



Universidade do Minho

Mestrado Integrado em Engenharia Informática
Licenciatura em Ciências da Computação

Unidade Curricular de Bases de Dados

Ano Lectivo de 2017/2018

Base de Dados do Agrupamento de Escuteiros 424 – Nogueira.

**Frederico Pinto a73639,
Ricardo Canela, A74568
Ricardo Leal, A75411
Lucas Ribeiro, A68547**

Novembro, 2017

BD

Data de Recepção	
Responsável	
Avaliação	
Observações	

Base de Dados do Agrupamento de Escuteiros 424 – Nogueira.

**Frederico Pinto a73639,
Ricardo Canela, A74568
Ricardo Leal, A75411
Lucas Ribeiro, A68547**

Novembro, 2017

Resumo

Neste relatório é descrito, passo-a-passo, a criação de uma base de dados para o agrupamento 424 – Nogueira. Está dividido em 5 partes, sendo estas a contextualização da base de dados, o levantamento de requisitos, o modelo concetual, o modelo lógico antes e depois da normalização, e o respetivo modelo físico final.

O programa utilizado foi o SQL WorkBench.

Palavras-Chave: Bases de Dados Relacionais, modelo conceptual, lógico e físico, entidades, atributos, relacionamentos, MySQL.

Índice

Resumo	i
Índice	ii
Índice de Figuras	iv
Índice de Tabelas	v
1.1. Contextualização	1
1.2. Fundamentação da Implementação da base de dados	2
1.3. Análise de Viabilidade do Processo	2
2.1. Método de Levantamento e de Análise de Requisitos	4
2.2. Requisitos Levantados	4
2.3. Análise Geral dos Requisitos	5
3.1. Apresentação da Abordagem de Modelação Realizada	6
3.2. Identificação e Caracterização das Entidades do Problema	6
3.3. Identificação dos Relacionamentos no Modelo	7
3.4. Identificação e Associação de Atributos com Entidades e Respetivos Domínios de Valores	8
3.5. Determinação das Chaves Candidatas e Chaves Primárias	11
3.5.1 Atributos	11
3.5.2 Chaves Candidatas e Chaves Primárias	12
3.6. Diagrama ER	13
3.6.1 Explicação do Diagrama ER	13
4.1. Construção do Modelo de Dados Lógico	15
4.2. Desenho do Modelo Lógico	16
4.3. Validação do Modelo através da Normalização	16
4.4. Validação do Modelo com Interrogações do Cliente	17
4.5. Validação do Modelo com Transações Estabelecidas	18
4.6. Revisão do Modelo Lógico com o Cliente	18
5.2. Tradução do Esquema Lógico para MySQL	19
5.3. Tradução das Interrogações do Utilizador	24
5.4. Tradução das Transações Estabelecidas	26
5.5. Tradução das Transações Estabelecidas	27
5.6. Definição e Caracterização das Vistas de Utilização	33

5.7. Definição e Caracterização dos Mecanismos de Segurança	34
5.8. Revisão do Sistema Implementado	34

Índice de Figuras

Figura 1 - Modelo Concetual	13
Figura 2 - Primeiro Modelo Lógico	16
Figura 3 - Modelo Lógico Final	17
Figura 4 - Código de Criação da Tabela Seccao	20
Figura 5 -Código de Criação da Tabela Equipa	20
Figura 6 -Código de Criação da Tabela Funcao	21
Figura 7 -Código de Criação da Tabela Elemento	23
Figura 8 -Código de Criação da Tabela Atividade	24
Figura 9 - Código da Resolução da Interrogação	24
Figura 10 -Código da Resolução da Interrogação	25
Figura 11 - Código da Resolução da Interrogação	25
Figura 12 - Código da Inserção de uma nova Atividade	26
Figura 13 - Código da Inserção de um novo Elemento	26
Figura 14 - Código da Arquivção de um dado Elemento	27
Figura 15 - Vista Chefes	33
Figura 16 - Código de Criação e Respetivos Previlégios dos três Utilizadores	34

Índice de Tabelas

Tabela 1 - Dicionário de Dados das Entidades	6
Tabela 2 - Dicionário de Relacionamentos do Modelo	8
Tabela 3 - Dicionário dos Atributos	9
Tabela 4 - Espaço Ocupado pela Entidade Elemento	27
Tabela 5 - Espaço Ocupado pela Entidade Seccao	28
Tabela 6 -Espaço Ocupado pela Entidade Equipa	29
Tabela 7 - Espaço Ocupado pela Entidade Atividade	30
Tabela 8 - Espaço Ocupado pela Entidade Funcao	30
Tabela 9 - Estimativa do Tamanho da Base de Dados	31
Tabela 10 - Estimativa do Tamanho da Base de Dados após um ano	33

1. Introdução

No contexto da Unidade Curricular de Base de Dados, foi-nos solicitado pelo professor da Unidade Curricular o desenvolvimento de um sistema de Base de Dados de um ambiente à nossa escolha que refletisse uma situação real e com complexidade suficiente não só para explorar a matéria abordada nas aulas, como também explorar outros mecanismos que encontraríamos a um nível mais profissional. Com influência de um dos membros que pertence aos escuteiros e deste fazer referência às dificuldades organizacionais desta instituição, decidimos explorar esta situação no nosso trabalho e desenvolver uma Base de Dados de um Agrupamento de Escuteiros.

O ciclo de desenvolvimento de uma Base de Dados tem etapas bem definidas como apresentamos neste relatório, começando no primeiro capítulo pela contextualização do problema de seguida é feita uma recolha e análise de requisitos. Efetuadas essas etapas partimos para a construção do modelo concetual, (primeira etapa do Database Design), que é caracterizada pela identificação e análise das entidades, relacionamentos, associações e chaves. Posteriormente, segunda etapa do Database Design, efetuamos o processo de construção do modelo lógico testando e validando os requisitos do utilizador, usando a técnica de normalização para eliminar redundâncias e incorreções lógicas. Por fim, utilizamos o MySQL para traduzir o modelo lógico num modelo físico.

1.1. Contextualização

Desde a origem do ser humano como o conhecemos que o seu principal instinto foi sobreviver. Inicialmente recolhendo recursos e caçando, viajando à procura destes suprimentos, mais tarde com a revolução agrícola foi possível assentar-se no mesmo local levando ao surgimento da civilização. Mesmo com esta transformação, o instinto de sobrevivência manteve-se vivo com a necessidade de explorar e disputar territorialmente e religiosamente. O início da revolução industrial deu origem a uma era moderna que continua a expandir-se e que atualmente se caracteriza pela falta de contacto com a realidade e natureza, derivada da não necessidade de sair de casa para prestar ou receber serviços. O adolescente que à 2000 anos atrás era um perito em sobrevivência e luta, evoluiu no caminho do sedentarismo e perda de várias técnicas de sobrevivência.

No ano de 1907 na ilha de Brownsea no Reino Unido aconteceu o primeiro acampamento escotista. Foi promovido por Baden-Powell com a intenção de ensinar a um grupo de jovens

entre 12 e 16 anos técnicas de observação, segurança, orientação e socorrismos. O escotismo cresceu tão rápido que passados dois anos, em 1909, aconteceu a primeira concentração de Escotas, estando presentes de 11.000 Escotas em Crystal Palace – Londres. Posteriormente Baden-Powell (BP) deixa o Exército para se dedicar inteiramente ao Escotismo.

