



**Universidade do Minho**

Mestrado Integrado em Engenharia Informática

Licenciatura em Ciências da Computação

## **Unidade Curricular de Bases de Dados**

Ano Lectivo de 2018/2019

### **Arquivo de Revistas da Ritinha**

**Bárbara Andreia Cardoso Ferreira A80453**

**José Augusto Ferreira Alves A82885**

**Miriam Miranda Pinto A42040**

**Rui Filipe Brito Azevedo A80789**

Novembro, 2018

# BD

Data de Recepção	
Responsável	
Avaliação	
Observações	

## Arquivo de Revistas da Ritinha

**Bárbara Andreia Cardoso Ferreira A80453**

**José Augusto Ferreira Alves A82885**

**Miriam Miranda Pinto A42040**

**Rui Filipe Brito Azevedo A80789**

Novembro, 2018

## Resumo

O presente relatório tem por objetivo descrever as várias fases que foram abordadas para o desenvolvimento de uma base de dados relacional. O objetivo desta base de dados é o armazenamento de informação digital, ou seja, o armazenamento digital de informação relativa a revistas, facilitando a procura e atualização do registo das mesmas.

Na primeira fase é feita a descrição do contexto do problema, fundamentada a necessidade da base de dados e analisada a viabilidade do processo. De seguida foi feito o levantamento e análise dos requisitos junto do cliente, a amável Dona Ritinha, que posteriormente foram divididos em três categorias: descrição, exploração e controlo. Após a sua análise foram redigidos numa linguagem mais técnica que por sua vez levou à identificação e caracterização das entidades, atributos e relacionamentos necessários para a passagem à fase seguinte, ou seja, a construção do modelo conceptual.

Com o modelo conceptual estabelecido foi gerado o modelo lógico onde se averiguou a normalização do mesmo e a sua capacidade de resposta às interrogações do cliente. Nesta fase do processo é preciso ter em atenção ao conjunto de operações que só podem ser executadas como um todo, no caso de uma falhar não se executa nenhuma. Este conjunto de operações dão origem a transações.

Após a validação do modelo lógico é gerado o modelo físico recorrendo à ferramenta *Forward Engineer* presente no *MySQL Workbench*. Nesta fase as interrogações do cliente e as transações passam a estar descritas em código *SQL*, são escolhidos os índices, é feita uma estimativa do espaço em disco da base de dados e da sua taxa de crescimento anual e são definidas e caracterizadas as vistas de utilização do *SQL*.

Após a concretização de todas estas fases pensamos estar perante uma base de dados relacional operacional, consistente, útil e apropriada para o projeto da nossa cliente.

**Área de Aplicação:** Criação de Sistemas de Bases de Dados

**Palavras-Chave:** Modelo Conceptual, Modelo Lógico, Base de Dados Relacional, *MySQL*, Entidade, Relacionamento, Atributo, Chave Primária, Chave Estrangeira

# Índice

Resumo .....	i
Índice de Figuras .....	v
Índice de Tabelas.....	vi
<b>1. Definição do sistema .....</b>	<b>1</b>
1.1. Contexto de aplicação do sistema .....	1
1.2. Fundamentação da implementação da base de dados .....	1
1.3. Análise da viabilidade do processo .....	2
1.4. Estrutura do Relatório .....	2
<b>2. Levantamento e Análise de Requisitos .....</b>	<b>3</b>
2.1. Método de levantamento e de análise de requisitos adotado .....	3
2.2. Requisitos levantados .....	3
2.2.1. Requisitos de Descrição .....	4
2.2.2. Requisitos de Exploração .....	4
2.2.3. Requisitos de Controlo .....	5
2.3. Análise geral dos requisitos .....	5
<b>3. Modelação conceptual.....</b>	<b>6</b>
3.1. Apresentação da abordagem de modelação realizada .....	6
3.2. Identificação e caracterização das entidades .....	6
3.3. Identificação e caracterização dos relacionamentos.....	7
3.4. Identificação e caracterização das associações dos atributos com as entidades e relacionamentos .....	10
3.5. Detalhe ou generalização de entidades .....	12
3.6. Apresentação e explicação do diagrama ER .....	12
3.7. Validação do modelo de dados com o utilizador .....	13
<b>4. Modelação Lógica .....</b>	<b>14</b>
4.1. Construção e validação do modelo de dados lógico .....	14
4.2. Desenho do modelo lógico.....	15
4.3. Validação do Modelo através da Normalização.....	16
4.3.1. Primeira forma normal (1NF) .....	16
4.3.2. Segunda forma normal (2NF) .....	16
4.3.3. Terceira forma normal (3NF) .....	16

<b>4.4. Validação do modelo com interrogações do utilizador .....</b>	<b>17</b>
4.4.1. Permitir a consulta das informações de uma empresa .....	17
4.4.2. Listar os contactos de uma dada empresa .....	17
4.4.3. Permitir a consulta das informações de um anúncio .....	17
4.4.4. Permitir a consulta das informações de um artigo.....	17
4.4.5. Listar todos as revistas de uma dada empresa.....	18
4.4.6. Listar todos os artigos contidos numa revista .....	18
4.4.7. Listar todos os anúncios contidos numa revista.....	18
4.4.8. Listar todos os artigos escritos por um dado escritor.....	18
4.4.9. Listar todas as revistas editadas por um dado editor .....	18
4.4.10. Identificar a quantidade de revistas visualizadas de uma empresa.....	19
4.4.11. Listar todos os escritores que têm pelo menos um artigo escrito numa determinada revista .....	19
4.4.12. Listar todas as revistas de uma dada empresa numa determinada data ou intervalo de datas .....	19
4.4.13. Listar todos os escritores que já contribuíram para a escrita de revistas de uma dada empresa .....	20
4.4.14. Listar todas as revistas em que aparecerem os anúncios de uma determinada empresa .....	20
4.4.15. Listar as N revistas com maior número de consultas.....	20
4.4.16. Listar as categorias por frequências relativas de consultas .....	21
<b>4.5. Validação do modelo com as transações estabelecidas .....</b>	<b>21</b>
<b>4.6. Revisão do modelo lógico com o utilizador .....</b>	<b>22</b>
<b>5. Implementação Física .....</b>	<b>23</b>
5.1. Seleção do sistema de gestão de bases de dados .....	23
5.2. Tradução do esquema lógico para o sistema de gestão de bases de dados escolhido em SQL .....	23
5.3. Tradução das interrogações do utilizador para SQL (alguns exemplos).....	23
5.3.1. Listar todos os escritores que já contribuíram para a escrita de revistas de uma dada empresa .....	24
5.3.2. Calcular o número total de consultas das revistas de uma empresa .....	24
5.3.3. Listar as categorias por frequências relativas de consultas.....	25
5.4. Tradução das transações estabelecidas para SQL (alguns exemplos) .....	25
5.4.1. Inserir uma empresa.....	26

5.4.2. Inserir um artigo .....	27
<b>5.5. Escolha, definição e caracterização de índices em SQL (alguns exemplos) .....</b>	<b>28</b>
<b>5.6. Estimativa do espaço em disco da base de dados e taxa de crescimento anual .....</b>	<b>28</b>
5.6.1. Estimativa do espaço em disco da base de dados .....	28
5.6.2. Taxa de crescimento anual .....	31
<b>5.7. Definição e caracterização das vistas de utilização do SQL (alguns exemplos) .....</b>	<b>31</b>
5.7.1. Vista da Tabela Escritor .....	31
5.7.2. Vista das empresas e respetivas visualizações das suas revistas.....	32
5.7.3. Vista do <i>Top 5</i> de escritores .....	32
<b>5.8. Definição e caracterização dos mecanismos de segurança em SQL (alguns exemplos)</b>	<b>33</b>
<b>5.9. Revisão do sistema implementado com o utilizador .....</b>	<b>34</b>
<b>6. Conclusões e Trabalho Futuro .....</b>	<b>35</b>
<b>Referências .....</b>	<b>36</b>
<b>Lista de Siglas e Acrónimos.....</b>	<b>42</b>

# Índice de Figuras

FIGURA 1 - RELACIONAMENTO REVISTA – ANÚNCIO.	7
FIGURA 2 - RELACIONAMENTO REVISTA – ARTIGO.	8
FIGURA 3 - RELACIONAMENTO REVISTA – EMPRESA.	8
FIGURA 4 - RELACIONAMENTO REVISTA – EDITOR.	8
FIGURA 5 - RELACIONAMENTO EMPRESA – ANÚNCIO.	9
FIGURA 6 - RELACIONAMENTO ARTIGO – ESCRITOR.	9
FIGURA 7 - MODELO CONCEPTUAL	12
FIGURA 8 - MODELO LÓGICO	15
FIGURA 9 - TRADUÇÃO DO RE5 PARA <i>SQL</i>	24
FIGURA 10 - TRADUÇÃO DO RE10 PARA <i>SQL</i>	24
FIGURA 11 - TRADUÇÃO DO RE15 PARA <i>SQL</i>	25
FIGURA 12 - TRADUÇÃO DA TRANSAÇÃO DE INSERÇÃO DE UMA EMPRESA EM <i>SQL</i>	26
FIGURA 13 - TRADUÇÃO DE TRANSAÇÃO DA INSERÇÃO DE UM ARTIGO EM <i>SQL</i>	27
FIGURA 14 - VISTA SOBRE A TABELA ESCRITOR	31
FIGURA 15 - VISTA DAS EMPRESAS E RESPETIVAS VISUALIZAÇÕES TOTAIS DAS SUAS REVISTAS	32
FIGURA 16 – VISTA DO <i>TOP 5</i> DE ESCRITORES	32
FIGURA 17 - CÓDIGO <i>SQL</i> PARA A CRIAÇÃO DA VISTA DE UTILIZAÇÃO E RESPETIVAS PERMISSÕES DO UTILIZADOR ADMIN	33
FIGURA 18 - CÓDIGO <i>SQL</i> PARA A CRIAÇÃO DA VISTA DE UTILIZAÇÃO E RESPETIVAS PERMISSÕES DO UTILIZADOR APP	33
FIGURA 19 - POVOAMENTO DA TABELA EMPRESA	37
FIGURA 20 - POVOAMENTO DA TABELA CONTACTOS	37
FIGURA 21 - POVOAMENTO DA TABELA ANÚNCIO	38
FIGURA 22 - POVOAMENTO DA TABELA EDITOR	38
FIGURA 23 - POVOAMENTO DA TABELA REVISTA	39
FIGURA 24 - POVOAMENTO DA TABELA ARTIGO	39
FIGURA 25 - POVOAMENTO DA TABELA ESCRITOR	40
FIGURA 26 - POVOAMENTO DA TABELA ANUNCIOREVISTA	40
FIGURA 27 - POVOAMENTO DA TABELA ESCRITORARTIGO	41

# Índice de Tabelas

TABELA 1 - DICIONÁRIO DE DADOS DAS ENTIDADES	7
TABELA 2 - DICIONÁRIO DE DADOS DOS RELACIONAMENTOS.	9
TABELA 3 - IDENTIFICAÇÃO DOS ATRIBUTOS	12
TABELA 4 - ESPAÇO POR ENTRADA DA TABELA REVISTA	28
TABELA 5 - ESPAÇO POR ENTRADA DA TABELA EMPRESA	29
TABELA 6 - ESPAÇO POR ENTRADA DA TABELA CONTACTO	29
TABELA 7 - ESPAÇO POR ENTRADA DA TABELA ARTIGO	29
TABELA 8 - ESPAÇO POR ENTRADA DA TABELA ESCRITOR	29
TABELA 9 - ESPAÇO POR ENTRADA DA TABELA ANUNCIO	30
TABELA 10 - ESPAÇO POR ENTRADA DA TABELA EDITOR	30
TABELA 11 - ESPAÇO POR ENTRADA DA TABELA ANUNCIOREVISTA	30
TABELA 12 - ESPAÇO POR ENTRADA DA TABELA ARTIGOESCRITOR	30



# **1. Definição do sistema**

Nesta secção será feita uma contextualização e fundamentação da implementação da base de dados, tendo em mente a viabilidade do processo. No final é feito um pequeno resumo da estrutura geral do relatório.

