional. *CAPSI*, 1.

ANEXOS

Anexo A - Guia de Utilização

Guia de Utilização

Instalação da Ferramenta QG4B

Para a correta execução da ferramenta, é necessário que o ambiente cumpra os seguintes requisitos:

Sistema Operativo	Windows XP ou superior, Linux ou MacOS
Plataforma .NET Framework	.NET framework 3.5 ou superior

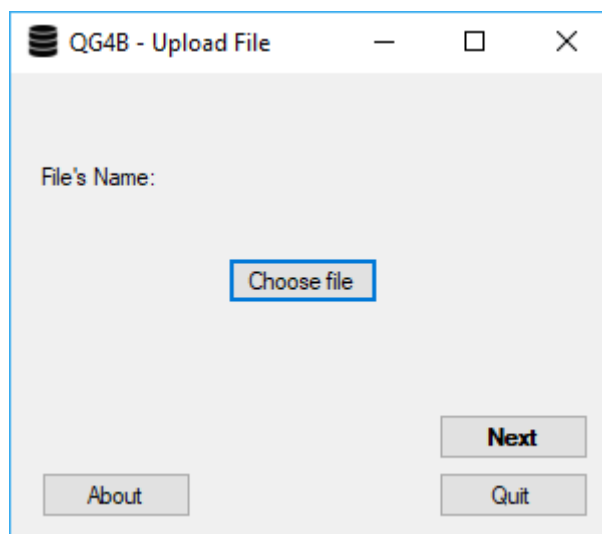
Antes de iniciar a aplicação, os ficheiros contidos no ficheiro *QG4B.zip* devem ser extraídos para uma pasta à escolha.

Como inicializar a ferramenta QG4B?

1. Abrir a pasta onde foram extraídos os ficheiros incluídos no ficheiro *QG4B.zip*. A pasta deve conter os seguintes ficheiros:



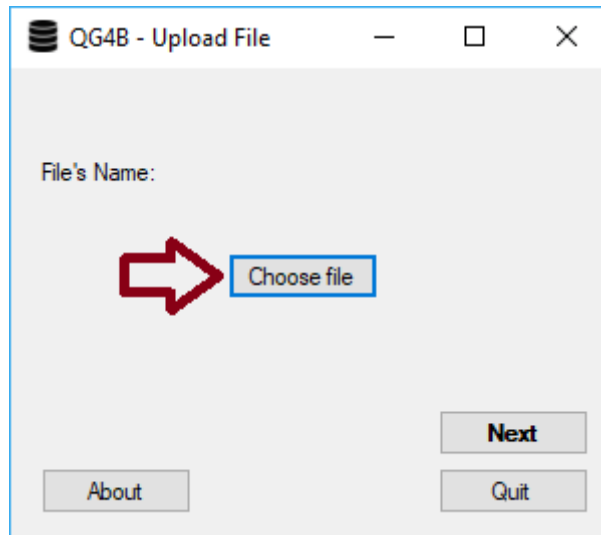
2. Abrir o executável da ferramenta *QG4B.exe*, a qual deve inicializar uma janela com o seguinte formato:



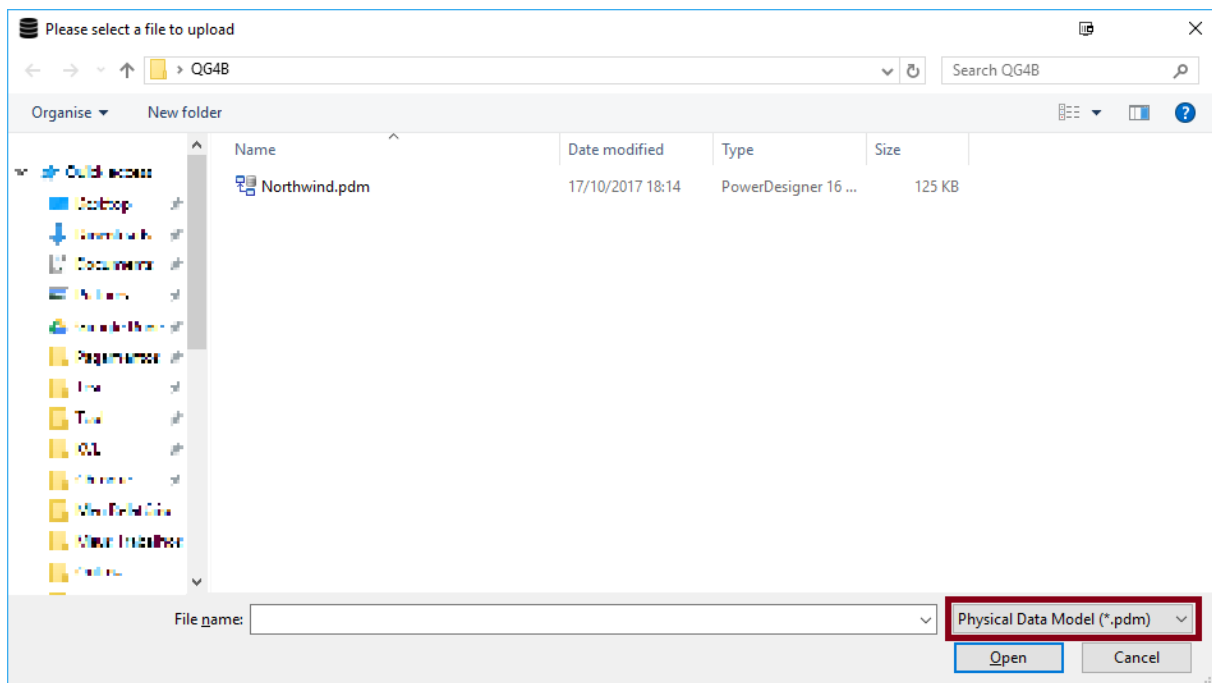
Como fazer *upload* de um modelo relacional?

Para fazer o *upload* do modelo relacional, o utilizador deve já ter inicializado a aplicação e, de seguida, deve realizar os próximos passos:

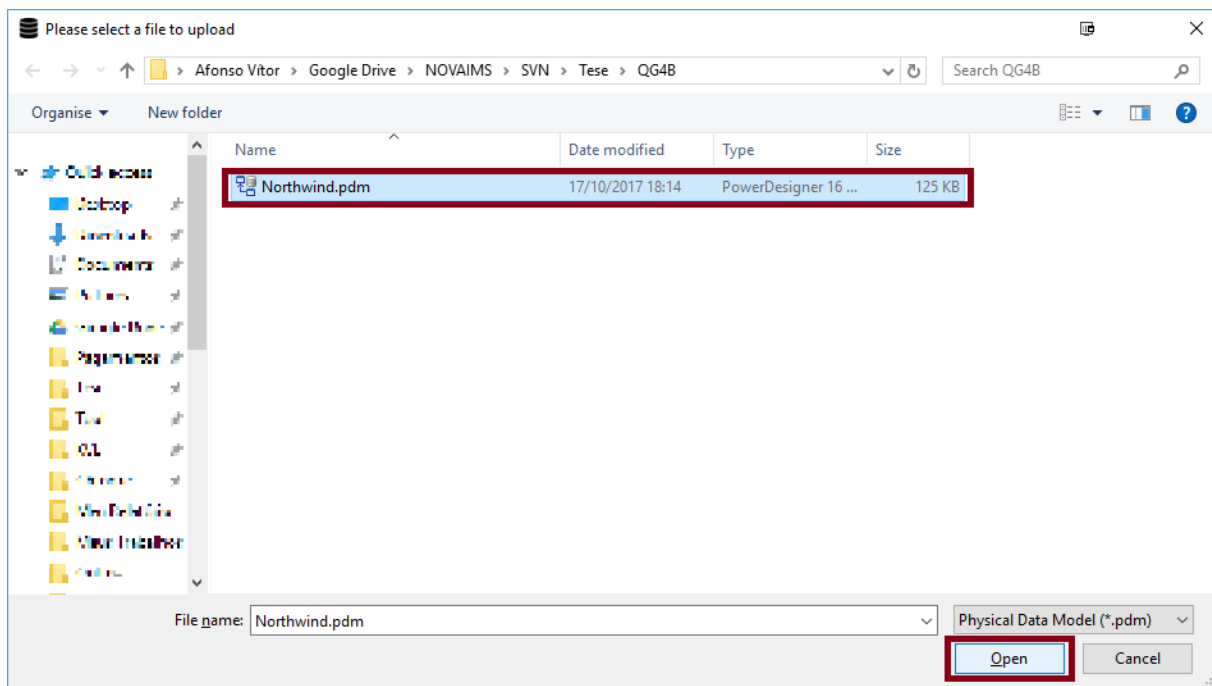
1. Pressionar o botão *Choose File*



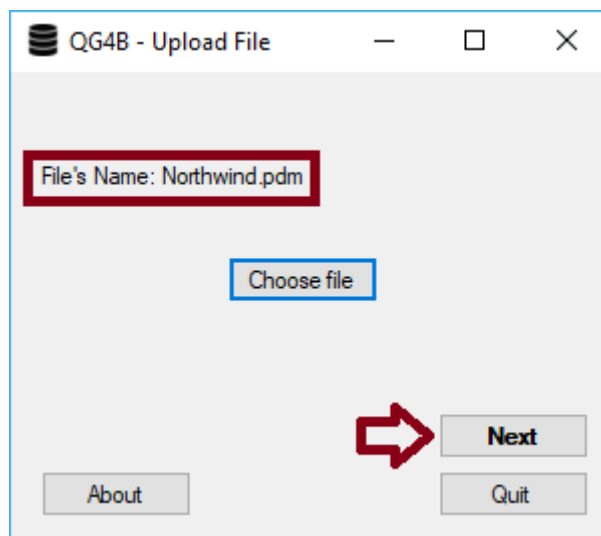
2. Na nova janela, verificar qual o tipo de modelo relacional que pretende inserir (.pdm ou .sql) mudando a opção assinalada com um retângulo vermelho



3. Escolher o ficheiro que pretende usar como modelo relacional e pressionar o botão *Open*



4. Caso o ficheiro tenha sido inserido com sucesso, o nome deste será visível no campo indicado com um retângulo vermelho. De seguida, deve pressionar o botão *Next*



5. A ferramenta irá então verificar o ficheiro inserido pelo utilizador e caso exista algum erro, este deverá ser apresentado numa caixa de texto com os detalhes deste. No entanto, se tudo estiver correto, a seguinte janela deverá ser aberta:

The screenshot shows a software window titled "QG4B - Generate Query". The window is divided into three main sections for configuring a query:

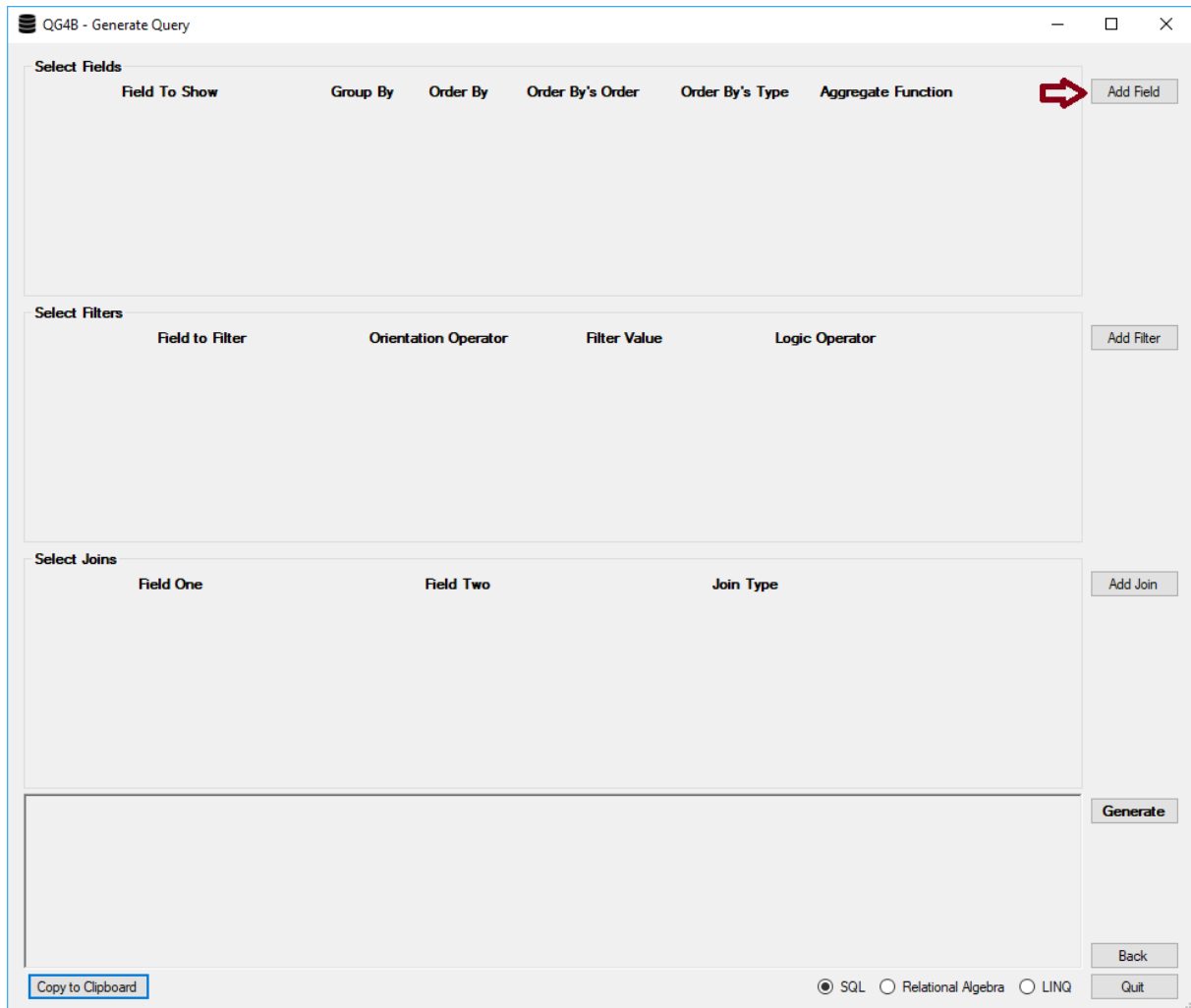
- Select Fields:** A table with columns: Field To Show, Group By, Order By, Order By's Order, Order By's Type, and Aggregate Function. An "Add Field" button is on the right.
- Select Filters:** A table with columns: Field to Filter, Orientation Operator, Filter Value, and Logic Operator. An "Add Filter" button is on the right.
- Select Joins:** A table with columns: Field One, Field Two, and Join Type. An "Add Join" button is on the right.