Já em Portugal, o Escutismo teve os primeiros passos no ano de 1911. Em 1912 é fundada em Lisboa a Associação dos Escoteiros de Portugal.

A Igreja Católica idealizou criar outra associação escotista, mas com caráter uniconfessional, destinada exclusivamente aos jovens que professavam a religião Católica Romana, portanto a 27 de Maio de 1923 foi fundado em Braga o Corpo Nacional de Escutas (CNE) pelo Arcebispo de Braga, D. Manuel Vieira de Matos. Assim, nasceram os escuteiros.

A Missão do Escutismo consiste em contribuir para a educação dos jovens, partindo dum sistema de valores enunciado na Lei e na Promessa escutistas, ajudando a construir um mundo melhor, onde as pessoas se sintam plenamente realizadas como indivíduos e desempenhem um papel construtivo na sociedade. Isto é alcançado:

- envolvendo os jovens, ao longo dos seus anos de formação, num processo de educação não-formal;
- utilizando um método original, segundo o qual cada indivíduo é o principal agente do seu próprio desenvolvimento, para se tornar uma pessoa autónoma, solidária, responsável e comprometida.
- ajudando os jovens na definição de um sistema de valores baseado em princípios espirituais, sociais e pessoais expressos na Promessa e na Lei.

Em 1983, o CNE é declarado Instituição de Utilidade Pública e passados vinte anos, em 2003 já existiam 30 milhões de escuteiros em todo o mundo!

1.2. Fundamentação da Implementação da base de dados

Gerir um Agrupamento de Escuteiros que tenha um tamanho considerável de elementos começa a ser trabalhoso se este for feito a nível manuscrito. A probabilidade de extravio/perda de informação é demasiado alta para a importância que ela tem na gestão do Agrupamento. Com intuito de melhorar a situação, o Chefe de Agrupamento 424- Nogueira, decidiu investir algum do dinheiro do Agrupamento na criação de uma base de dados para a gestão do mesmo. O objetivo da implementação é armazenar toda a informação relevante sobre o Agrupamento.

1.3. Análise de Viabilidade do Processo

Um Sistema de Base de Dados Relacional é essencial se o objetivo é obtermos uma solução em que haja organização e integridade da informação que é tratada. Numa organização como o nosso caso de estudo torna-se evidente a importância desta. Tarefas como pedidos de acesso a informação, alteração e gestão de dados entre outras, ficam portanto facilitadas como

a implementação de uma estrutura em que os dados inseridos são basicamente atômicos e independentes entre si.

Consideramos assim que o processo viável e bem justificado para o nosso caso de estudo.

2. Levantamento e Análise de Requisitos

2.1. Método de Levantamento e de Análise de Requisitos

A base de dados que estamos a implementar, tem como finalidade armazenar e gerir dados referentes a um agrupamento de escuteiros de modo a melhorar a sua organização. Desta forma foi necessário discutir com o cliente as necessidades que pretende suprimir com a sua construção, com isto queremos dizer que para cada particularidade da BD quais são as ações possíveis de realizar.

O cliente definiu que a base de dados deveria estar estruturada/funcionar um pouco “à imagem” de uma organização hierárquica dos Escuteiros.

Este Agrupamento terá que guardar informações sobre quatro secções às quais os elementos podem pertencer. Estas secções têm nomes definidos: Lobitos, Exploradores, Pioneiros e Caminheiros e dentro de cada uma delas existe um número ilimitado de equipas.

Para além disso, o cliente exige que seja possível armazenar informação sobre as atividades realizadas pelas equipas e que todos os elementos têm que ter um cargo. Deseja também que seja possível inserir e retirar um elemento da base de dados e adicionar uma atividade.

O cliente quer que a base de dados responda a várias perguntas desde, “Quantos elementos tem o Agrupamento” até “Qual o nome e cargo de um elemento que pertence a uma dada secção e participou em determinada atividade”.

2.2. Requisitos Levantados

Apresentamos agora os requisitos do funcionamento do Agrupamento:

Cada Elemento:

- Pertence a uma secção (com exceção dos que possuem um determinado cargo);
- Tem associado uma série de informações pessoais: Nome, Idade, Sexo, Data de Nascimento, Morada, Telemóvel, Email, Equipa a que pertence bem como o número de noites de campo;
- Tem um número de identificação único (NIN);
- Possui um cargo de entre os seguintes:

- Chefe de Agrupamento-Cargo mais alto dentro do Agrupamento;
- Chefe Tesoureiro-Chefe de contas;
- Chefe Secretário-Chefe da papelada;
- Chefe dos Lobitos-Líder dos Lobitos;
- Chefe dos Exploradores-Líder dos Exploradores;
- Chefe dos Pioneiros-Líder dos Pioneiros;
- Chefe dos Caminheiros-Líder dos Caminheiros;
- Relações Públicas-Trata das relações públicas;
- Cozinheiro-Trata das refeições;
- Animador-Anima a equipa;
- Guia-Coordenador da equipa;
- Tesoureiro-Gere o dinheiro da equipa;
- Chefe de Secção-Chefe de uma determinada secção;
- Secretário-Trata da papelada da equipa;
- Guarda-material-Responsável pelo material da equipa;
- Socorrista-Cuida das questões de saúde;

Cada Equipa:

- É constituída por elementos;
- Tem associado o seu Nome, Lema e Grito e a Secção a que pertence;
- Tem atribuído um número de identificação único;

Cada Secção:

- O número de secções está restringida a 4, estas representam um nome de identificação único;- Designação de entre os seguintes: “Lobitos”(entre 6 e 10 anos), “Caminheiros”(mais de 18 anos), “Exploradores” (entre os 11 e 14 anos)e “Exploradores”(15 aos 17 anos) ;
- É constituída por Equipas;
- Tem associado uma Simbologia e um patrono;

Cada Atividade:

- Tem um nome, descrição e data;
- Tem atribuído um número de identificação único;

2.3. Análise Geral dos Requisitos

Depois de efetuado uma nova reunião com o nosso cliente, mostramos os requisitos numa abordagem para base de dados, este aprovou-os. Obtemos a autorização para avançar para a realização do modelo concetual.

3. Modelo Conceitual

3.1. Apresentação da Abordagem de Modelação Realizada

O processo de modelação conceptual é a primeira fase da metodologia para a criação de um sistema de base de dados (SGBD) e inicia-se após uma análise dos requisitos.

Como o sistema de base de dados a implementar não é complexo, decidimos abordar a modelação de uma maneira centralizada.

3.2. Identificação e Caracterização das Entidades do Problema

Como foi dito anteriormente, o primeiro passo para a construção do nosso modelo concetual é a identificação das entidades do problema, isto é, os principais componentes para a estruturação da nossa base de dados. Para se conseguir identificar com sucesso as entidades é necessária uma observação cuidadosa, pois embora se tenha um objeto com uma grande importância para o modelo, este pode não ser necessariamente uma entidade.

Neste capítulo irá ser apresentado o Dicionário de dados das entidades do modelo:

Tabela 1 - Dicionário de Dados das Entidades

<i>Entidade</i>	<i>Descrição</i>	<i>Ocorrência</i>
Elemento	Entidade representativa da pessoa que pertence ao Agrupamento.	O Elemento é uma entidade fundamental pois é ele quem usufrui do Agrupamento.