## **1.1. Contexto de aplicação do sistema**

O crescimento da informação acompanha a constante mudança de um mundo turbulento e inquieto e o formato em que esta é apresentada pode surgir em diversos tipos, tais como digital ou em papel.

Com o avanço das tecnologias surgiu um enorme crescimento no ‘consumo’ de informação em formato digital, que apresenta um vasto leque de artigos variados, redigidos por diversos escritores. Toda essa informação encontra-se dispersa por diferentes revistas.

Neste contexto é apropriado organizar os dados relevantes das revistas, tais como dados acerca dos artigos e anúncios que estas contêm, dos editores responsáveis pela edição das várias revistas, dos escritores responsáveis pelos artigos e das empresas, que podem ser produtoras de anúncios e/ou produtoras de revistas, de modo a fornecer as informações relativas às preferências dos leitores.

## **1.2. Fundamentação da implementação da base de dados**

A realização deste estudo é feita por solicitação da Dona Ritinha, uma senhora simpática e singela que requisitou os serviços de quatro heroicos futuros engenheiros da Universidade do Minho para que organizassem os dados relativamente às revistas vendidas da sua livraria digital.

A Dona Ritinha tinha por objetivo obter informações acerca das preferências dos leitores e com base nas mesmas ter um arquivo que facilitasse o acesso a informações gerais e específicas relativamente às revistas. Perante esta necessidade decidiu organizar-se os dados de forma a que seja possível consultar detalhadamente revistas armazenados digitalmente, os seus editores, os seus artigos, anúncios, empresas e escritores.

As revistas são caracterizadas por nome, categoria, edição e n.º de consultas. O editor de uma ou várias revistas possui um nome. Os artigos possuem data, título, corpo, n.º de consultas e categoria. Um artigo pode ser redigido por um ou mais escritores os quais são caracterizados por nome. Os anúncios têm categoria, título, conteúdo, n.º de consultas e contato. Uma empresa está caracterizada por nome, morada, contato e quantidade de revistas. Cada empresa pode ter uma ou mais revistas associadas e pode publicar anúncios na revista.

### **1.3. Análise da viabilidade do processo**

Este sistema foi pensado com o intuito de satisfazer dois tipos de utilizadores: as pessoas que estão constantemente em busca de informação digital e a Dona Ritinha que pretende melhorar o seu serviço.

Como sabemos, atualmente, cada vez mais o ser humano sente a necessidade de estar informado acerca do que se passa à sua volta. As pessoas procuram respostas, fontes de inspiração ou uma mera distração na Web e nem sempre é fácil obter o que realmente se pretende.

Perante este cenário pensamos ser adequado que a informação das revistas da Dona Ritinha esteja organizada de um modo simples e completo permitindo uma procura de informação cómoda e pouco dispendiosa. Apesar de uma revista impressa ser uma grande 'reliquia', nos dias correntes, poucas são as pessoas que se deslocam a um espaço físico para a adquirir e folhear em procura de um dado assunto. O tempo tornou-se algo escasso e tudo o que pode ser feito num clique de rato faz com que se descarte qualquer outra hipótese mais dispendiosa em tempo e dinheiro.

Este sistema de base de dados visa auxiliar qualquer tipo de aplicação que seja utilizada pelos dois utilizadores referidos no início desta secção e satisfazer as suas necessidades neste contexto.

### **1.4. Estrutura do Relatório**

Após a definição do sistema feita ao longo desta secção, será apresentada a metodologia, o levantamento e a análise dos diversos tipos de requisitos ao longo do capítulo 2. No capítulo 3 será exposto o modelo conceptual elaborado com base nos dados recolhidos e expostos até então. O modelo lógico surgirá no capítulo 4, onde será explicada a sua elaboração abordada a sua validade. No penúltimo capítulo surge o modelo físico com recurso ao *MySQL Workbench* e as interrogações do cliente assim como as transações passam a estar descritas em código *SQL*. No último capítulo será feita uma reflexão relativamente ao trabalho elaborado, ao que se pode melhorar e o que se gostaria de fazer no futuro.

## **2. Levantamento e Análise de Requisitos**

Nesta secção será abordado o método de levantamento e de análise dos vários requisitos levantados e no final será feita uma análise geral dos mesmos.

### **2.1. Método de levantamento e de análise de requisitos adotado**

Quando um profissional se encontra perante um projeto/desafio, o primeiro passo a ser executado é compreender o que o cliente que solicitou o mesmo pretende. Com esta ideia em mente o nosso primeiro intuito foi abordar a Dona Ritinha através de uma entrevista para o levantamento de requisitos. Com essa entrevista conseguimos determinar que a Dona Ritinha pretende um site onde os seus clientes possam aceder a informações relativas a diversas revistas, tais como artigos e respetivos escritores, editor, anúncios publicados e respetivas empresas anunciantes ou a própria empresa da revista em causa.

A Dona Ritinha quer armazenar, relativamente a cada revista, o nome, a edição, a categoria e o número de consultas. Para cada artigo, que pode ser redigido por um ou mais escritores, ela pretende que seja armazenada a data, a categoria, o número de consultas, o corpo e o título. Relativamente a cada escritor dos artigos e editores de revistas apenas pretende guardar o seu nome como informação.

No decorrer da entrevista a Dona Ritinha frisou que quer manter um registo dos anúncios que são publicados nas várias revistas e quer manter registado a sua categoria, o contato das empresas anunciantes, o conteúdo, o título e o número de consultas. A estes anúncios quer também que estejam associadas as empresas que os publicam e que seja possível aceder ao seu nome, à sua morada e contatos. A Dona Ritinha manifestou também que quer manter o registo das empresas responsáveis pelas revistas tendo disponível no site as informações sobre o seu nome, a morada, os seus contatos e a quantidade de revistas produzidas por elas.

### **2.2. Requisitos levantados**

De seguida serão apresentados os requisitos que foram levantados através da entrevista à nossa cliente e que deverão ser suportados pelo sistema de base de dados a desenvolver para cada um dos diferentes tipos (exploração e administração).

### **2.2.1. Requisitos de Descrição**

RD1: Uma empresa é caracterizada pelo nome, a morada e os contactos.

RD2: Uma revista é caracterizada pelo nome, a edição, a categoria e número de consultas (visualizações).

RD3: Os anúncios são caracterizados pelo seu título, o conteúdo, a categoria, o contacto do anunciante e o número de consultas.

RD4: Um artigo é caracterizado pelo título, a data em que foi redigido, o corpo(conteúdo), a categoria e o número de consultas.

RD5: Um escritor é caracterizado pelo seu nome.

RD6: Um editor é caracterizado pelo seu nome.

### **2.2.2.Requisitos de Exploração**

RE1: Permitir a consulta das informações de uma empresa.

RE2: Listar os contatos de uma dada empresa

RE3: Permitir a consulta das informações de um anúncio.

RE4: Permitir a consulta das informações de um artigo.

RE5: Listar todas as revistas de uma dada empresa.

RE6: Listar todos os artigos contidos numa revista.

RE7: Listar todos os anúncios contidos numa revista.

RE8: Listar todos os artigos escritos por um dado escritor.

RE9: Listar todas as revistas editadas por um dado editor.

RE10: Identificar a quantidade de revistas visualizadas de uma empresa.

RE11: Listar todos os escritores que têm pelo menos um artigo escrito numa determinada revista.

RE12: Listar todas as revistas de uma dada empresa numa determinada data ou intervalo de datas.

RE13: Listar todos os escritores que já contribuíram para a escrita de revistas de uma dada empresa.

RE14: Listar todas as revistas em que aparecerem os anúncios de uma determinada empresa.

RE15: Listar as N revistas com maior número de consultas.

RE16: Listar as categorias por frequência relativas de consultas.

### **2.2.3.Requisitos de Controlo**

RC1: O gerente da aplicação deverá ter permissão para adicionar revistas e todas as informações das mesmas.

RC2: Atualizar visualizações associadas a uma revista, anúncio ou artigo sempre que um destes seja consultado.

RC3: Impedir a criação de uma revista com uma data de edição inferior à última edição criada.

RC4: O utilizador da aplicação apenas poderá consultar informações.

## **2.3. Análise geral dos requisitos**

Após o levantamento dos requisitos, os quatro candidatos a bons e dedicados engenheiros decidiram reunir-se para elaborar uma estratégia de ataque/ação face ao problema que a Dona Ritinha havia colocado.

Foram necessárias algumas reuniões até que o conjunto final de requisitos ficasse finalmente estabelecido e com a concordância de todos os membros.

O sistema de base de dados em questão deve ser utilizado com o auxílio a uma aplicação *web* ou *mobile* desenvolvida neste contexto.

A administradora, Dona Ritinha, deve conseguir inserir novas revistas e os respetivos dados associados à mesma. Por outro lado, os utilizadores devem apenas conseguir consultar todos os artigos e anúncios contidos no sistema.

### 3. Modelação conceptual

Nesta secção começaremos por apresentar a abordagem de modelação realizada e de seguida identificar e caracterizar as entidades, os relacionamentos e as associações dos atributos com as entidades e relacionamentos. Será também apresentado o modelo conceptual e feita a sua validação com o utilizador.

#### 3.1. Apresentação da abordagem de modelação realizada

A abordagem de modelação realizada envolve duas vistas de análise de desenvolvimento: o utilizador da aplicação que só poderá consultar os dados e a Dona Ritinha que poderá aceder, inserir e manipular as informações armazenadas na base de dados.

#### 3.2. Identificação e caracterização das entidades

Com base na análise dos requisitos que foram apresentados anteriormente, identificamos a necessidade da existência das seguintes seis entidades:

- **Empresa:** representa todas as empresas responsáveis pela produção das revistas e/ou patrocínio de anúncios;
- **Revista:** representa todas as revistas existentes no arquivo;
- **Anúncio:** representa todos os anúncios publicados;
- **Artigo:** representa todos os artigos publicados;
- **Editor:** representa todos os editores de revistas;
- **Escritor:** representa todos os escritores dos artigos publicados.

Apresentamos agora o dicionário de dados que representa as entidades com algumas informações das mesmas:

Nome da Entidade	Descrição	Ocorrências
Empresa	Termo geral que descreve todas as empresas existentes, produtoras de revistas e/ou patrocinadores de anúncios.	Cada empresa é responsável pela produção das revistas e/ou pelo patrocínio de anúncios.
Revista	Termo geral que descreve as revistas existentes no arquivo.	Cada revista é produzida por uma empresa e contém artigos e anúncios.
Anúncio	Termo geral que descreve todos os anúncios publicados.	Cada revista tem anúncios publicados.
Artigo	Termo geral que descreve todos os artigos publicados.	Cada revista tem artigos publicados.
Editor	Termo geral que descreve todos os editores de revistas.	Cada editor é o responsável por fazer a edição das revistas.
Escritor	Termo geral que descreve todos os escritores dos artigos publicados.	Cada escritor é o responsável pela escrita dos artigos publicados.

Tabela 1 - Dicionário de dados das entidades

### 3.3. Identificação e caracterização dos relacionamentos

Tendo em consideração todos os requisitos apresentados anteriormente e as entidades existentes, iremos agora a apresentar os seis relacionamentos que identificamos entre as mesmas.

Visto que a entidade Revista é o elemento central do nosso arquivo, esta tem vários relacionamentos. Nomeadamente, uma revista contém artigos e anúncios, pertence a uma empresa e é editada por um editor. Quanto aos restantes, um artigo é redigido por um escritor e uma empresa financia anúncios.

- **Relacionamento Revista - Anúncio**

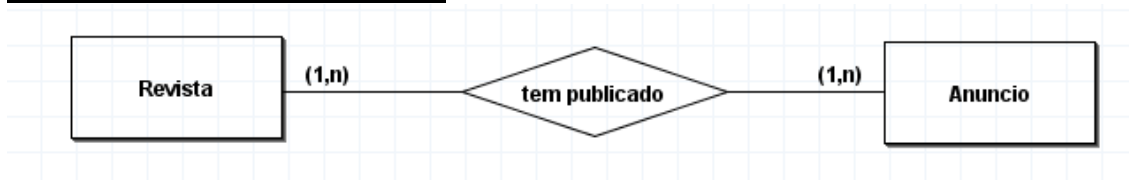


Figura 1 - Relacionamento Revista – Anúncio.