Below these sections is a large empty area for the generated query. At the bottom of the window, there is a "Generate" button, a "Copy to Clipboard" button, and radio buttons for "SQL" (selected), "Relational Algebra", and "LING". There are also "Back" and "Quit" buttons.

Como gerar uma *query*

Para gerar uma *query*, o utilizador deve já ter realizado o *upload* do modelo relacional. Posteriormente, deverá realizar os seguintes passos:

1. Adicionar um campo para mostrar, pressionando o botão *Add Field*



The screenshot displays the 'QG4B - Generate Query' application window. It features a structured interface for building a query. The 'Select Fields' section is the primary focus, containing columns for 'Field To Show', 'Group By', 'Order By', 'Order By's Order', 'Order By's Type', and 'Aggregate Function'. A red arrow points to the 'Add Field' button on the right side of this section. Below it are 'Select Filters' and 'Select Joins' sections, each with their respective columns and 'Add Filter' or 'Add Join' buttons. At the bottom right, there are 'Generate', 'Back', and 'Quit' buttons. The 'Quit' button is accompanied by radio buttons for selecting the query language: SQL (selected), Relational Algebra, and LINQ. A 'Copy to Clipboard' button is located at the bottom left.

- No campo adicionado (assinalado com um retângulo vermelho), deve alterar os valores das caixas criadas da maneira que pretende

QG4B - Generate Query

Select Fields

Field To Show	Group By	Order By	Order By's Order	Order By's Type	Aggregate Function
<input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>			<input type="text"/> <input type="button" value="Remove"/>