Secção	Entidade representativa da secção a que determinado Elemento pertence.	A Secção é também uma entidade muito importante pois divide os Elementos pelas Secções. Sem ela, o funcionamento do Agrupamento não seria o suposto.
Equipa	Entidade representativa de uma Equipa do Agrupamento que pertence a uma dada Secção e que tem certos Elementos.	A Equipa é também uma entidade importante pois divide os Elementos por Equipas dentro de uma dada secção. Sem ela, o funcionamento do <u>Agrupamento</u> não seria o suposto.
Atividade	Entidade representativa de uma atividade efetuada por uma dada Equipa que consequentemente pertence a uma Secção e tem certos Elementos.	A Atividade é efetuada pela Equipa.

3.3. Identificação dos Relacionamentos no Modelo

Após se ter conseguido identificar todas as entidades no modelo, proceder-se-á agora à identificação dos relacionamentos entre entidades.

Elemento e Secção:

O relacionamento “pertence” entre a entidade Elemento e a entidade Secção caracterizam-se por apresentar uma cardinalidade de 1 para N, ou seja, um Elemento pertence apenas a uma Secção, mas a uma Secção pertencem N Elementos.

Secção e Equipa:

O relacionamento “pertence” entre a entidade Secção e a entidade Equipa caracterizam-se por apresentar uma cardinalidade de 1 para N, ou seja, uma Secção pode pertencer a N Equipas, mas cada Equipa só pode pertencer a uma Secção.

Elemento e Equipa:

O relacionamento “pertence” entre a entidade Elemento e a entidade Equipa caracterizam-se por apresentar uma cardinalidade de 1 para N, na medida em que um Elemento apenas pode pertencer a uma Equipa, mas a uma equipa podem pertencer N Elementos.

Equipa e Atividade:

O relacionamento “efetuou” entre a entidade Equipa e a entidade Atividade caracterizam-se por apresentar uma cardinalidade de 1 para N, portanto, uma Equipa tem N Atividades mas uma Atividade apenas pertence a uma Equipa.

Tabela 2 - Dicionário de Relacionamentos do Modelo

<i>Entidade</i>	<i>Multiplicidade</i>	<i>Relacionamento</i>	<i>Multiplicidade</i>	<i>Entidade</i>
<i>Secção</i>	<i>1</i>	<i>Tem</i>	<i>N</i>	<i>Elemento</i>
<i>Secção</i>	<i>1</i>	<i>É composta</i>	<i>N</i>	<i>Equipa</i>
<i>Equipa</i>	<i>1</i>	<i>Possui/efectua</i>	<i>N</i>	<i>Atividade</i>

3.4. Identificação e Associação de Atributos com Entidades e Respetivos Domínios de Valores

Depois de se ter identificado todas as entidades do problema e posteriormente as ter relacionado entre si, chegou-se agora ao momento de identificar todas os seus atributos que se considerou importantes para o problema.

Tabela 3 - Dicionário dos Atributos

Entidade	Atributo	Descrição	Tipo de dados e tamanho	Null	Tipo de atributo	Domínio
Elemento	<u>Nin</u>	Número de Identificação dos Escuteiros	INT	Não	Chave primária	Inteiro
	Nome	Nome do Elemento	VARCHAR (45)	Não	Simples	45 char. variáveis
	Idade	Idade do Elemento	INT	Não	Simples	Inteiro
	Sexo	Sexo do Elemento ('M','F')	VARCHAR (1)	Não	Simples	1 char. variáveis
	DatadeNasc	Data de Nascimento do Elemento	DATE	Não	Simples	'ANO-MÊS- DIA'
	Funcao	Função dentro do Agrupamento	VARCHAR(100)	Não	Simples	100 char. variáveis
	Morada	Morada do Elemento	VARCHAR (45)	Não	Simples	45 char. variáveis
	Telemovel	Nº de telemóvel do Elemento	INT	Não	Simples	Inteiro
	Email	Email do		Sim	Simples	45 char.

	NoitesdeCampo	Elemento Quantidade de noite que o Elemento passou no campo	VARCHAR(45) INT	Não	Simples	variáveis Inteiro
Secção	<u>Designacao</u>	Nome da Secção	VARCHAR (45)	Não	Chave primária	45 char. variáveis
	Simbologia	Simbologia da Secção	VARCHAR (100)	Sim	Simples	100 char. variáveis
	Patrono	Padroeiro da Secção	VARCHAR (45)	Não	Simples	45 char. variáveis
Equipa	<u>idEquipa</u>	Nº identificador da Equipa	INT	Não	Chave primária	Inteiro
	Nome	Nome da Equipa	VARCHAR (45)	Não	Simples	45 char. variáveis
	Lema	Lema da Equipa	VARCHAR (45)	Não	Simples	45 char. variáveis
	Grito	Grito da	VARCHAR (45)	Não	Simples	45 char.

		Equipa				variáveis
Atividade	<u>idAtividade</u>	Número que identifica uma atividade	INT	Não	Chave primária	Inteiro
	Designacao	Nome da atividade	VARCHAR (40)	Não	Simples	40 char. variáveis
	Descricao	Descrição da Atividade	VARCHAR (150)	Não	Simples	150 char. variáveis
	dia	Data da Atividade	DATE	Não	Simples	'ANO-MÊS-DIA'

3.5. Determinação das Chaves Candidatas e Chaves Primárias

3.5.1 Atributos

Elemento:

- **Nin** -> nº de identificação do escuteiro;
- **Nome** -> Nome do elemento;
- **Idade** -> Idade do elemento;
- **Sexo** -> Sexo do elemento;
- **DatadeNasc** -> Data de nascimento do elemento;
- **Morada** -> Morada do elemento;
- **Telemovel** -> Número de telemovel do elemento;
- **Email** -> Email do elemento;
- **NoitesdeCampo** -> Número de noites que o elemento passou no campo;
- **Funcao** -> Função dentro do Agrupamento

Secção:

- **Designacao** -> Nome da Secção;
- **Simbologia** -> Simbologia da Secção;
- **Patrono** -> Padroeiro da Secção;

Equipa:

- **IdEquipa** -> Número identificador da equipa;
- **Nome** -> Nome da Equipa;
- **Lema** -> Lema da Equipa;
- **Grito** -> Grito da Equipa;

Atividade:

- **IdAtividade** -> Número identificador da atividade;
- **Designacao** -> Nome da atividade;
- **Descricao** -> Descrição da atividade;
- **Dia** -> Data da atividade;

3.5.2 Chaves Candidatas e Chaves Primárias

Após se ter identificado e compreendido todas as entidades e os seus atributos tem se agora o momento de determinar quais desses atributos poderão constituir chaves primárias.

Uma chave primária corresponde ao atributo de uma dada entidade que é capaz de a identificar num registo que a represente, por outro lado, ao conjunto de todos os atributos de uma entidade chama-se chaves candidatas.

Elemento:

Chaves Candidatas: {Nin};

Chave Primária: O Nin é a única chave candidata que é única e invariável para cada Elemento, portanto o Nin é a chave primária.

Secção:

Chaves Candidatas: {Designação, Patrono}

Chave Primária: ambas as chaves candidatas são únicas e invariáveis para cada Secção, mas elegemos a Designação como chave primária.