Este relacionamento identifica os anúncios publicados nas revistas.

A cardinalidade apresentada foi baseada no facto de uma revista ter vários anúncios e de um anúncio poder estar publicado em várias revistas.

- **Relacionamento Revista – Artigo**



Figura 2 - Relacionamento Revista – Artigo.

Este relacionamento identifica os artigos publicados em cada revista.

A multiplicidade deste relacionamento é 1:N, visto que uma revista tem vários artigos publicados e um artigo é publicado em uma revista.

- **Relacionamento Revista – Empresa**

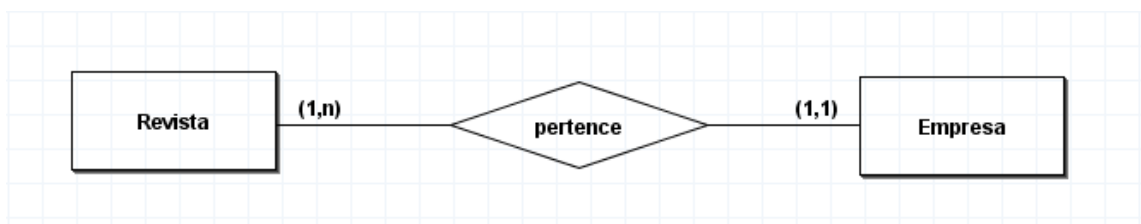


Figura 3 - Relacionamento Revista – Empresa.

O relacionamento agora apresentado identifica as revistas que pertencem a cada empresa.

Este relacionamento apresenta cardinalidade N:1, ou seja, significa que uma revista pertence a uma e uma só empresa, e uma empresa pode ser produtora de muitas revistas.

- **Relacionamento Revista – Editor**



Figura 4 - Relacionamento Revista – Editor.

Este relacionamento indica-nos as revistas que foram editadas por cada um dos editores.

Este relacionamento apresenta, mais uma vez, uma cardinalidade N:1, tendo em conta que uma revista é editada por um e um só editor, e um editor edita várias revistas.



- **Relacionamento Empresa – Anúncio**

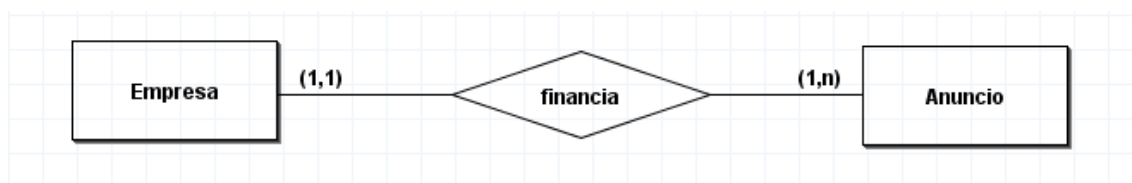


Figura 5 - Relacionamento Empresa – Anúncio.

O relacionamento entre a empresa e o anúncio identifica quais são os anúncios que são financiados por uma empresa.

A multiplicidade apresentada deve-se ao facto de uma empresa poder financiar muitos anúncios e um anúncio ser financiado por uma só empresa.

- **Relacionamento Artigo - Escritor**



Figura 6 - Relacionamento Artigo – Escritor.

Este relacionamento associa os artigos aos seus respetivos escritores.

A cardinalidade deste relacionamento apresentada é de N:N, visto que um artigo pode ser redigido por vários escritores e um escritor é autor de vários artigos.

Apresentamos agora o dicionário de dados que representa os relacionamentos com as informações dos mesmos:

Entidade	Multiplicidade	Relação	Multiplicidade	Entidade
Revista	N	tem publicado	N	Anúncio
	1	tem publicado	N	Artigo
	N	pertence	1	Empresa
	N	é editada	1	Editor
Empresa	1	Financia	N	Anúncio
Artigo	N	é redigido	N	Escritor

Tabela 2 - Dicionário de dados dos relacionamentos.

### 3.4. Identificação e caracterização das associações dos atributos com as entidades e relacionamentos

Tendo em consideração os requisitos, as entidades e os relacionamentos estabelecidos anteriormente, determinamos agora os atributos referentes a cada uma das entidades.

Na caracterização da entidade Empresa foi utilizado o atributo composto Morada, que é composto por Cidade e Rua. Foi também utilizado o atributo multi-valor Contacto, pois a empresa tem vários contactos associados. Para além deste, ainda para a entidade Empresa, decidimos também criar o atributo derivado QuantidadeRevistas, que indica o número de revistas que uma empresa tem.

Cada uma das entidades tem um atributo definido como a chave primária. Como esta tem que ser única, decidimos criar um Id para cada uma delas como sendo o identificador único da entidade.

Na entidade Revista poderíamos ter usado como chave primária a Edição, que é a data da mesma, mas não seria única, pois as revistas são de várias empresas e por este motivo, seriam várias as revistas com a mesma data. Logo, poderiam existir chaves primárias iguais. Definindo o Id como chave primária, garantimos que não haverá este problema.

A identificação dos atributos encontra-se na seguinte tabela:

Entidade	Atributo	Descrição	Domínio	Nulo	Composto	Multivalor	Derivado
Revista	Id	Identificador da revista.	INT	Não	Não	Não	Não
	Nome	Nome da revista.	VARCHAR(45)	Não	Não	Não	Não
	Nr consultas	Número de consultas feitas à revista.	INT	Não	Não	Não	Não
	Categoria	Categoria da revista.	VARCHAR(45)	Não	Não	Não	Não
	Edicao	Data da revista.	DATE	Não	Não	Não	Não
Anúncio	Id	Identificador do anúncio.	INT	Não	Não	Não	Não
	Categoria	Categoria do anúncio.	VARCHAR(45)	Não	Não	Não	Não
	Conteúdo	Conteúdo do anúncio.	VARCHAR(2500)	Não	Não	Não	Não

	Nr consultas	Número de consultas feitas ao anúncio.	INT	Não	Não	Não	Não
	Contacto	Contacto da empresa que publicou o anúncio.	VARCHAR(45)	Sim	Não	Não	Não
	Título	Título do anúncio.	VARCHAR(45)	Não	Não	Não	Não
Artigo	Id	Identificador do artigo.	INT	Não	Não	Não	Não
	Título	Título do artigo.	VARCHAR(45)	Não	Não	Não	Não
	Categoria	Categoria do artigo.	VARCHAR(45)	Não	Não	Não	Não
	Corpo	Conteúdo do artigo.	VARCHAR(2500)	Não	Não	Não	Não
	Data	Data de quando o artigo foi escrito.	DATE	Não	Não	Não	Não
	Nr consultas	Número de consultas feitas ao artigo.	INT	Não	Não	Não	Não
Empresa	Id	Identificador da empresa.	INT	Não	Não	Não	Não
	Nome	Nome da empresa.	VARCHAR(45)	Não	Não	Não	Não
	Contacto	Contactos da empresa.	VARCHAR(45)	Não	Não	Sim	Não
	Morada	Morada da empresa.	VARCHAR(45)	Não	Sim	Não	Não
	Composto por:						
	Cidade	Cidade da empresa.	VARCHAR(45)	Não	Não	Não	Não
	Rua	Rua da empresa.	VARCHAR(45)	Não	Não	Não	Não
	QuantidadeRevistas	Quantidade de revistas que pertencem a uma empresa.	INT	Não	Não	Não	Sim
Editor	Id	Identificador do editor.	INT	Não	Não	Não	Não

	Nome	Nome do editor.	VARCHAR(45)	Não	Não	Não	Não
Escritor	Id	Identificador do escritor.	INT	Não	Não	Não	Não
	Nome	Nome do escritor.	VARCHAR(45)	Não	Não	Não	Não

Tabela 3 - Identificação dos atributos

### 3.5. Detalhe ou generalização de entidades

Com base em toda a informação recolhida dos requisitos e identificadas as nossas entidades, decidimos que não seria necessário detalhar nem fazer uma generalização das mesmas.

### 3.6. Apresentação e explicação do diagrama ER

Tendo em conta as entidades, relacionamentos e atributos identificados e como forma de apresentação do nosso modelo ao utilizador, desenvolveu-se o diagrama conceptual apresentado de seguida.

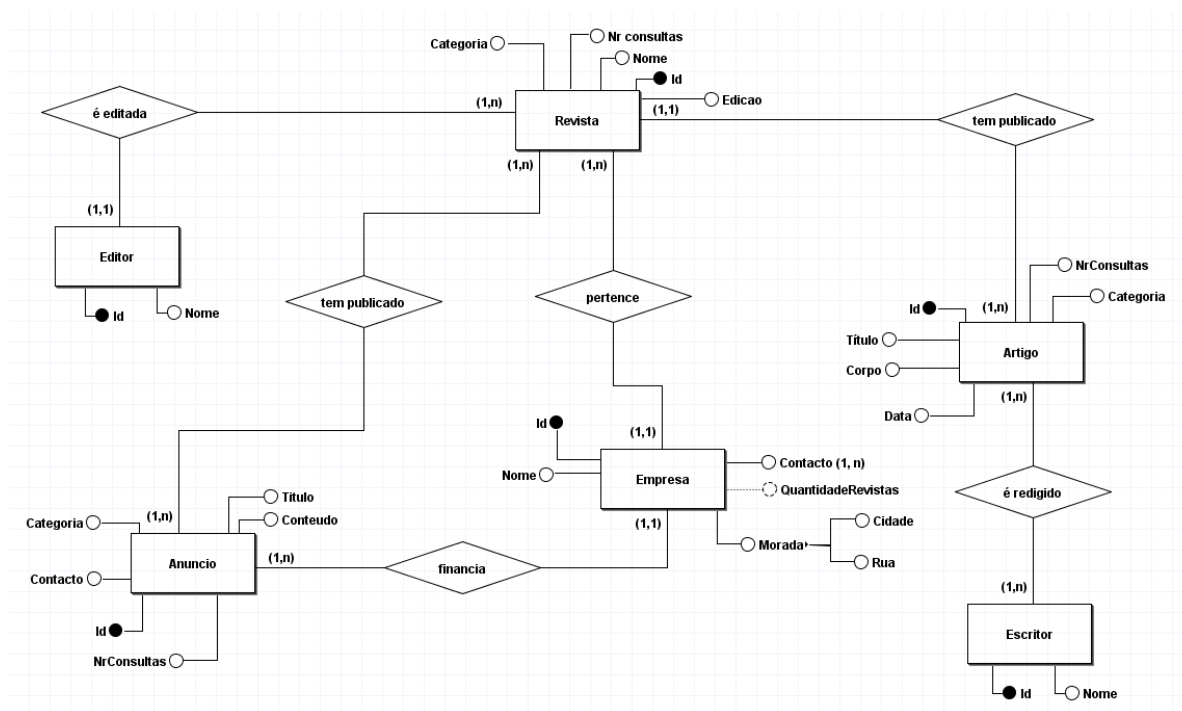


Figura 7 - Modelo Conceptual

### **3.7. Validação do modelo de dados com o utilizador**

Concluído o modelo conceptual, achamos por bem pedir a aprovação da nossa cliente e perceber o que concorda e o que não concorda da organização deste projeto.

É importante que o nosso modelo represente todos os requisitos que foram solicitados pela Dona Ritinha.

Assim, expôs-se o nosso modelo à Dona Ritinha esclarecendo cada uma das decisões tomadas ao longo de toda a elaboração do modelo.

A cliente avaliou a opção que lhe foi proposta e verificou então que todos os requisitos solicitados estão todos representados de forma correta.

Dado que todos os requisitos são satisfeitos, a cliente autorizou a continuação do projeto, agora a elaboração do modelo lógico.

## 4. Modelação Lógica

Nesta secção iremos descrever a construção e validação do nosso modelo lógico, apresentando o desenho do mesmo com recurso ao programa *MySQL Workbench*.

No processo de validação do nosso modelo verificamos que este se encontrava normalizado e que era capaz de responder às interrogações e transações por parte de cada utilizador.

### 4.1. Construção e validação do modelo de dados lógico

A obtenção do modelo lógico foi feita a partir do modelo conceptual através da aplicação de variados passos específicos. De acordo com esses passos, criou-se uma tabela para cada uma das entidades, para cada relacionamento de N para N e para cada atributo multivalorado representados no modelo conceptual já apresentado anteriormente.