Select Filters

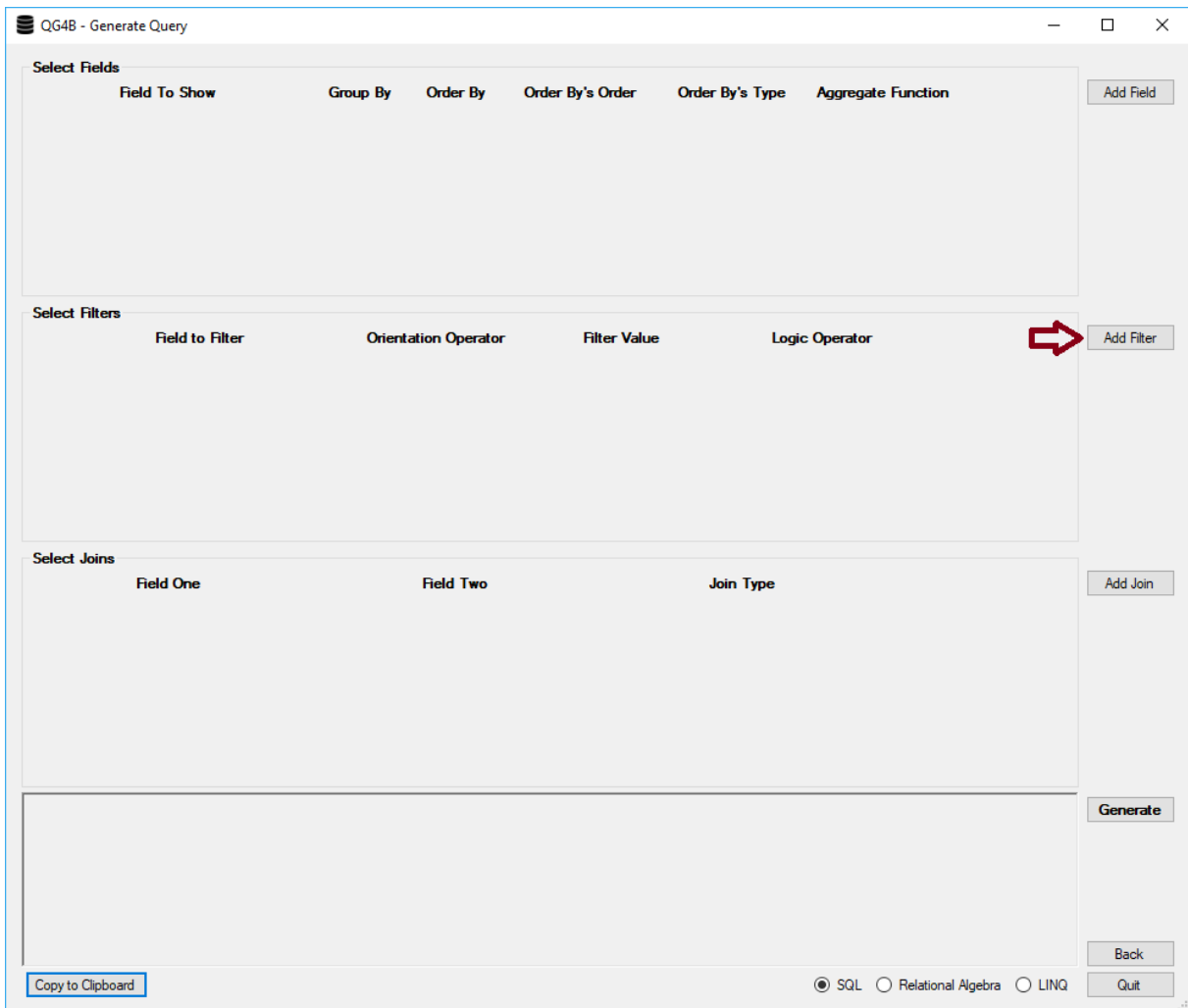
Field to Filter	Orientation Operator	Filter Value	Logic Operator
<input type="text"/>			

Select Joins

Field One	Field Two	Join Type
<input type="text"/>	<input type="text"/>	

SQL Relational Algebra LING

3. Caso pretenda adicionar um filtro, deve pressionar o botão *Add Filter*



- No filtro adicionado (assinalado com um retângulo vermelho), deve alterar os valores das caixas criadas da maneira que pretende

QG4B - Generate Query

Select Fields

Field To Show	Group By	Order By	Order By's Order	Order By's Type	Aggregate Function
---------------	----------	----------	------------------	-----------------	--------------------

Add Field

Select Filters

Field to Filter	Orientation Operator	Filter Value	Logic Operator
()