Equipa:

Chave Candidatas: {idEquipa}

Chave Primária: O idEquipa é a única chave candidata que é única e invariável para cada Equipa, logo o idEquipa é a chave primária;

Atividade:

Chaves Candidatas: {idAtividade}

Chave Primária: O idAtividade é a única chave candidata que é única e invariável para cada Atividade, logo o idAtividade é a chave primária;

3.6. Diagrama ER

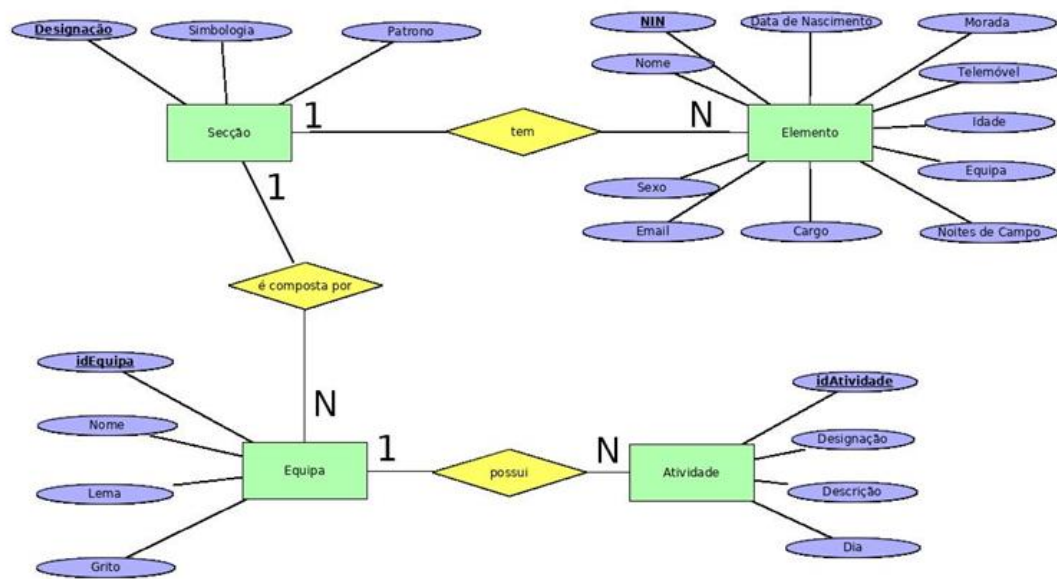


Figura 1 - Modelo Conceitual

3.6.1 Explicação do Diagrama ER

A entidade Elemento é identificado pelo Nin, nome, telemóvel e cargo. Para além disso tem também informações adicionais como email, noites de campo, morada, data de nascimento, sexo. Esta entidade está relacionada com a entidade Equipa e entidade Secção.

A Secção é identificada pela designação e tem mais alguns atributos como patrono e simbologia. Para além disso esta entidade está relacionada com a entidade Elemento e a entidade Equipa.

A Equipa é identificada pelo idEquipa e para além disso tem como atributos o nome, o lema e o grito. Está também relacionada com a entidade Secção e com a entidade Atividade. A entidade Atividade é identificada pelo idAtividade e tem também como atributos a designação, a descrição e o dia. Está relacionada apenas com a entidade Equipa.

Após a explicação do diagrama ao cliente, este validou os requisitos e modelo concetual. O que nos permitiu avançar para a próxima fase do nosso projeto.

4. Modelo Lógico

Concluída a conceptualização do problema é agora momento de traduzir esse mesmo modelo num modelo lógico.

4.1. Construção do Modelo de Dados Lógico

Para construir o modelo lógico tivemos primeiro que visualizar o nosso modelo concetual e voltar a verificar as entidades, atributos e relacionamentos existentes.

As entidades presentes no modelo concetual permaneceram as mesmas do modelo lógico:

- **Elemento**
- **Secção**
- **Equipa**
- **Atividade**

Posteriormente verificamos os relacionamentos entre entidades, encontrando apenas um tipo de relacionamento.

- Relacionamentos binários de um para muitos (1->N).

Este relacionamento tem a particularidade de que a entidade com multiplicidade N, possui uma chave estrangeira, que vai ser a chave primária da entidade de multiplicidade 1.

- Elemento (Nin, Nome, (...), Equipa_idEquipa, Seccao_Designacao)

Tem como chave primária: Nin;

Tem como chave estrangeira: Equipa_idEquipa(Entidade Equipa) e Seccao_Designacao(Entidade Secção)

- Equipa (idEquipa, Nome, (...), Seccao_Designacao)

Tem como chave primária: idEquipa;

Tem como chave estrangeira: Seccao_Designacao;

- Atividade (idAtividade, Designacao, (...), Equipa_idEquipa)

Tem como chave primária: idAtividade;

Tem como chave estrangeira: Equipa_idEquipa;

4.2. Desenho do Modelo Lógico

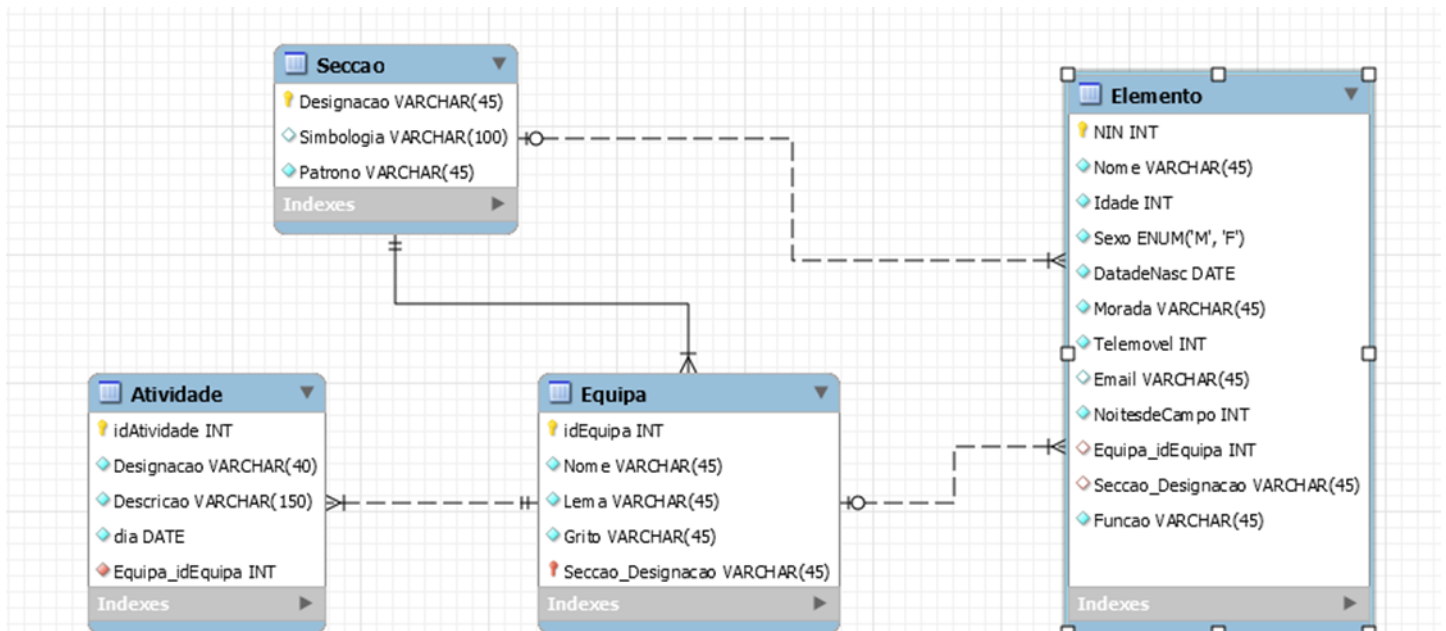


Figura 2 - Primeiro Modelo Lógico

4.3. Validação do Modelo através da Normalização

Ao analisarmos o nosso modelo até à terceira forma normal deparamo-nos com uma anomalia.