De seguida iremos enumerar as nossas relações, representando as chaves primárias usando um sublinhado contínuo e as chaves estrangeiras usando um sublinhado tracejado. Para as chaves que são simultaneamente primárias e estrangeiras utilizamos o duplo tracejado.

- Revista: ID, Nome, Edição, Categoria, NrConsultas, Editor\_ID, Empresa\_ID
- Artigo: ID, Título, Conteúdo, Data, Categoria, NrConsultas, Revista\_ID
- Escritor: ID, Nome
- Editor: ID, Nome
- Empresa: ID, Nome, Cidade, Rua, QuantidadeRevistas
- Anuncio: ID, Contacto, Categoria, Título, Conteúdo, NrConsultas, Empresa\_ID
- AnuncioRevista: Anuncio\_ID, Revista\_ID
- EscritorArtigo: Artigo\_ID, Escritor\_ID
- Contactos: Empresa\_ID, Contacto

Na tabela Empresa, o atributo composto Morada presente no modelo conceptual é traduzido pelos seus atributos simples (Cidade e Rua) no modelo lógico.

Na tabela Revista, os relacionamentos de N para 1 entre esta entidade e o Editor e entre esta entidade e a Empresa levaram à criação de duas chaves estrangeiras, Editor\_ID e Empresa\_ID.

Na tabela Artigo, o relacionamento de N para 1 entre esta entidade e a Revista levou à criação da chave estrangeira Revista\_ID.

Na tabela Anúncio, o relacionamento de N para 1 entre esta a entidade Empresa levou à criação da chave estrangeira Empresa\_ID.

As relações *AnuncioRevista* e *EscritorArtigo* representam, respetivamente, os relacionamentos entre as entidades *Anuncio* e *Revista* e entre *Escritor* e *Artigo* que no modelo conceptual eram relacionamentos de N para N. Nas tabelas *AnuncioRevista* e *EscritorArtigo* podemos constatar que cada uma tem uma cópia das chaves primárias de ambas as entidades envolvidas, que servem de chaves estrangeiras. Assim, estas chaves vão fazer parte da chave primária da tabela do relacionamento.

Houve necessidade da criação da tabela *Contactos*, pois este (*Contacto*) era um atributo multivalorado da empresa, o que levou então à criação de uma tabela com duas chaves estrangeiras que também são primárias: *Empresa\_id* e o próprio atributo *contacto*.

## 4.2. Desenho do modelo lógico

Com recurso ao *MySQL Workbench* obtivemos o seguinte modelo lógico:

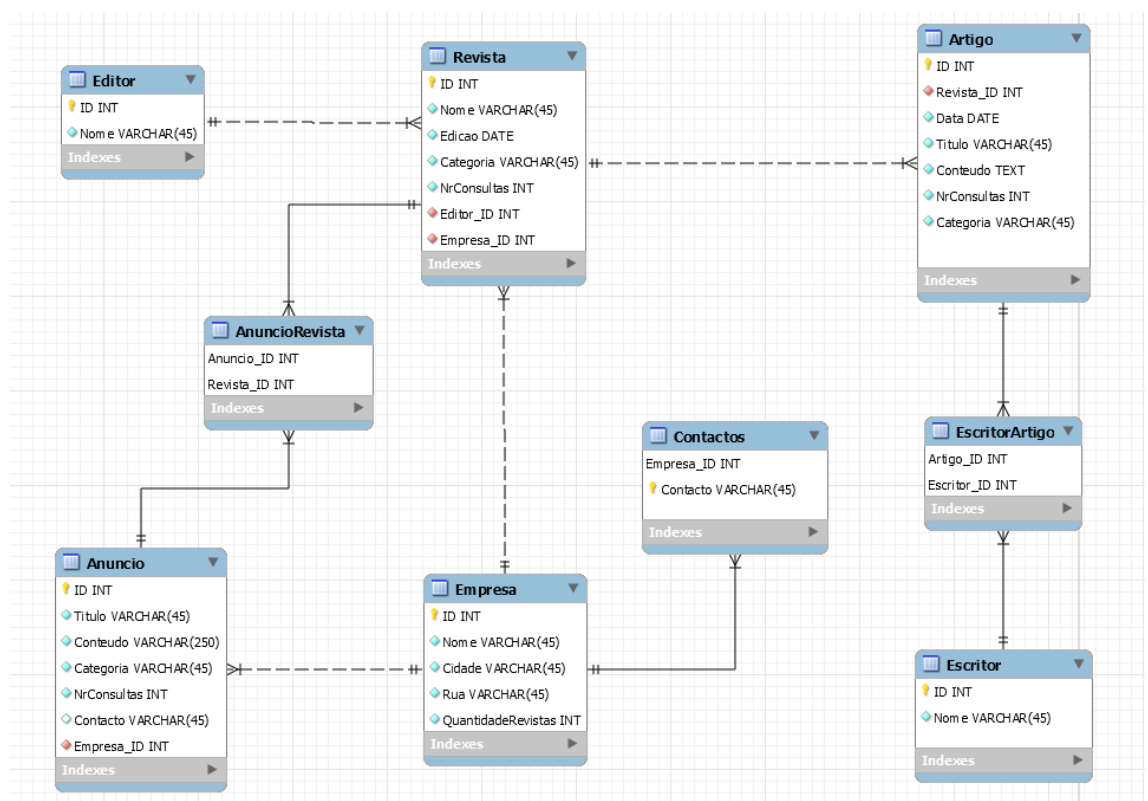


Figura 8 - Modelo Lógico

## 4.3. Validação do Modelo através da Normalização

### 4.3.1. Primeira forma normal (1NF)

Segundo a definição, presente em Connolly (Connolly,2015), podemos afirmar que se uma relação na qual a interseção de cada linha e coluna contém um e um só elemento, está na primeira forma normal.

O nosso modelo lógico obedece a esta regra logo nada temos a alterar.

### 4.3.2. Segunda forma normal (2NF)

Todos os atributos de uma tabela que não são chaves primárias têm que ser totalmente e funcionalmente dependentes da chave primária da mesma (Connolly, 2015).

O nosso modelo lógico cumpre esta regra, pois não existem dependências funcionais parciais.

### 4.3.3. Terceira forma normal (3NF)

Uma relação que está na primeira e segunda forma normal e em que todos os atributos, que não são chaves primárias, são transitivamente dependentes da chave primária está na terceira forma normal (Connolly, 2015).

O nosso modelo lógico cumpre a terceira forma normal e para se fazer esta verificação analisamos as dependências funcionais de cada relação do modelo lógico com o objetivo de se encontrar dependências transitivas. As dependências funcionais dessas relações são:

Revista: ID → Nome, Edição, Categoria, NrConsultas, Editor\_ID, Empresa\_ID

Editor: ID → Nome

Artigo: ID → Título, Corpo, Data, Categoria, NrConsultas, Revista\_ID

Escritor: ID → Nome

Empresa: ID → Nome, Cidade, Rua, QuantidadeRevistas

Anuncio: ID → Contacto, Categoria, Título, Conteúdo, NrConsultas, Empresa\_ID

Concluimos assim que todos os atributos, à exceção daqueles que são chave primária, dependem apenas da respetiva chave primária da relação e não de outros atributos.



## 4.4. Validação do modelo com interrogações do utilizador

Ao entrevistarmos a Dona Ritinha facilmente concluímos que ela pretendia que a aplicação fosse capaz de fazer vários tipos de pesquisas considerando diversos parâmetros. Tendo em conta os desejos da nossa cliente elaboramos os requisitos de exploração, e apresentamos agora como é suposto a base de dados “responder” a estas.

### 4.4.1. Permitir a consulta das informações de uma empresa

As informações relativas a uma empresa podem ser obtidas através da consulta da tabela Empresa.

$$\pi_{Nome,Cidade,Rua,QuantidadeRevistas}(\sigma_{Nome=nomeEmpresa}Empresa)$$

### 4.4.2. Listar os contactos de uma dada empresa

A partir da relação Empresa e da relação Contacto, somos capazes de obter os contactos da empresa em questão.

$$\pi_{Contacto}Contactos \bowtie_{Empresa.ID = Contactos.Empresa\_ID \text{ and } Empresa.Nome = nomeEmpresa} Empresa$$

### 4.4.3. Permitir a consulta das informações de um anúncio

As informações relativas a um anúncio podem ser obtidas através da consulta da tabela Anuncio.

$$\pi_{Titulo,Conteudo,Categoria,Contacto}(\sigma_{Titulo=tituloAnuncio}Anuncio)$$

### 4.4.4. Permitir a consulta das informações de um artigo

As informações relativas a um artigo podem ser obtidas através da consulta da tabela Artigo.

$$\pi_{Titulo,Data,Corpo}(\sigma_{Titulo=nomeArtigo \text{ and } Data=dataArtigo}Artigo)$$

#### 4.4.5. Listar todos as revistas de uma dada empresa

De forma a obter as revistas de uma dada empresa, basta consultar as revistas cujo ao atributo Empresa\_ID da mesma seja igual ao atributo ID da empresa para a qual pretendemos obter a informação.

$$\pi_{R.Nome, R.Categoria}(\sigma_{E.Nome=nomeEmpresa}(\rho_R Revista \bowtie_{E.ID = R.Empresa\_ID} \rho_E Empresa))$$

#### 4.4.6. Listar todos os artigos contidos numa revista

Para obter os artigos contidos numa revista, basta consultar os artigos cujo ao atributo Revista\_ID seja igual ao atributo ID da revista para a qual pretendemos obter a informação.

$$\pi_{A.Titulo, A.Data, A.Corpo}(\sigma_{R.Nome=nomeRevista \text{ and } A.Revista\_ID=R.ID}(\rho_A Artigo \bowtie \rho_R Revista))$$

#### 4.4.7. Listar todos os anúncios contidos numa revista

Para obter os anúncios contidos numa revista, basta consultar as relações da tabela AnuncioRevista para as quais o atributo Revista\_ID seja igual ao ID da revista em questão.

$$\pi_{A.Titulo, A.Categoria, A.NrConsultas, A.Contacto, A.Conteudo}(\sigma_{R.Nome = nomeRevista \text{ and } R.Edicao = edicao}(\rho_A Anuncio \bowtie_{A.ID = AR.Anuncio\_ID} \rho_{AR} AnuncioRevista \bowtie_{AR.Revista\_ID = R.ID} \rho_R Revista))$$

#### 4.4.8. Listar todos os artigos escritos por um dado escritor

De forma a obter os artigos escritos por um escritor, basta consultar as relações da tabela ArtigoEscritor para as quais o atributo Escritor\_ID seja igual ao ID do escritor em questão.

$$\pi_{A.Titulo, A.Data, A.Categoria, A.Corpo}(\sigma_{E.Nome = nomeEscritor}(\rho_A Artigo \bowtie_{A.ID = EA.Artigo\_ID} \rho_{EA} ArtigoEscritor \bowtie_{EA.Escritor\_ID = E.ID} \rho_E Escritor))$$

#### 4.4.9. Listar todas as revistas editadas por um dado editor

De forma a obter as revistas editadas por um editor, basta percorrer as revistas e verificar se o editor da mesma coincide com o pretendido.

$$\pi_{R.Nome, R.Edicao}(\sigma_{R.Editor\_ID = E.ID \text{ and } E.Nome = nomeEditor}(\rho_R Revista \bowtie \rho_E Editor))$$

#### 4.4.10. Identificar a quantidade de revistas visualizadas de uma empresa

Para obter a quantidade de revistas visualizadas de uma empresa, basta somar o número de consultas (NrConsultas) de cada revista que a empresa em questão publicou.

$$\tau_{R.NrConsultas\ desc} (\pi_{R.Nome, R.Edicao, R.NrConsultas} (\sigma_{E.Nome = nomeEmpresa\ and\ E.ID = R.Empresa\_ID} (\rho_R Revista \bowtie \rho_E Empresa))))$$