Add Filter

Select Joins

Field One	Field Two	Join Type
-----------	-----------	-----------

Add Join

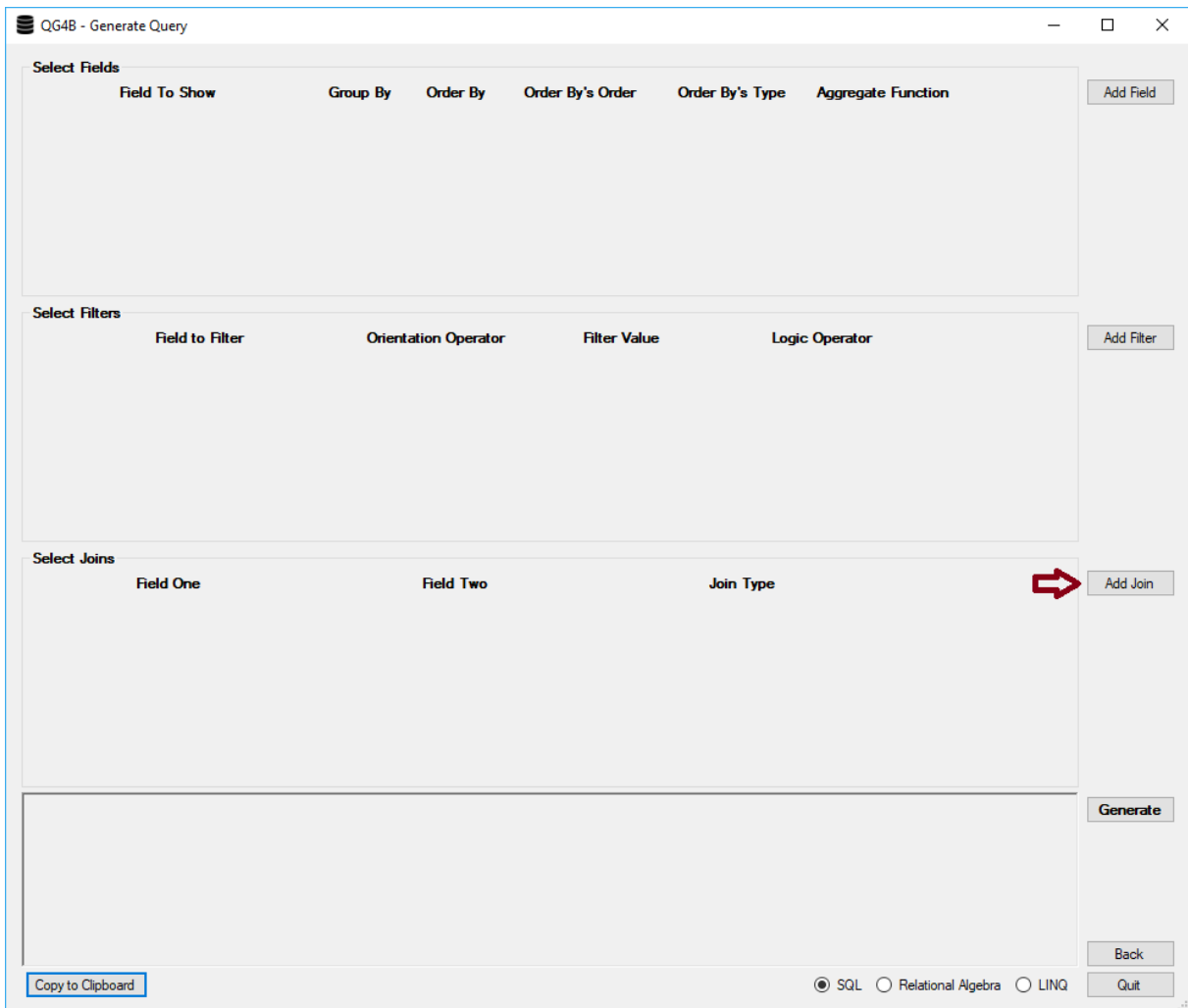
Generate

Copy to Clipboard

SQL Relational Algebra LING

Back Quit

5. Caso pretenda adicionar um *join*, deve pressionar o botão *Add Join*



Nota: Se existe uma relação entre duas tabelas no modelo relacional, então será criado automaticamente um *join* entre essas tabelas caso as mesmas sejam utilizadas na ferramenta.