O nosso primeiro modelo não está de acordo com a primeira forma da normalização.

A primeira forma normal diz que o domínio de cada atributo de uma dada relação apenas pode conter valores atômicos, e o valor de cada atributo é constituído apenas por um simples valor do seu domínio. No nosso primeiro modelo, na entidade Elemento, encontramos um atributo que não é atômico. O atributo Funcao assume valores do tipo “Chefe de Lobitos” ou ainda “Chefe de Pioneiros”, e este atributo não é atômico porque o poderíamos partir em Chefe e Lobitos ou “Chefe” e “Pioneiros”. Para a resolução do problema, teríamos que adicionar uma secção extra, esta de nome ‘Dirigentes’ ou ‘Chefes’ e um relacionamento opcional de “chefia” entre as entidades Elemento e Seccao, passando assim o relacionamento já existente a obrigatório e este sendo um relacionamento de “pertença”.

A segunda forma normal diz que numa tabela, nenhum atributo que não é chave é funcionalmente dependente num subconjunto próprio de qualquer chave candidata. Como o nosso modelo não tem chaves compostas, este cumpre a segunda forma normal, assumindo que cumpre a primeira, excetuando o problema anteriormente descrito.

A terceira forma normal diz que todo o atributo que não é chave de uma relação não é transitivamente dependente de uma chave candidata de uma tabela. Por isso, se assumirmos que o nosso modelo está de acordo com a segunda forma normal, também vai estar de acordo

com a terceira forma normal, já que não existem atributos não-chave que são transitivamente dependentes de uma chave candidata dessa tabela.

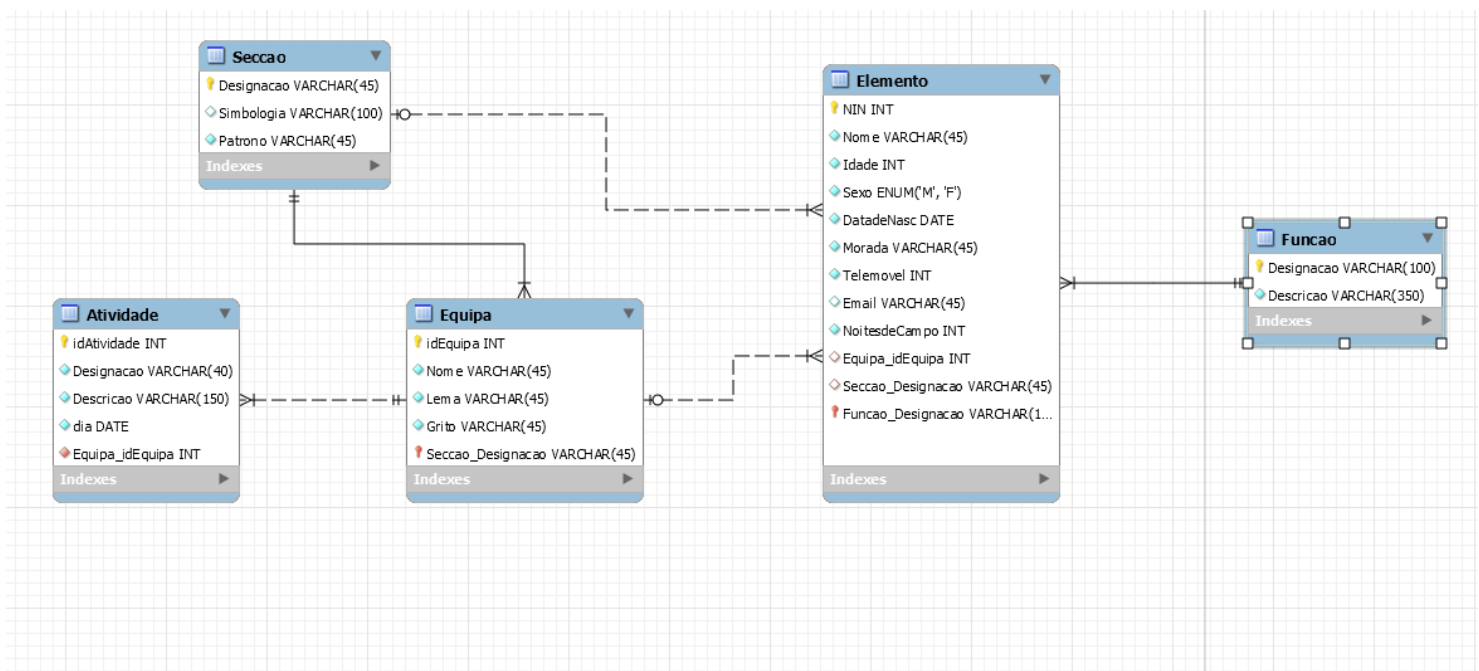


Figura 3 - Modelo Lógico Final

4.4. Validação do Modelo com Interrogações do Cliente

Outra maneira de obter a validação do modelo é com interrogações à nossa base de dados, ou seja, vamos fazer determinadas questões para obter informação mais específica para melhor perceber o seu funcionamento e verificar se o modelo tem problemas.

Fizemos cinco questões, de acordo com os pedidos do cliente, efetuadas numa fase anterior:

- **Qual é o Nin, Nome, Função e Secção de todos os elementos do Agrupamento?**

A determinação do Nin, Nome, Função e Secção é feita a partir da tabela Elemento.

- **Qual o número de elementos do Agrupamento?**

O número de elementos do Agrupamento é determinado também a partir da tabela Elemento.

- **Quais as funções existentes numa dada secção?**

Para determinar as funções de uma dada secção temos que recolher dados as tabelas Seccao, Elemento e Funcao. A relação Seccao e Elemento permite verificar se o elemento era da secção desejada pelo utilizador. A relação Elemento e Funcao permite determinar as funções.

- **Qual o Patrono de todas as secções que foram representadas em determinada atividade por uma equipa?**

Para determinar a resposta a esta pergunta, temos que recolher dados da tabela Seccao, Equipa e Atividade. Com a relação Seccao e Equipa retiramos as equipas e na relação Equipa e Atividade conseguimos descobrir as equipas que pertencem a determinada secção e participaram na atividade e posteriormente descobrir os patronos das secções.

- **Qual é o nome e respetivas funções dos elementos de uma determinada secção que fizeram determinada atividade?**

Para devolver o nome e a função dos elementos temos primeiro que recolher dados da tabela Funcao, Seccao, Atividade e Elemento. Com a relação Funcao e Elemento conseguimos recolher as funções dos elementos, de seguida a relação Elemento e Seccao permitem verificar se a secção do elemento é a secção pretendida pelo utilizador e finalmente a relação Elemento e Atividade permitem nos descobrir se a designação da atividade é também a desejada pelo Utilizador. Permitindo determinar qual o nome e a função dos elementos.

4.5. Validação do Modelo com Transações Estabelecidas

Como foi estabelecido com o cliente, temos que verificar se o modelo suporta as transações propostas previamente.

- **Inserção de uma nova atividade;**

Para executar esta operação só é preciso obter os dados corretos para a criação de um novo valor na tabela Atividade.