#### 4.4.11. Listar todos os escritores que têm pelo menos um artigo escrito numa determinada revista

Para listar todos os escritores que têm pelo menos um artigo escrito numa determinada revista, temos de percorrer os artigos publicados na revista pretendida e de seguida verificar os escritores que contribuíram na redação desse artigo.

$$\pi_{E.Nome} (\sigma_{R.Nome = nomeRevista\ and\ R.Edicao = edicaoRevista} (\rho_E Escritor \bowtie_{E.ID = EA.Escritor\_ID} \rho_{EA} EscritorArtigo \bowtie_{EA.Artigo\_ID = A.ID} \rho_A Artigo \bowtie_{A.Revista\_ID = R.ID} \rho_R Revista)))$$

#### 4.4.12. Listar todas as revistas de uma dada empresa numa determinada data ou intervalo de datas

De forma a obter a listagem de todas as revistas de uma dada empresa numa determinada data ou intervalo de datas, basta percorrer todas as revistas de uma dada empresa e verificar se o atributo Data se encontra dentro dos parâmetros pretendidos.

$$\sigma_{Empresa.Nome = nomeEmpresa\ and\ Revista.Edicao \geq dataInicio\ and\ Revista.Edicao \leq dataFim} (\rho_R Revista \bowtie_{R.Empresa\_ID = E.ID} \rho_E Empresa)$$

#### 4.4.13. Listar todos os escritores que já contribuíram para a escrita de revistas de uma dada empresa

De forma a obter a listagem de todos os escritores que já contribuíram para a escrita de revistas de uma dada empresa, temos de percorrer as revistas publicadas por uma empresa, verificar os artigos publicados na mesma para depois verificarmos os escritores contribuintes na sua redação.

$$\pi_{E.Nome} (\sigma_{E.Nome = nomeEmpresa} (\rho_E Escritor \bowtie_{EA.Escritor\_ID = E.ID} \rho_{EA} EscritorArtigo \bowtie_{A.ID = EA.Artigo\_ID} \rho_A Artigo \bowtie_{R.ID = A.Revista\_ID} \rho_R Revista \bowtie_{E.ID = R.Empresa\_ID} \rho_E Empresa)))$$

#### 4.4.14. Listar todas as revistas em que aparecerem os anúncios de uma determinada empresa

Para obter todas as revistas em que aparecerem os anúncios de uma determinada empresa, basta verificar a relação entre Anuncio e Empresa, através do campo Empresa\_ID, para posteriormente verificarmos as revistas nas quais esses anúncios estão presentes.

$$\pi_{R.Nome,R.Edicao} (\sigma_{E.Nome = nomeEmpresa} (\rho_R Revista \bowtie_{AR.Revista\_ID = Revista.ID} \rho_{AR} AnuncioRevista \bowtie_{A.ID = EA.Anuncio\_ID} \rho_A Anuncio \bowtie_{E.ID = A.Empresa\_ID} \rho_E Empresa)))$$

#### 4.4.15. Listar as N revistas com maior número de consultas

Para a resolução desta interrogação basta ordenar decrescentemente as revistas por número de consultas (NrConsultas) e aplicando um limite (N) ao número de resultados devolvidos.

$$\sigma_{rownum() > 0 \text{ and } rownum() \leq N} (\tau_{NrConsultas \text{ desc}} (\pi_{Nome,Edicao,Categoria,NrConsultas})) Revista$$

#### 4.4.16. Listar as categorias por frequências relativas de consultas

Para calcular as frequências relativas das consultas das categorias, primeiro calculamos o número de consultas totais, ou seja, o número de consultas de todas as revistas. De seguida, agrupamos a tabela das revistas por categoria e somamos o número de consultas das revistas da mesma categoria. Por fim, dividimos esse valor pelo número total de consultas multiplicando por cem de modo a obter as frequências relativas.

$$\tau_{nTotalC\ desc} (\pi_{Categoria, nTotalC} \gamma_{Categoria});$$
$$(SUM(NrConsultas) / nTotal) * 100 \rightarrow nTotalC \text{ ) Revista}$$

#### 4.5. Validação do modelo com as transações estabelecidas

Para que as informações que constam na base de dados sejam facilmente acedidas, optamos por desenvolver operações que permitem executar tarefas básicas na aplicação onde irá operar a base de dados. A este conjunto de operações dá-se o nome de transações. Quando as operações de uma transação são todas executadas, a transação é executada. No caso de ocorrer um erro numa dada operação, todas as outras serão anuladas e a transação não ocorre.

Consideramos essencial o desenvolvimento de transações que permitam a inserção de uma nova revista, uma vez que este era o principal objetivo da Dona Ritinha. As transações que permitem remover e atualizar dados não deverão ser executadas com muita frequência tendo em conta o objetivo do projeto.

Para inserir uma revista é necessário que a transação tenha acesso a todas as novas informações da mesma, tais como nome, edição e categoria (o número de consultas começará sempre igual a zero). Primeiro são preenchidos todos os atributos do novo registo na tabela Revista com a nova informação e, de seguida, completa-se a tabela Artigo com os artigos que constam na Revista recém adicionada. Na tabela Escritor serão inseridos todos os que ainda não constarem da base de dados e efetua-se o relacionamento dos mesmos com os Artigos, adicionando na tabela EscritorArtigo. Na tabela Anuncio são inseridos os novos registos e após isso é preenchida a tabela que traduz o relacionamento entre Anuncio e Revista, inserindo os respetivos identificadores. Na tabela Empresa preenche-se o que ainda não constar da base de dados, assim como na tabela Editor.

## **4.6. Revisão do modelo lógico com o utilizador**

Após a elaboração do modelo lógico, entrevistamos novamente a Dona Ritinha para podermos expor o novo modelo e obter o seu parecer relativamente ao mesmo. Tendo em conta que a interpretação do modelo lógico já requer mais conhecimentos de base de dados, tentamos passar a informação de um modo mais funcional e simplificado para que a Dona Ritinha pudesse ter uma visão geral do funcionamento.

Após ter compreendido o modelo a Dona Ritinha verificou que todos os requisitos que havia estabelecido no início estavam presentes e que os atributos de cada entidade eram capazes de armazenar toda a informação que se pretendia. De seguida a Dona Ritinha questionou como seria possível responder às várias questões que surgem ao pesquisar os vários dados. No seguimento desta questão apresentamos as questões e respetivas respostas presentes na secção 4.4 as quais deram uma visão do método de pesquisa e tiveram a sua concordância.

Perante esta situação tivemos autorização por parte da Dona Ritinha para avançar com o projeto.

## **5. Implementação Física**

Nesta secção iremos abordar os aspetos relativos à metodologia para a construção do modelo físico bem como alguns exemplos. Será detalhado o processo de transição do modelo lógico para o modelo físico, utilizando para esse efeito uma ferramenta de gestão de Base de Dados, nomeadamente o MySQL.

### **5.1. Seleção do sistema de gestão de bases de dados**

O Sistema de Gestão de Base de Dados utilizado para desenvolver a base de dados relacional para este projeto foi o MySQL. Esta ferramenta apresenta várias vantagens, as quais passamos de seguida a enumerar algumas delas:

- É um software grátis;
- É fácil de utilizar;
- Tem um elevado desempenho;
- Possui compatibilidade cruzada, isto é, pode ser instalado em todos os sistemas operativos principais sem perda de desempenho.

Como nenhum sistema é perfeito o MySQL também tem algumas desvantagens, mas tendo em conta todos os fatores considerou-se que o MySQL seria uma boa opção para o projeto em questão.

### **5.2. Tradução do esquema lógico para o sistema de gestão de bases de dados escolhido em SQL**

O programa utilizado para o desenvolvimento desta base de dados foi o “MySQL Workbench”. Esta ferramenta possibilita a utilização de uma opção, Forward Engineer, após o modelo lógico estar definido. Este comando transforma o modelo lógico desenvolvido em físico, facilitando a implementação física da base de dados ao gerar automaticamente o código SQL.

### **5.3. Tradução das interrogações do utilizador para SQL (alguns exemplos)**

Todos os requisitos de exploração definidos na secção 2.2.2 foram criados através de procedimentos com exceção do requisito RE9 que foi criado como uma função. Essa decisão baseou-se no facto de se pretender procurar informações na base de dados através de parâmetros, pelo que apenas é necessário projetar informação das tabelas. No requisito RE9

optamos por criar uma função pois o objetivo é devolver um número (número total de consultas das revistas de uma determinada empresa).

Para além dos conceitos abordados acima, foram também usadas as funções elementares do SQL como *SELECT*, *WHERE*, *ORDER BY*, *GROUP BY*, *INNER JOIN*.

De seguida encontram-se alguns exemplos do código SQL criado de forma a responder aos requisitos de exploração.

### 5.3.1. Listar todos os escritores que já contribuíram para a escrita de revistas de uma dada empresa

Este procedimento faz a pesquisa em cinco tabelas, daí serem usados quatro join's para obter a tabela resultante. É ainda feita uma restrição ao nome da empresa.

```
-- (5) Listar todos os artigos contidos numa revista.  
DELIMITER $$  
CREATE PROCEDURE getArtigosByRevista (IN nomeRevista VARCHAR(45))  
BEGIN  
    SELECT A.Titulo , A.Data , A.Corpo FROM Artigo AS A , Revista AS R  
    WHERE R.Nome = nomeRevista AND A.Revista_ID = R.ID;  
END$$
```

Figura 9 - Tradução do RE5 para SQL

### 5.3.2. Calcular o número total de consultas das revistas de uma empresa

Esta função foi a única que necessitamos de criar pois é o único requisito que devolve apenas um valor. Primeiro, tabelámos a coluna que contém o número de consultas das revistas, seleccionando as revistas que referenciam a empresa pretendida. De seguida, aplicámos a função SUM do SQL para somar o número de consultas obtendo o resultado pretendido.

```
-- (12) Listar todos os escritores que já contribuíram para a escrita de revistas de uma dada empresa.  
DELIMITER $$  
CREATE PROCEDURE getEscritoresByEmpresa(IN nomeEmpresa VARCHAR(45))  
BEGIN  
    SELECT DISTINCT Escritor.Nome FROM Escritor  
    INNER JOIN EscritorArtigo ON EscritorArtigo.Escritor_ID = Escritor.ID  
    INNER JOIN Artigo ON Artigo.ID = EscritorArtigo.Artigo_ID  
    INNER JOIN Revista ON Revista.ID = Artigo.Revista_ID  
    INNER JOIN Empresa ON Empresa.ID = Revista.Empresa_ID  
    WHERE Empresa.Nome = nomeEmpresa;  
END$$
```

Figura 10 - Tradução do RE10 para SQL



### 5.3.3. Listar as categorias por frequências relativas de consultas

Este procedimento traduz o requisito que pede para calcular as frequências relativas de consultas por categoria. Numa primeira fase, calculámos a tabela que contém as categorias e os respetivos números de consultas totais usando a função GROUP BY, que permite agrupar as categorias pelo respetivo somatório do número de consultas, eliminando as repetições. De seguida, a cada elemento da coluna que contém a soma total do número de consultas de uma certa categoria, vamos dividi-lo pelo número total de visualizações de todas as revistas e multiplicámos por cem, obtendo assim as frequências relativas.

```
-- (16) Listar as categorias por frequências relativas de consultas
DELIMITER $$
CREATE PROCEDURE topCategoriasFR()
BEGIN
    DECLARE nTotal INT DEFAULT 1;

    SELECT SUM(nrConsultas) INTO nTotal FROM Revista;

    SELECT Categoria , (nTotalC / nTotal) * 100 AS 'Visualizações (%)' FROM
    (SELECT Categoria , SUM(NrConsultas) AS nTotalC FROM Revista
    GROUP BY Categoria
    ORDER BY nTotalC DESC) AS X;
END$$
```

Figura 11 - Tradução do RE15 para SQL

### 5.4. Tradução das transações estabelecidas para SQL (alguns exemplos)

Nesta seção apresentamos as transações criadas para a base de dados. Foram desenvolvidas transações para inserção de dados em todas as tabelas do arquivo de revistas. De seguida, apresentamos alguns exemplos dessas transações.