6. No *join* adicionado (assinalado com um retângulo vermelho), deve alterar os valores das caixas criadas da maneira que pretende

QG4B - Generate Query

Select Fields

Field To Show	Group By	Order By	Order By's Order	Order By's Type	Aggregate Function
---------------	----------	----------	------------------	-----------------	--------------------

Add Field

Select Filters

Field to Filter	Orientation Operator	Filter Value	Logic Operator
-----------------	----------------------	--------------	----------------

Add Filter

Select Joins

Field One	Field Two	Join Type	
		Inner Join	Remove

Add Join

Generate

Back

Copy to Clipboard

SQL Relational Algebra LING

Quit

7. Selecionar qual a linguagem em que deseja visualizar a sua *query* (assinalada com um retângulo vermelho)

QG4B - Generate Query

Select Fields

Field To Show	Group By	Order By	Order By's Order	Order By's Type	Aggregate Function
---------------	----------	----------	------------------	-----------------	--------------------

Add Field

Select Filters

Field to Filter	Orientation Operator	Filter Value	Logic Operator
-----------------	----------------------	--------------	----------------

Add Filter

Select Joins

Field One	Field Two	Join Type
-----------	-----------	-----------

Add Join

Generate

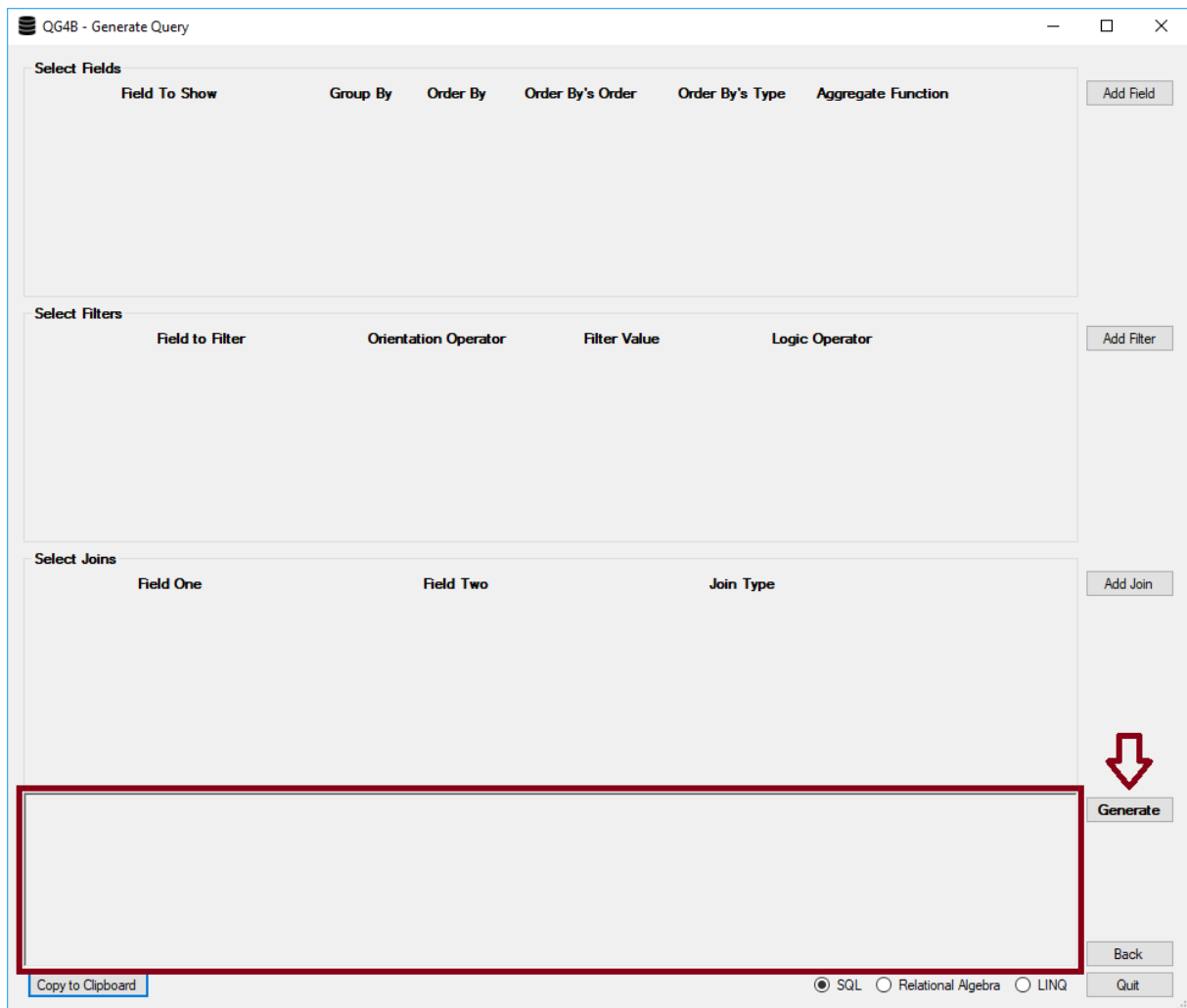
Copy to Clipboard

SQL Relational Algebra LING

Back

Quit

- Depois de preencher todos os campos que deseja e escolher a linguagem pretendida, deve pressionar o botão *Generate* e caso não exista nenhum erro, a *query* será gerada na caixa de texto no fundo da janela (assinalada com um retângulo vermelho)



Anexo B - Questionário de Satisfação e Qualidade

Questionário

O seguinte inquérito irá auxiliar a compreender se o desenvolvimento da ferramenta QG4B contribuiu para a aprendizagem inicial em base de dados. O questionário é dividido em duas partes, uma primeira parte para testar o funcionamento da ferramenta, e a segunda parte para verificar a qualidade e usabilidade desta. Na última página deste questionário encontra-se um diagrama dos ficheiros dos modelos relacionais dado, o qual pode ser usado para compreender melhor como as tabelas se relacionam entre elas e quais os seus campos.

Idade: _____

Nível de conhecimentos em base de dados:

- Nenhum
- Básico
- Intermédio
- Avançado

Recorrendo à ferramenta QG4B e com o apoio do guia de utilização, responda às seguintes questões:

1. Crie uma *query* com o preço por unidade mais elevado de um produto (*Products.UnitPrice*)

Modelo relacional do tipo .pdm
Modelo relacional do tipo .sql

2. Crie uma query com os todos os nomes dos clientes (*Customers.Name*) da cidade de London (*Customers.City*) com numero de fax associado (*Customers.Fax*)

Modelo relacional do tipo .pdm
Modelo relacional do tipo .sql

3. Crie uma *query* com todos os nomes dos produtos (*Products.ProductName*) e os seus preços por unidade (*Products.UnitPrice*) e que sejam fornecidos pelo fornecedor chamado *Exotic Liquids* (*Suppliers.CompanyName*)

Modelo relacional do tipo .pdm
Modelo relacional do tipo .sql

4. Crie uma *query* com todas as identificações das encomendas (*Orders.OrderID*), qual o primeiro e último nome do empregado associado a cada uma dessas encomendas (*Employees.FirstName* e *Employees.LastName*) e que o nome do expedidor seja *Federal Shipping* (*Shippers.CompanyName*)

Modelo relacional do tipo .pdm
Modelo relacional do tipo .sql

5. Crie uma *query* com todos os nomes dos clientes (*Customers.CompanyName*) agrupados por cidade (*Customers.City*), ordenados descendentemente por região (*Customers.Region*) e que o país de despacho da encomenda (*Orders.ShipCountry*) seja *Portugal*

Modelo relacional do tipo .pdm
Modelo relacional do tipo .sql

6. Crie uma *query* com todos os nomes dos fornecedores (*Suppliers.CompanyName*) e em que o nome da categoria do produto seja Beverages (*Categories.CategoryName*)

Dica: Verifique o modelo relacional existente na última página deste questionário de forma a criar corretamente os *joins* entre as tabelas *Suppliers* e *Categories*

Modelo relacional do tipo .pdm
Modelo relacional do tipo .sql

Numa escala de um a cinco, sendo 1 a nota mais reduzida e 5 a mais elevada, responda com uma cruz (X) às seguintes perguntas:

Pergunta	Avaliação				
	1	2	3	4	5
Achou a ferramenta de fácil instalação?					
Achou a aplicação de fácil utilização?					
Achou a interface <i>user-friendly</i> ?					
Achou a aplicação intuitiva?					
Caso tenha tido algum erro ao gerar uma <i>query</i> , conseguiu perceber com clareza o que devia corrigir? (Ignore esta questão caso não tenha tido nenhum erro)					
Esta ferramenta poderá ser-lhe útil em alguma situação futura?					
O manual de utilizador foi importante durante o uso da ferramenta?					
Qual o seu grau de satisfação após usar esta ferramenta?					

Obrigado pela colaboração! 😊