- **Inserção de um novo elemento;**

Para conseguir a inserção de um novo elemento através da criação de um novo valor na tabela Elemento é preciso fornecer apenas Nin, Nome, Idade, Sexo, Data de Nascimento, Telemóvel, Email, Noites de Campo e Função.

- **Arquivar os dados de um elemento;**

Para efetuar com sucesso o processo de arquivar um elemento do Agrupamento deve fornecer o Nin.

4.6. Revisão do Modelo Lógico com o Cliente

Nesta fase do modelo lógico foi necessário voltar a reunir com o cliente. Com a normalização do modelo(assumindo que esta está normalizada), apresentamos a nossa solução e explicamos a situação em que nos encontrávamos. Revendo todos os requisitos propostos pelo cliente através das interrogações e transações apresentadas, e após confrontarmos o cliente com o problema antes referido, este decidiu avançar com o projeto.

5. Modelo Físico

5.1. Seleção do Sistema de Gestão de Bases de Dados

Após a reunião em que o nosso modelo lógico foi aceite, avançamos para a tradução do mesmo para o respetivo modelo físico usando um Sistema de Gestão de Base de Dados. Com isto os nossos dados passam a estar organizados em tabelas que na realidade não são tabelas, mas sim armazenados em ficheiros na memória.

Para o nosso projeto utilizamos como SGBD o MySQL que utiliza o motor de busca InnoDB. Algumas das razões pela qual optamos pelo MySQL foram:

- A sua alta performance;
- Pelo seu nível de segurança, sendo adotado em empresas por todo o mundo;

5.2. Tradução do Esquema Lógico para MySQL

Relações Base:

Relação Secção

Domínio Designacao	String com tamanho 45
Domínio Simbologia	String com tamanho 100
Domínio Patrono	String com tamanho 45

Seccao (

Designacao	Nome da Secção	NOT NULL,
Simbologia	Simbologia da Secção	NULL,
Patrono	Padroeiro daSecção	NOT NULL,

PRIMARY KEY(Designacao));

```

-----
-- Table `Agrup424`.`Seccao`
-----
CREATE TABLE IF NOT EXISTS `Agrup424`.`Seccao` (
  `Designacao` VARCHAR(45) NOT NULL,
  `Simbologia` VARCHAR(100) NULL,
  `Patrono` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`Designacao`),
  UNIQUE INDEX `Designacao_UNIQUE` (`Designacao` ASC))
ENGINE = InnoDB;

```

Figura 4 - Código de Criação da Tabela Seccao

Relação Equipa

Domínio idEquipa	Int
Domínio Nome	String com tamanho 45
Domínio Lema	String com tamanho 45
Domínio Grito	String com tamanho 45
Domínio Seccao_Designacao	String com tamanho 45

Equipa(

IdEquipa	Id da Equipa	NOT NULL,
Nome	Nome da Equipa	NOT NULL,
Lema	Lema da Equipa	NOT NULL,
Grito	Grito da Equipa	NOT NULL,
Seccao_Designacao	Nome da Secção	NOT NULL,

PRIMARY KEY(idEquipa)),

FOREIGN KEY(Seccao_Designacao) REFERENCES Seccao(Designacao);

```

-----
-- Table `Agrup424`.`Equipa`
-----
CREATE TABLE IF NOT EXISTS `Agrup424`.`Equipa` (
  `idEquipa` INT NOT NULL,
  `Nome` VARCHAR(45) NOT NULL,
  `Lema` VARCHAR(45) NOT NULL,
  `Grito` VARCHAR(45) NOT NULL,
  `Seccao_Designacao` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`idEquipa`, `Seccao_Designacao`),
  UNIQUE INDEX `idEquipa_UNIQUE` (`idEquipa` ASC),
  INDEX `fk_Equipa_Seccao1_idx` (`Seccao_Designacao` ASC),
  CONSTRAINT `fk_Equipa_Seccao1`
    FOREIGN KEY (`Seccao_Designacao`)
      REFERENCES `Agrup424`.`Seccao` (`Designacao`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

Figura 5 -Código de Criação da Tabela Equipa

Relação Função

Domínio Designacao	String com tamanho 100
Domínio Descricao	String com tamanho 350

Funcao(

Designacao	Função	NOT NULL,
Descricao	Descrição da Função	NOT NULL,

PRIMARY KEY(Designacao));

```
-- Table `Agrup424`.`Funcao`  
-----  
CREATE TABLE IF NOT EXISTS `Agrup424`.`Funcao` (  
  `Designacao` VARCHAR(100) NOT NULL,  
  `Descricao` VARCHAR(350) NOT NULL,  
  PRIMARY KEY (`Designacao`),  
  UNIQUE INDEX `Descricao_UNIQUE` (`Descricao` ASC),  
  UNIQUE INDEX `Designacao_UNIQUE` (`Designacao` ASC))  
ENGINE = InnoDB;
```

Figura 6 -Código de Criação da Tabela Funcao

Relação Elemento

Domínio NIN	Int
Domínio Nome	String com tamanho 45
Domínio Idade	Int
Domínio Sexo	Enumeração
Domínio DatadeNasc	Data no formato “YYYY-MM-DD”
Domínio Morada	String com tamanho 45
Domínio Telemovel	Int
Domínio Email	String com tamanho 45
Domínio NoitesdeCampo	Int
Domínio Equipa_idEquipa	Int
Domínio Seccao_Designacao	String com tamanho 45
Domínio Funcao_Designacao	String com tamanho 100

```

Elemento(
  NIN                Id do elemento                NOT NULL,
  Nome               Nome do elemento              NOT NULL,
  Idade              Idade do Elemento              NOT NULL,
  Sexo               Sexo do Elemento              NOT NULL,
  DatadeNasc         Data de Nascimento do Elemento      NOT NULL,
  Morada             Morada do Elemento            NOT NULL,
  Telemovel          Número de Telemovel do Elemento     NOT NULL,
  Email              Email do Elemento              NULL,
  Noitesdecampo      Noites de Campo do Elemento      NOT NULL,
  Equipa_idEquipa     Id da Equipa do Elemento          NULL,
  Seccao_Designacao  Nome da Secção      do Elemento    NULL,
  Funcao_Designacao  Função      do Elemento            NOT NULL,

  PRIMARY KEY(NIN)),
  FOREIGN KEY(Equipa_idEquipa) REFERENCES Equipa(idEquipa);
  FOREIGN KEY(Seccao_Designacao) REFERENCES Seccao(Designacao);
  FOREIGN KEY(Funcao_Designacao) REFERENCES Funcao(Designacao);

```

```

-----
-- Table `Agrup424`.`Elemento`
-----
CREATE TABLE IF NOT EXISTS `Agrup424`.`Elemento` (
  `NIN` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `Nome` VARCHAR(45) NOT NULL,
  `Idade` INT NOT NULL,
  `Sexo` ENUM('M', 'F') NOT NULL,
  `DatadeNasc` DATE NOT NULL,
  `Morada` VARCHAR(45) NOT NULL,
  `Telemovel` INT NOT NULL,
  `Email` VARCHAR(45) NULL,
  `NoitesdeCampo` INT NOT NULL,
  `Equipa_idEquipa` INT NULL,
  `Seccao_Designacao` VARCHAR(45) NULL,
  `Funcao_Designacao` VARCHAR(100) NOT NULL,
  PRIMARY KEY (`NIN`, `Funcao_Designacao`),
  UNIQUE INDEX `NIN_UNIQUE` (`NIN` ASC),
  INDEX `fk_Elemento_Equipa1_idx` (`Equipa_idEquipa` ASC),
  INDEX `fk_Elemento_Seccao1_idx` (`Seccao_Designacao` ASC),
  INDEX `fk_Elemento_Funcao1_idx` (`Funcao_Designacao` ASC),
  CONSTRAINT `fk_Elemento_Equipa1`
    FOREIGN KEY (`Equipa_idEquipa`)
    REFERENCES `Agrup424`.`Equipa` (`idEquipa`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Elemento_Seccao1`
    FOREIGN KEY (`Seccao_Designacao`)
    REFERENCES `Agrup424`.`Seccao` (`Designacao`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Elemento_Funcao1`
    FOREIGN KEY (`Funcao_Designacao`)
    REFERENCES `Agrup424`.`Funcao` (`Designacao`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