### 5.4.1. Inserir uma empresa

O procedimento recebe como parâmetros o nome, a cidade e a rua da empresa. Primeiro é calculado o índice que esta nova entrada na tabela vai ter como chave primária. De seguida, testamos se existem empresas com o mesmo nome. Caso não haja, a nova empresa é adicionada à tabela usando o comando *COMMIT* e é lançada uma mensagem de sucesso. Caso haja uma empresa com o mesmo nome, através do comando *ROLLBACK*, desfazemos as alterações feitas e é lançada uma mensagem de erro.

```
DELIMITER $$
CREATE PROCEDURE insertEmpresa(IN nomeE VARCHAR(45) , IN cidadeE VARCHAR(45) , IN ruaE VARCHAR(45))
BEGIN
    DECLARE teste INT;
    DECLARE idE INT;

    SET idE = 1 + (SELECT COUNT(ID) FROM Empresa);
    SET teste = (SELECT COUNT(Nome) FROM Empresa WHERE Nome = nomeE);

    START TRANSACTION;

    IF( teste = 0 ) THEN
        INSERT INTO Empresa
        (ID, Nome, Cidade, Rua)
        VALUES
        (idE, nomeE, cidadeE, ruaE);
        SELECT 'Empresa registada com sucesso!!!';
        COMMIT;
    ELSE
        SELECT 'Empresa já existe!!!';
        ROLLBACK;
    END IF;
END$$
```

Figura 12 - Tradução da transação de inserção de uma empresa em SQL

### 5.4.2. Inserir um artigo

O procedimento recebe como parâmetros o nome da revista à qual pertence o artigo a inserir, a data em que foi escrito, o seu título, o corpo (conteúdo), a categoria em que está inserido e, por fim, o seu escritor. Numa primeira fase é calculado os *ID*'s da revista e do escritor do artigo bem como o *ID* que a nova entrada irá ter. De seguida, caso os *ID*'s da revista e do escritor existam, é adicionado o artigo à tabela com referência à respetiva revista, e é atualizada a tabela *EscritorArtigo*, adicionando o novo relacionamento entre o artigo adicionado e o seu escritor através do comando *COMMIT*. Caso os *ID*'s da revista ou do escritor do artigo não existam, a inserção fica sem efeito aplicando o comando *ROLLBACK*.

```
DELIMITER $$
CREATE PROCEDURE inserirArtigo(IN nomeRevistaA VARCHAR(45), IN dataA DATE, IN tituloA VARCHAR(45),
                              IN corpoA VARCHAR(45), IN categoriaA VARCHAR(45), IN escritorA VARCHAR(45))
BEGIN
    DECLARE idR INT;
    DECLARE idE INT;
    DECLARE idA INT;

    SET idR = (SELECT ID FROM Revista WHERE Nome = nomeRevistaA);
    SET idE = (SELECT ID FROM Escritor WHERE Nome = escritorA);
    SET idA = 1 + (SELECT MAX(ID) FROM Artigo);

    IF(idR IS NOT NULL && idE IS NOT NULL)
    THEN
        INSERT INTO Artigo
        (ID, Revista_ID, Data, Titulo, Corpo, Categoria)
        VALUES
        (idA, idR, dataA, tituloA, corpoA, categoriaA);
        INSERT INTO EscritorArtigo
        (Escritor_ID, Artigo_ID)
        VALUES
        (idE, idA);
        SELECT 'Artigo adicionado com sucesso!!!';
        COMMIT;
    ELSE
        SELECT 'Revista ou escritor não existem!!!';
        ROLLBACK;
    END IF;
END$$
```

Figura 13 - Tradução de transação da inserção de um artigo em SQL

## 5.5. Escolha, definição e caracterização de índices em SQL (alguns exemplos)

O *MySQL* faz a atribuição automática de índices aos atributos que sejam chaves primárias, pelo que todas as chaves primárias envolvidas nas tabelas da base de dados são índices. Para além disso, todos os atributos que sejam restringidos como *UNIQUE*, também são indexados.

Face às traduções das interrogações dos utilizadores, podemos verificar que a maior parte das consultas às tabelas são feitas através do atributo 'Nome' das diferentes entidades. Com isto, considerámos restringir esses mesmos atributos como *UNIQUE* para que, consequentemente, a pesquisa sobre as tabelas fosse mais eficiente. No entanto, tal decisão não fazia sentido dado o contexto do problema pois é permitido haver, por exemplo, duas empresas com o mesmo nome ou duas revistas com o mesmo nome.

## 5.6. Estimativa do espaço em disco da base de dados e taxa de crescimento anual

### 5.6.1. Estimativa do espaço em disco da base de dados

Cada entrada em cada relação vai ocupar um certo número de bytes. Começemos por calcular esse espaço:

#### Relação Revista

Nome da Tabela	Atributos	Tipo	Espaço ocupado (bytes)	Total
Revista	ID	INT	4	109
	Nome	VARCHAR(45)	45	
	Edicao	DATE	3	
	Categoria	VARCHAR(45)	45	
	NrConsultas	INT	4	
	Editor_ID	INT	4	
	Empresa_ID	INT	4	

Tabela 4 - Espaço por entrada da tabela Revista

### Relação Empresa

Nome da Tabela	Atributos	Tipo	Espaço ocupado (bytes)	Total
Empresa	ID	INT	4	143
	Nome	VARCHAR(45)	45	
	Cidade	VARCHAR(45)	45	
	Rua	VARCHAR(45)	45	
	QuantidadeRevistas	INT	4	

Tabela 5 - Espaço por entrada da tabela Empresa

### Relação Contacto

Nome da Tabela	Atributos	Tipo	Espaço ocupado (bytes)	Total
Contacto	Empresa_ID	INT	4	49
	Contacto	VARCHAR(45)	45	

Tabela 6 - Espaço por entrada da tabela Contacto

### Relação Artigo

Nome da Tabela	Atributos	Tipo	Espaço ocupado (bytes)	Total
Artigo	ID	INT	4	2605
	Revista_ID	INT	45	
	Data	DATE	250	
	Titulo	VARCHAR(45)	45	
	Corpo	VARCHAR(2500)	4	
	NrConsultas	INT	45	
	Categoria	VARCHAR(45)	4	

Tabela 7 - Espaço por entrada da tabela Artigo

### Relação Escritor

Nome da Tabela	Atributos	Tipo	Espaço ocupado (bytes)	Total
Escritor	ID	INT	4	49
	Nome	VARCHAR(45)	45	

Tabela 8 - Espaço por entrada da tabela Escritor

### Relação Anuncio

Nome da Tabela	Atributos	Tipo	Espaço ocupado (bytes)	Total
Anuncio	ID	INT	4	397
	Titulo	VARCHAR(45)	45	
	Conteudo	VARCHAR(250)	250	
	Categoria	VARCHAR(45)	45	
	NrConsultas	INT	4	
	Contacto	VARCHAR(45)	45	
	Empresa_ID	INT	4	

Tabela 9 - Espaço por entrada da tabela Anuncio

### Relação Editor

Nome da Tabela	Atributos	Tipo	Espaço ocupado (bytes)	Total
Editor	ID	INT	4	49
	Nome	VARCHAR(45)	45	

Tabela 10 - Espaço por entrada da tabela Editor

### Relação AnuncioRevista

Nome da Tabela	Atributos	Tipo	Espaço ocupado (bytes)	Total
AnuncioRevista	Anuncio_ID	INT	4	8
	Revista_ID	INT	4	

Tabela 11 - Espaço por entrada da tabela AnuncioRevista

### Relação ArtigoEscritor

Nome da Tabela	Atributos	Tipo	Espaço ocupado (bytes)	Total
ArtigoEscritor	Artigo_ID	INT	4	8
	Escritor_ID	INT	4	

Tabela 12 - Espaço por entrada da tabela ArtigoEscritor

Necessitamos agora de uma estimativa do espaço total ocupado. Para tal iremos ter como base revistas apenas do presente ano. Iremos considerar o seguinte caso: 250 revistas, 830 artigos, 30 empresas, 35 anúncios, 20 editores e 17 escritores. Iremos também considerar que cada empresa terá 2 contactos, existirão 10 anúncios por revista, as revistas apenas terão 1 editor e os artigos irão ter 3 escritores.

Assim, para este tipo de povoamento, o espaço ocupado seria cerca de:

$$250 * 109 + 830 * 2605 + 30 * 143 + 35 * 397 + 20 * 49 + 17 * 49 + 30 * 49 * 2 + 250 * 10 * 8 + 250 * 8 + 830 * 8 * 3 = 2239513 \text{ bytes} = 2.135 \text{ MB.}$$

### 5.6.2. Taxa de crescimento anual

Para determinar a taxa de crescimento anual, vamos supor que diariamente são inseridas 150 revistas, com 25 anúncios e 70 artigos cada uma. Iremos também assumir que cada revista terá 1 editor, os artigos irão ter 3 escritores cada um e que serão inseridas 0.1 empresas novas, 0.2 editores e 0.3 escritores. Assim sendo e tendo em conta que um ano tem 365 dias, o espaço ocupado aumenta anualmente da seguinte forma:

$$365 * (150 * 109 + 25 * 397 + 70 * 2605 + 70 * 8 + 3 * 70 * 8 + 0.2 * 8 + 0.3 * 8 + 0.1 * 143) = 76972404.5 \text{ bytes} = 73.4 \text{ MB}$$

## 5.7. Definição e caracterização das vistas de utilização do SQL (alguns exemplos)

As interrogações traduzidas para SQL baseiam-se todas em pesquisas personalizadas pelo que necessitam quase sempre de parâmetros de entrada para poderem ser executadas. Consequentemente, decididos criar vistas sobre todas as tabelas com informação relevante tanto para o visitante do site como para o utilizador da aplicação. Foram então criadas vistas sobre todas as entidades da base de dados com exceção das tabelas que surgiram do relacionamento entre entidades. Para além das vistas básicas sobre as entidades, foram também criadas mais duas vistas, uma para a consulta das empresas e o número total de consultas das suas revistas e outra para listar o *Top 5* de escritores, ou seja, os escritores cujos artigos têm mais visualizações. De seguida, encontram-se alguns exemplos do código SQL de algumas vistas.

### 5.7.1. Vista da Tabela Escritor

Vista muito simples que permite aceder ao nome de todos os escritores que estão na base de dados do arquivo de revistas.

```
-- View da tabela Escritor
CREATE VIEW viewEscritor AS
SELECT Nome FROM Escritor;
```

Figura 14 - Vista sobre a tabela Escritor

### 5.7.2. Vista das empresas e respetivas visualizações das suas revistas

Vista que permite visualizar o nome de todas as empresas e respetivas visualizações das suas revistas. Para tal, juntámos as duas tabelas Revista e Empresa, associando uma empresa às respetivas revistas. De seguida, aplicámos a função *GROUP BY* de modo a eliminar as repetições e somar o número de consultas. O resultado é apresentado ordenado por ordem decrescente, para tal, usámos a função *ORDER BY*.

```
-- View da tabela das empresas e do numero de vizualizações das suas revistas
CREATE VIEW viewEmpresaV AS
SELECT Empresa.Nome , SUM(Revista.NrConsultas) AS nTotal FROM Empresa
INNER JOIN Revista ON Empresa.ID = Revista.Empresa_ID
GROUP BY Empresa.Nome
ORDER BY nTotal DESC;
```

Figura 15 - Vista das empresas e respetivas visualizações totais das suas revistas

### 5.7.3. Vista do *Top 5* de escritores

Vista que permite visualizar o *Top 5* de escritores. Para a vista ser criada, simplesmente juntámos as tabelas Escritor e EscritorArtigo e agrupámos a tabela pelo nome de escritor contando o número de ocorrências que um dado escritor tem na última tabela mencionada. Para o resultado final pretendido, usámos a função de ordenação *ORDER BY* para ordenar as entradas pelo número total de artigos e limitando a tabela em cinco entradas através da função *LIMIT* com o parâmetro cinco.

```
-- Top 5 escritores
CREATE VIEW viewTop5Escritores AS
SELECT Nome , COUNT(EscritorArtigo.Escritor_ID) AS NumeroArtigos FROM Escritor
INNER JOIN EscritorArtigo ON EscritorArtigo.Escritor_ID = Escritor.ID
GROUP BY Nome
ORDER BY NumeroArtigos DESC
LIMIT 5;
```

Figura 16 – Vista do *Top 5* de escritores



## 5.8. Definição e caracterização dos mecanismos de segurança em SQL (alguns exemplos)

O site desenvolvido para a Dona Ritinha é utilizado por visitantes (ações de consulta) e pela própria Dona Ritinha (ações de inserção, atualização, remoção e consulta). Dado isto, decidimos criar apenas dois tipos de utilizadores, o administrador (admin) da base de dados e o da aplicação (app), visto que é esta que vai gerir os movimentos da base de dados.

O administrador da base de dados tem todas as permissões possíveis sobre a base de dados visto que é este que vai gerir a mesma.

O utilizador da aplicação tem permissões de seleção, inserção, execução, atualização e de remoção sobre todas as tabelas com exceção das tabelas que tiveram origem das ligações (N:N).

```
-- Criação do perfil de administrador  
  
-- Criação do utilizador  
CREATE USER 'admin'@'localhost'  
  IDENTIFIED BY 'admin';  
  
-- Permissões  
GRANT ALL ON Arquivo.* TO 'admin'@'localhost';
```

Figura 17 - Código SQL para a criação da vista de utilização e respetivas permissões do utilizador admin

```
-- Criação do perfil de app  
  
-- Criação do utilizador  
CREATE USER 'app'@'localhost'  
  IDENTIFIED BY 'app';  
  
-- Permissões  
GRANT SELECT , INSERT , UPDATE ON Arquivo.Anuncio TO 'app'@'localhost';  
GRANT SELECT , INSERT , UPDATE ON Arquivo.Artigo TO 'app'@'localhost';  
GRANT SELECT , INSERT , UPDATE ON Arquivo.Contactos TO 'app'@'localhost';  
GRANT SELECT , INSERT , UPDATE ON Arquivo.Editor TO 'app'@'localhost';  
GRANT SELECT , INSERT , UPDATE ON Arquivo.Empresa TO 'app'@'localhost';  
GRANT SELECT , INSERT , UPDATE ON Arquivo.Escritor TO 'app'@'localhost';  
GRANT SELECT , INSERT , UPDATE ON Arquivo.Revista TO 'app'@'localhost';
```

Figura 18 - Código SQL para a criação da vista de utilização e respetivas permissões do utilizador app

## **5.9. Revisão do sistema implementado com o utilizador**

Terminada a implementação do modelo físico, a equipa reuniu-se com a Dona Ritinha uma última vez para garantir que todos os requisitos foram cumpridos e que o projeto desenvolvido se encontra apto para ser lançado.

Tendo em conta que a Dona Ritinha não possui conhecimentos de informática, a equipa decidiu realizar uns requisitos extra que virão a ser necessários para a aplicação que irá gerir a base de dados, como por exemplo, as definições das vistas sobre as tabelas mais importantes da base de dados, assim como um trigger que atribui o número total de revistas de uma empresa ao atributo que guarda tal informação na entidade Empresa.

No geral, a Dona Ritinha está satisfeita com o trabalho desenvolvido para o seu arquivo de jornais e considera que todas as suas exigências foram cumpridas.

## 6. Conclusões e Trabalho Futuro

De uma forma geral, os quatro corajosos aspirantes a engenheiros informáticos avaliam de uma forma positiva o trabalho desenvolvido para a criação de uma base de dados para armazenamento digital de revistas. Na nossa opinião, um dos aspetos mais relevantes para o desenvolvimento desta base de dados foi desde início a enorme preocupação em sermos consistentes e evitar redundâncias, o que levaria a erros que mais tarde seriam difíceis ou impossíveis de resolver.

Tendo em mente os requisitos a que nos propusemos no início do projeto perante a nossa querida cliente, tivemos um excelente *feedback* da sua parte e um bolinho de chocolate como agradecimento, deduzimos assim que atingimos os nossos objetivos.

Num futuro trabalho pretendemos desenvolver uma base de dados maior e elaborar interrogações e transações mais complexas e diversificadas, no caso do negócio da Dona Ritinha se expandir e ganhar proporções que o exijam. Estamos certos que o tempo, a experiência e as aulas de Base de Dados nos tornarão aptos para isso.

## Referências

- Connolly, T.M. and Begg C. E., 2005. Database Systems A Practical Approach to Design, Implementation, and Management. 6<sup>th</sup> Edition, England: Pearson Education Limited.

## Anexos

```
USE Arquivo;

-- Povoamento da tabela 'Empresa'
INSERT INTO Empresa
(ID, Nome, Cidade, Rua)
VALUES
(1, 'Soberania do Povo Editora, S.A.', 'AGUADA DE BAIXO', 'Avenida Dr. Eugénio Ribeiro, 89 - 3ª '),
(2, 'Edições Saída de Emergência, Lda', 'PORTO SALVO', 'Rua Profª Anibal Cavaco Silva '),
(3, 'Ocasão - Edições Periódicas, Lda', 'ALMADA', 'Rua Capitão Leitão, 66 N '),
(4, 'Dicas e Pistas, Lda', 'Lisboa', 'Rua Manuel Marques, 14, Loja H '),
(5, 'Tipografia Lobão, Lda', 'ALMADA', 'Rua Quinta do Gato Bravo, 5 '),
(6, 'Revista de Propriedades', 'CASCAIS', 'Rua Nova de Alfarrobeira, 8 '),
(7, 'Nugon-Publicações Representações', 'LOURES', 'Rua Nelson Pereira Neves, Lojas 1 e 2 '),
(8, 'Mundo Português', 'LISBOA', 'Avenida Elias Garcia, 57 - 7ª '),
(9, 'RBA Revistas Portugal', 'LISBOA', 'Rua Filipe Folque, 40 - 4ª '),
(10, 'Empresa Região de Leiria', 'LEIRIA', 'Rua Dom Carlos I, 2-4 '),
(11, 'Edições Plural Especializadas', 'LISBOA', 'Avenida Duque de Ávila, 69 '),
(12, 'Company One - Publicações', 'LISBOA', 'Avenida Almirante Reis, 39 '),
(13, 'Empresa Jornalística O Eco', 'POMBAL', 'Largo do Carmo, 20 - R/C Esqª '),
(14, 'Publiotel, Lda', 'LISBOA', 'Rua General Firmino Miguel, 3 '),
(15, 'PM Media', 'MATOSINHOS', 'Rua António Nobre, 23'),
(16, 'Ideias, Projectos e Publicações', 'ESTORIL', 'Avenida Senhora Monte da Saúde, 319 C '),
(17, 'Editorial Grupo V Portugal, Lda', 'LISBOA', 'Campor Grande, 56 - 7ªA '),
(18, 'Júlio Manuel Otero, Lda', 'PAIO PIRES', 'Praceta José Sebastião e Silva'),
(19, 'Edições Cotovia, Lda', 'LISBOA', 'Rua Nova da Trindade, 24 '),
(20, 'Sediton', 'TONDELA', 'Rua Dr. António Marques da Costa '),
(21, 'Editorial Caminho, S.A.', 'AMADORA', 'Rua Cidade de Cordova, 2 '),
(22, 'IF Executives', 'LISBOA', 'Rua Basílio Teles, 35 - 1ª Dtª '),
(23, 'Brandscape, Lda', 'LISBOA', 'Rua das Fontainhas Nº 74 '),
(24, 'Armazens Cesar De Carvalho', 'BARQUINHA', 'Estrada Nacional 3, Km 87 '),
(25, '2020 Editora, Lda', 'ALMADA', 'Rua Alfredo Da Silva Nr. 12 '),
(26, 'Paulus Editora', 'LISBOA', 'Rua Dom Pedro Cristo, Nr. 10/12 '),
(27, 'Rui Romano - Edições', 'LISBOA', 'Rua Filipe Foque, 2, Loja 2 C/D '),
(28, 'Goody - Consultoria S.A.', 'LISBOA', 'Avenida Infante D. Henrique, 306'),
(29, 'Euromedice - Edições', 'ALGÉS', 'Alameda António Sérgio, 22, 4ª B '),
(30, 'Sociedade Vicra Desportiva, S.A.', 'LISBOA', 'Travessa da Queimada, 23'),
(99, 'Particular', '', '');
```

Figura 19 - Povoamento da tabela Empresa

```
-- Povoamento da tabela 'Contacto'
INSERT INTO Contactos
(Empresa_ID, Contacto)
VALUES
(1, '234123654'),
(1, '234176454'),
(2, '200234874'),
(3, '212267856'),
(4, '213143256'),
(4, '213574656'),
(4, '213807980'),
(5, '212783256'),
(6, '211823454'),
(7, '214252474'),
(8, '213345556'),
(8, '213346344'),
(9, '213143568'),
(10, '221782434'),
(11, '213457457'),
(...)
```

Figura 20 - Povoamento da tabela Contactos

```
-- Povoamento da tabela 'Anuncio'
INSERT INTO Anuncio
(ID, Titulo, Conteudo, Categoria, NrConsultas, Contacto, Empresa_ID)
VALUES
(1, 'Vendo T2+1', 'Vendo t2+1 lindo, espetacular e fantástico', 'Imóveis', FLOOR(RAND()*(1+30)), '543216722', 99),
(2, 'Renault Clio 1.9d', '73km, Gasolina, 2018, 16.250€', 'Automóveis', FLOOR(RAND()*(1+30)), '123456722', 99),
(3, 'Vendo Sogra', 'Ótima dona de casa, pena é ser chata', 'Bricolage', FLOOR(RAND()*(1+30)), '123476511', 4),
(4, 'Fato', 'Vendo fato para caça submarina', 'Desporto', FLOOR(RAND()*(1+30)), '123456722', 99),
(5, 'Rede para Peixe', 'Vendo rede de peixe para caça submarina', 'Desporto', FLOOR(RAND()*(1+30)), '123456722', 99),
(6, 'Vendo T3+4', 'Apartamento em ótimas condições - 150 000€', 'Imóveis', FLOOR(RAND()*(1+30)), '123456722', 99),
(7, 'Procura-se funcionário de restauração', 'Restaurante Sabor da Fruta 400€/mês', 'Emprego', FLOOR(RAND()*(1+30)), '123456722', 99),
(8, 'Vestido de cerimónia', 'Vendo vestido de cerimónia 15€', 'Moda', FLOOR(RAND()*(1+30)), '123456722', 99),
(9, 'Fato Homem', 'Vendo fato de homem 15€', 'Moda', FLOOR(RAND()*(1+30)), '123456722', 99),
(10, 'SweatShirt', 'Vendo SweatShirt 30€', 'Moda', FLOOR(RAND()*(1+30)), '123456722', 99),
(11, 'Nikon 3500-0', 'Vendo Nikon nova em folha - 500€', 'Tecnologia', FLOOR(RAND()*(1+30)), '123456722', 99),
(12, 'Vendo computador Lenovo', 'Vendo computador Lenovo como novo - 1550€', 'Tecnologia', FLOOR(RAND()*(1+30)), '123456722', 99),
(13, 'Máquina de aquecimento em banho maria', 'Máquina com 5 anos - 300€', 'Casa e Jardim', FLOOR(RAND()*(1+30)), '123456722', 99),
(14, 'Máquina de soldar', 'Máquina de soldar com 30 anos - 3500€', 'Casa e Jardim', FLOOR(RAND()*(1+30)), '123456722', 99),
(15, 'Guincho elétrico', 'Guincho novinho em folha - 2€', 'Casa e Jardim', FLOOR(RAND()*(1+30)), '123456722', 99),
(16, 'Telhas Diversas', 'Vendo telhas velhas novinhas em folha', 'Casa e Jardim', FLOOR(RAND()*(1+30)), '123456722', 99),
(17, 'Gerador', 'Gerador com muita capacidade de gasóleo', 'Casa e Jardim', FLOOR(RAND()*(1+30)), '123456722', 99),
(18, 'Compro discos de vinil', 'Compro discos de vinil de jazz moderno', 'Música', FLOOR(RAND()*(1+30)), '123456722', 99),
(19, 'Vendo discos de vinil', 'Tenho discos de vinil para despachar - 10€/unidade', 'Música', FLOOR(RAND()*(1+30)), '123456722', 99),
(20, 'Vendo guitarra acústica', 'Guitarra acústica com 30 anos - 450€', 'Música', FLOOR(RAND()*(1+30)), '123456722', 99),
(21, 'Amplificador Marshall XPT0', 'Vendo amplificador Marshall com 30 anos - 3500€', 'Música', FLOOR(RAND()*(1+30)), '123456722', 99),
(22, 'Gibson SG', 'Compro Gibson SG a bom preço', 'Música', FLOOR(RAND()*(1+30)), '123456722', 99),
(...)
```