Figura 7 -Código de Criação da Tabela
Elemento

Relação Atividade

Domínio idAtividade	Int
Domínio Designacao	String com tamanho 40
Domínio Descricao	String com tamanho 150
Domínio dia	Data no formato “YYYY-MM-DD”
Domínio Equipa_idEquipa	Int

Atividade(
idAtividade	Id da Atividade	NOT NULL,AUTO_INCREMENT
Designacao	Nome da Atividade	NOT NULL,
Descricao	Descrição da atividade	NOT NULL,
Dia	Data da Atividade	NOT NULL,
Equipa_idEquipa	Id da Equipa	NOT NULL,

PRIMARY KEY(idAtividade)),

FOREIGN KEY(Equipa_idEquipa) REFERENCES Equipa(idEquipa);

```

-----
-- Table `Agrup424`.`Atividade`
-----
CREATE TABLE IF NOT EXISTS `Agrup424`.`Atividade` (
  `idAtividade` INT NOT NULL AUTO_INCREMENT,
  `Designacao` VARCHAR(40) NOT NULL,
  `Descricao` VARCHAR(150) NOT NULL,
  `dia` DATE NOT NULL,
  `Equipa_idEquipa` INT NOT NULL,
  PRIMARY KEY (`idAtividade`),
  UNIQUE INDEX `idAtividade_UNIQUE` (`idAtividade` ASC),
  INDEX `fk_Atividade_Equipa1_idx` (`Equipa_idEquipa` ASC),
  CONSTRAINT `fk_Atividade_Equipa1`
    FOREIGN KEY (`Equipa_idEquipa`)
      REFERENCES `Agrup424`.`Equipa` (`idEquipa`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

Figura 8 -Código de Criação da Tabela Atividade

5.3. Tradução das Interrogações do Utilizador

Quais as funções existentes numa dada secção?

```

DELIMITER $$
CREATE procedure funcoes(IN seccaopretendida VARCHAR(45))

BEGIN

SELECT DISTINCT Funcao.Designacao AS 'Funções' FROM Elemento
INNER JOIN Funcao
  ON Funcao.Designacao = Elemento.Funcao_Designacao
INNER JOIN Seccao
  ON Elemento.Seccao_Designacao = seccao.Designacao
  WHERE seccao.Designacao = seccaopretendida;

END $$

SET @seccaopretendida = 'Lobitos';
CALL funcoes(@seccaopretendida);

DROP procedure funcoes;

```

Figura 9 - Código da Resolução da Interrogação

Qual o Patrono de todas as secções que foram representadas em determinada atividade por uma equipa?

```

DELIMITER $$
CREATE procedure cinco(IN atividade VARCHAR(40))
BEGIN
    SELECT DISTINCT Patrono FROM Seccao
    INNER JOIN Equipa
        ON Equipa.Seccao_Designacao = Seccao.Designacao
    INNER JOIN Atividade
        ON Atividade.Equipa_idEquipa = Equipa.idEquipa
    WHERE Atividade.designacao = atividade;

END; $$

SET @atividade = 'Convivio de Natal';
CALL cinco(@atividade);

DROP procedure cinco;

```

Figura 10 -Código da Resolução da Interrogação

Qual é o nome e respetivas funções dos elementos de uma determinada secção que fizeram determinada atividade?

```

DELIMITER $$
CREATE procedure quatro(IN seccaopretendida VARCHAR(45), IN atividade VARCHAR(40))
BEGIN
    SELECT nome, Funcao.Designacao FROM Elemento
    INNER JOIN Funcao
    ON Elemento.Funcao_Designacao = Funcao.Designacao
    INNER JOIN Seccao
    ON Elemento.Seccao_Designacao = Seccao.Designacao
    INNER JOIN Atividade
    ON Elemento.Equipa_idEquipa = Atividade.Equipa_idEquipa
    WHERE (Seccao.Designacao = seccaopretendida AND Atividade.Designacao = atividade);

END; $$

SET @seccao = 'Lobitos';
SET @atividade = 'Concurso de Jogo do Galo';
CALL quatro(@seccao, @atividade);

DROP procedure quatro;

```

Figura 11 - Código da Resolução da Interrogação

5.4. Tradução das Transações Estabelecidas

Inserir uma atividade na base de dados:

```
DELIMITER $$
CREATE procedure InsereAtividade (IN designacao VARCHAR(40), IN descricao VARCHAR(150), IN idEquipa INT, IN dias DATE)

BEGIN
DECLARE Erro BOOL DEFAULT 0;
DECLARE CONTINUE HANDLER FOR SQLEXCEPTION SET Erro=1;
START TRANSACTION;

INSERT INTO Atividade(Designacao,Descricao, dia, Equipa_idEquipa)
VALUES (designacao, descricao, dias, idEquipa);

IF erro
THEN ROLLBACK;
ELSE COMMIT;
END IF;
END $$

SET @designacao = 'Debates de Personalidades';
SET @descricao = 'Forma de promover o diálogo dentro da Equipa';
SET @dias = '2015-03-15';
SET @idEquipa = 1;
CALL InsereAtividade(@designacao,@descricao,@idEquipa,@dias);*/
INSERT INTO Elemento(Nin,Nome,Idade,DataDeNasc,Morada,Telemovel,Email,NoitesdeCampo,Equipa_idEquipa,Seccao_Designacao,Funcao_Designacao)
VALUES (40,'Ricardinho Costa',12,'2005-07-10','Rua dos Peões nº16',912523487,null,30,null,null,'Cozinheiro');*/
```

Figura 12 - Código da Inserção de uma nova Atividade

Inserir um elemento na base de dados:

```
DELIMITER $$
CREATE procedure InserirElemento(IN Nin_u INT,IN Nome_u VARCHAR(45),INOUT Idade_u INT,
IN DataDeNasc_u DATE,IN Morada_u VARCHAR(45),IN Telemovel_u INT,
IN Email_u VARCHAR(45),IN NoitesdeCampo_u INT,
IN Funcao_Designacao_u VARCHAR(45))

BEGIN
DECLARE erro bool DEFAULT 0;
DECLARE CONTINUE HANDLER FOR sqlexception SET erro =1;
SET @Seccao_Designacao = AtribuirSeccao(@Idade);
SET @Equipa_idEquipa = AtribuirEquipa();
INSERT INTO Elemento(Nin,Nome,Idade,DataDeNasc,Morada,Telemovel,Email,NoitesdeCampo,
Equipa_idEquipa,Seccao_Designacao,Funcao_Designacao)
VALUES (Nin_u,Nome_u,Idade_u,DataDeNasc_u,Morada_u,Telemovel_u,Email_u,NoitesdeCampo_u,
@Equipa_idEquipa,@Seccao_Designacao,Funcao_Designacao_u);
IF erro THEN rollback;
else commit;
end if;
END $$
DELIMITER ;