Figura 21 - Povoamento da tabela Anúncio

```
-- Povoamento da tabela 'Editor'
INSERT INTO Editor
(ID, Nome)
VALUES
(1, 'José Carlos Malato'),
(2, 'Ferro Rodrigues'),
(3, 'Vasco Carvalho'),
(4, 'José Sampaio'),
(5, 'Rui Pires'),
(6, 'João Teixeira'),
(7, 'Carlos Costa'),
(8, 'Sandro Ferreira'),
(9, 'Melissa Bumstead'),
(10, 'Cristian Harbor'),
(11, 'Samuel Ferreira'),
(12, 'Miguel Mendes'),
(13, 'Luis Campelos'),
(14, 'Luanda Matos'),
(15, 'Regina Miguel'),
(16, 'Larissa Sampaio'),
(17, 'José Alves'),
(18, 'Miriam Pinto'),
(19, 'Rui Azevedo'),
(20, 'Barbara Cardoso');
```

Figura 22 - Povoamento da tabela Editor

```
-- Povoamento da tabela 'Revista'
INSERT INTO Revista
(ID, Empresa_ID, Nome, Edicao, Categoria, NrConsultas, Editor_ID)
VALUES
(1, 2, 'Fofuquices com a Vizinha', '2018-01-07', 'Ela', FLOOR(RAND()*(200)+799), 1),
(4, 2, 'Fofuquices com a Vizinha', '2018-04-07', 'Ela', FLOOR(RAND()*(200)+799), 1),
(7, 2, 'Fofuquices com a Vizinha', '2018-07-07', 'Ela', FLOOR(RAND()*(200)+799), 5),
(10, 2, 'Fofuquices com a Vizinha', '2018-10-07', 'Ela', FLOOR(RAND()*(200)+799), 4),
(12, 8, 'Visão', '2018-01-01', 'TV e Sociedade', FLOOR(RAND()*(200)+799), 2),
(13, 8, 'Visão', '2018-02-01', 'TV e Sociedade', FLOOR(RAND()*(200)+799), 3),
(14, 8, 'Visão', '2018-03-01', 'TV e Sociedade', FLOOR(RAND()*(200)+799), 9),
(15, 8, 'Visão', '2018-04-01', 'TV e Sociedade', FLOOR(RAND()*(200)+799), 20),
(16, 8, 'Visão', '2018-05-01', 'TV e Sociedade', FLOOR(RAND()*(200)+799), 14),
(17, 8, 'Visão', '2018-06-01', 'TV e Sociedade', FLOOR(RAND()*(200)+799), 2),
(18, 8, 'Visão', '2018-07-01', 'TV e Sociedade', FLOOR(RAND()*(200)+799), 2),
(19, 8, 'Visão', '2018-08-01', 'TV e Sociedade', FLOOR(RAND()*(200)+799), 3),
(20, 8, 'Visão', '2018-09-01', 'TV e Sociedade', FLOOR(RAND()*(200)+799), 3),
(21, 8, 'Visão', '2018-10-01', 'TV e Sociedade', FLOOR(RAND()*(200)+799), 3),
(22, 8, 'Visão', '2018-11-01', 'TV e Sociedade', FLOOR(RAND()*(200)+799), 3),
(31, 1, 'Sábado', '2018-01-11', 'TV e Sociedade', FLOOR(RAND()*(200)+799), 2),
(32, 1, 'Sábado', '2018-02-11', 'TV e Sociedade', FLOOR(RAND()*(200)+799), 3),
(33, 1, 'Sábado', '2018-03-11', 'TV e Sociedade', FLOOR(RAND()*(200)+799), 9),
(34, 1, 'Sábado', '2018-04-11', 'TV e Sociedade', FLOOR(RAND()*(200)+799), 20),
(35, 1, 'Sábado', '2018-05-11', 'TV e Sociedade', FLOOR(RAND()*(200)+799), 14),
(36, 1, 'Sábado', '2018-06-11', 'TV e Sociedade', FLOOR(RAND()*(200)+799), 6),
(37, 1, 'Sábado', '2018-07-11', 'TV e Sociedade', FLOOR(RAND()*(200)+799), 2),
(38, 1, 'Sábado', '2018-08-11', 'TV e Sociedade', FLOOR(RAND()*(200)+799), 3),
(39, 1, 'Sábado', '2018-09-11', 'TV e Sociedade', FLOOR(RAND()*(200)+799), 15),
(40, 1, 'Sábado', '2018-10-11', 'TV e Sociedade', FLOOR(RAND()*(200)+799), 12),
(41, 1, 'Sábado', '2018-11-11', 'TV e Sociedade', FLOOR(RAND()*(200)+799), 7),
(51, 8, 'Caras', '2018-01-24', 'TV e Sociedade', FLOOR(RAND()*(200)+799), 2),
(52, 8, 'Caras', '2018-02-24', 'TV e Sociedade', FLOOR(RAND()*(200)+799), 3),
(53, 8, 'Caras', '2018-03-24', 'TV e Sociedade', FLOOR(RAND()*(200)+799), 9),
(54, 8, 'Caras', '2018-04-24', 'TV e Sociedade', FLOOR(RAND()*(200)+799), 20),
(55, 8, 'Caras', '2018-05-24', 'TV e Sociedade', FLOOR(RAND()*(200)+799), 14),
(...)
```

Figura 23 - Povoamento da tabela Revista

```
-- Povoamento da tabela 'Artigos'
INSERT INTO Artigo
(ID, Revista_ID, Data, Categoria, Titulo, Corpo, NrConsultas)
VALUES
(1, 1, '2017-12-31', 'Saúde', 'Viva Melhor', '...', FLOOR(RAND()*(50)+100)),
(2, 1, '2017-01-02', 'Ela', 'As Melhores Fofocas', '...', FLOOR(RAND()*(50)+100)),
(3, 1, '2017-01-02', 'Ela', 'Cuidados do Corpo', '...', FLOOR(RAND()*(50)+100)),
(4, 1, '2017-12-28', 'Familia', 'Familia feliz', '...', FLOOR(RAND()*(50)+100)),
(5, 1, '2017-12-22', 'Saúde', 'Pele Bonita', '...', FLOOR(RAND()*(50)+100)),
(6, 1, '2017-12-09', 'Lazer', 'Passeio Rural', '...', FLOOR(RAND()*(50)+100)),
(7, 1, '2017-12-30', 'Lazer', 'Novas Diversões', '...', FLOOR(RAND()*(50)+100)),
(8, 4, '2017-04-01', 'Saúde', 'Viva Melhor', '...', FLOOR(RAND()*(50)+100)),
(9, 4, '2017-04-02', 'Ela', 'As Melhores Fofocas', '...', FLOOR(RAND()*(50)+100)),
(10, 4, '2017-04-03', 'Ela', 'Cuidados do Corpo', '...', FLOOR(RAND()*(50)+100)),
(11, 4, '2017-04-04', 'Familia', 'Familia feliz', '...', FLOOR(RAND()*(50)+100)),
(12, 4, '2017-04-05', 'Saúde', 'Pele Bonita', '...', FLOOR(RAND()*(50)+100)),
(13, 4, '2017-04-06', 'Lazer', 'Passeio Rural', '...', FLOOR(RAND()*(50)+100)),
(14, 4, '2017-04-06', 'Lazer', 'Novas Diversões', '...', FLOOR(RAND()*(50)+100)),
(15, 7, '2017-07-01', 'Saúde', 'Viva Melhor', '...', FLOOR(RAND()*(50)+100)),
(16, 7, '2017-07-02', 'Ela', 'As Melhores Fofocas', '...', FLOOR(RAND()*(50)+100)),
(17, 7, '2017-07-03', 'Ela', 'Cuidados do Corpo', '...', FLOOR(RAND()*(50)+100)),
(18, 7, '2017-07-04', 'Familia', 'Familia feliz', '...', FLOOR(RAND()*(50)+100)),
(19, 7, '2017-07-05', 'Saúde', 'Pele Bonita', '...', FLOOR(RAND()*(50)+100)),
(20, 7, '2017-07-06', 'Lazer', 'Passeio Rural', '...', FLOOR(RAND()*(50)+100)),
(21, 7, '2017-07-06', 'Lazer', 'Novas Diversões', '...', FLOOR(RAND()*(50)+100)),
(22, 10, '2017-10-01', 'Saúde', 'Viva Melhor', '...', FLOOR(RAND()*(50)+100)),
(23, 10, '2017-10-02', 'Ela', 'As Melhores Fofocas', '...', FLOOR(RAND()*(50)+100)),
(...)
```

Figura 24 - Povoamento da tabela Artigo

```
-- Povoamento da tabela 'Escritor'
INSERT INTO Escritor
  (ID, Nome)
VALUES
  (1 , 'Ezequiel Benancio'),
  (2 , 'Rogerio Furão'),
  (3 , 'Amilcar Lopez'),
  (4 , 'Carina Zimbabue'),
  (5 , 'Lurdes Ferreira'),
  (6 , 'Simão Sampaio'),
  (7 , 'Carlos Costa'),
  (8 , 'Felipe Sousa'),
  (9 , 'Megan Schmitz'),
  (10, 'Jenifer Cruz'),
  (11, 'Luis Reis'),
  (12, 'António Serafim'),
  (13, 'Gilberto Campos'),
  (14, 'Maria Antonieta'),
  (15, 'Francisco Pereira'),
  (16, 'Sebastião Morto'),
  (17, 'Miguel Meireles');
```

Figura 25 - Povoamento da tabela Escritor

```
-- Povoamento da tabela 'AnuncioRevista'
DELIMITER //
CREATE PROCEDURE Fill_AnuncioRevista()
BEGIN
  SET @es_total = 35;
  SET @cnt = 1;
  SET @LM = 1;
  WHILE (@cnt <= 1500) DO
    IF EXISTS (SELECT * FROM Revista WHERE ID = @cnt) THEN
      BEGIN
        WHILE (@LM < 10) DO
          SET @r = FLOOR(RAND()*@es_total+1);
          IF NOT EXISTS (SELECT * FROM AnuncioRevista WHERE (Revista_ID = @cnt) AND (Anuncio_ID = @r)) THEN
            BEGIN
              INSERT INTO AnuncioRevista
                (Anuncio_ID, Revista_ID)
              VALUES
                (@r, @cnt);
            END;
          END IF;
          SET @LM = @LM + 1;
        END WHILE;
      END;
    END IF;
    SET @cnt = @cnt + 1;
    SET @LM = 1;
  END WHILE;
END//

CALL Fill_AnuncioRevista();
```

Figura 26 - Povoamento da tabela AnuncioRevista



```

-- Povoamento da tabela 'EscritorArtigo'

DELIMITER //
CREATE PROCEDURE Fill_EscritorArtigo()
BEGIN
    SET @es_total = 17;
    SET @cnt = 1;
    SET @LM = 1;
    WHILE (@cnt <= 2000) DO
        IF EXISTS (SELECT * FROM Artigo WHERE ID = @cnt) THEN
            BEGIN
                WHILE (@LM < 4) DO
                    SET @r = FLOOR(RAND()*@es_total+1);
                    IF NOT EXISTS (SELECT * FROM EscritorArtigo WHERE (Artigo_ID = @cnt) AND (Escritor_ID = @r)) THEN
                        BEGIN
                            INSERT INTO EscritorArtigo
                                (Artigo_ID, Escritor_ID)
                                VALUES
                                    (@cnt, @r);
                        END;
                    END IF;
                    SET @LM = @LM + 1;
                END WHILE;
            END;
        END IF;
        SET @cnt = @cnt + 1;
        SET @LM = 1;
    END WHILE;
END//

CALL Fill_EscritorArtigo();

```

Figura 27 - Povoamento da tabela EscritorArtigo

## Lista de Siglas e Acrónimos

SQL      Structured Query Language