DELIMITER $$
CREATE function AtribuirSeccao (Idade int)
returns varchar(45)
BEGIN
CASE
WHEN Idade BETWEEN 6 AND 10 THEN RETURN 'Lobitos';
WHEN Idade BETWEEN 11 AND 14 THEN RETURN 'Exploradores';
WHEN Idade BETWEEN 15 AND 17 THEN RETURN 'Pioneiros';
WHEN Idade >= 18 THEN RETURN 'Caminheiros';
END CASE;
END $$
DELIMITER ;

DELIMITER $$
CREATE function AtribuirEquipa()
returns int
BEGIN
return (select distinct FLOOR(RAND()*(count(idEquipa)-1)+1) from Equipa
where Seccao_Designacao = @Seccao_Designacao);
END $$
DELIMITER ;

SET @Idade =12;
SET @Nin = 74;
SET @Seccao_Designacao = null;
SET @Equipa_idEquipa = null;

call InserirElemento(@Nin,'Teste Teste4',@Idade,'1995-6-6',
'rua das ruas',912345987,null,100,'Cozinheiro');

drop procedure InserirElemento;
drop function AtribuirEquipa;
drop function AtribuirSeccao;
```

Figura 13 - Código da Inserção de um novo Elemento

Arquivar os dados de determinado elemento:

```
DELIMITER $$
CREATE procedure ArquivarElemento(INOUT Nin_u INT)
BEGIN
DECLARE erro bool DEFAULT 0;
DECLARE CONTINUE HANDLER FOR sqlexception SET erro =1;

CREATE TABLE if not exists ElementoArquivado LIKE Elemento;
DELETE from Elemento where Elemento.Nin = Nin_u;

IF erro THEN rollback;
else commit;
end if;
END $$
DELIMITER ;

DELIMITER $$
CREATE TRIGGER tr_ArquivarElem BEFORE DELETE
ON Elemento
FOR EACH ROW
BEGIN

INSERT INTO ElementoArquivado
SELECT * FROM Elemento
WHERE Elemento.Nin=@Nin;

END $$
DELIMITER ;

SET @Nin = 36;
call ArquivarElemento(@Nin);

select * from Elemento;
select * from ElementoArquivado;
drop procedure ArquivarElemento;
```

Figura 14 - Código da Arquivação de um dado Elemento

5.5. Tradução das Transações Estabelecidas

Tabela 4 - Espaço Ocupado pela Entidade Elemento

<i>Tabela</i>	<i>Atributo</i>	<i>Tipo de dados</i>	<i>Espaço Ocupado</i>
---------------	-----------------	----------------------	-----------------------

Elemento	<u>NIN</u>	INT	4 bytes
	Nome	VARCHAR(45)	46 bytes
	Idade	INT	4 bytes
	Sexo	ENUM	3 bytes
	DatadeNasc	DATE	3 bytes
	Morada	VARCHAR(45)	46 bytes
	Telemovei	INT	4 bytes
	Email	VARCHAR(45)	46 bytes
	NoitesdeCampo	INT	4 bytes
	Equipa_idEquipa	INT	4 bytes
	Seccao_Designacao	VARCHAR(45)	46 bytes
	Funcao_Designacao	VARCHAR(100)	101 bytes
Total			Nº de Elementos * tamanho dos dados para um elemento = 39*311 = 12 129 bytes

Tabela 5 - Espaço Ocupado pela Entidade Seccao

<i>Tabela</i>	<i>Atributo</i>	<i>Tipo de dados</i>	<i>Espaço Ocupado</i>
---------------	-----------------	----------------------	-----------------------

Seccao	<u>Designacao</u>	VARCHAR (45)	46 bytes
	Simbologia	VARCHAR(100)	101 bytes
	Patrono	VARCHAR(45)	46 bytes
Total			Nº de Secções * tamanho dos dados para uma secção = $4 \times 193 = 772$ bytes

Tabela 6 -Espaço Ocupado pela Entidade Equipa

<i>Tabela</i>	<i>Atributo</i>	<i>Tipo de dados</i>	<i>Espaço Ocupado</i>
Equipa	<u>idEquipa</u>	INT	4 bytes
	Nome	VARCHAR(45)	46 bytes
	Lema	VARCHAR(45)	46 bytes
	Grito	VARCHAR(45)	46 bytes
	Seccao_Designacao	VARCHAR(45)	46 bytes
Total			Nº de Equipas * tamanho dos dados para uma equipa = $5 \times 188 = 940$ bytes

Tabela 7 - Espaço Ocupado pela Entidade Atividade

<i>Tabela</i>	<i>Atributo</i>	<i>Tipo de dados</i>	<i>Espaço Ocupado</i>
Atividade	<u>idAtividade</u>	INT	4 bytes
	Designacao	VARCHAR(40)	41 bytes
	Descricao	VARCHAR(150)	151 bytes
	dia	DATE	3 bytes
	Equipa_idEquipa	INT	4 bytes
Total			Nº de Atividades * tamanho dos dados para uma atividade = 10*203 = 2030 bytes

Tabela 8 - Espaço Ocupado pela Entidade Funcao

<i>Tabela</i>	<i>Atributo</i>	<i>Tipo de dados</i>	<i>Espaço Ocupado</i>
---------------	-----------------	----------------------	-----------------------

Funcao	<u>Designacao</u>	VARCHAR(100)	101 bytes
	Descricao	VARCHAR(350)	351 bytes
Total			Nº de Funções * tamanho dos dados para uma Função = 15*452 = 6780 bytes

Tabela 9 - Estimativa do Tamanho da Base de Dados

<i>Tabela</i>	<i>Tamanho</i>
Elemento	12129 bytes
Equipa	940 bytes
Seccao	772 bytes
Atividade	2030 bytes
Funcao	6780 bytes

Tamanho com povoamento	22.12012 Kbytes
-------------------------------	------------------------

Estimativa do tamanho da taxa de crescimento anual:

De modo a criar uma estimativa do crescimento do Agrupamento, voltamos a ter uma reunião com o cliente. Este fez a previsão que o número de elementos irá crescer 10% da população inicial, por mês. Por ano haverá um aumento de 3 equipas e durante um ano será acrescentado à base de dados 15 atividades.

Elementos: Se crescer a uma taxa de 10%, 3 elementos irão ser adicionados à base de dados por mês o que perfaz um total de 36 elementos no final de um ano.

<i>Nº de elementos ao fim de um ano</i>	<i>Tamanho final da tabela Elemento</i>
$39 + 36 = 75$	$75 * 311$ (tamanho dos dados de um elemento) = 23325 bytes

Equipa:

<i>Nº de Equipas ao fim de um ano</i>	<i>Tamanho final da tabela Equipa</i>
$5 + 3 = 8$	$8 * 188$ (tamanho dos dados de uma equipa) = 1504 bytes

Atividade:

<i>Nº de Atividades ao fim de um ano</i>	<i>Tamanho final da tabela Atividade</i>
$10 + 15 = 25$	$25 * 203$ (tamanho dos dados de uma atividade) = 5075 bytes

Tabela 10 - Estimativa do Tamanho da Base de Dados após um ano

<i>Tabela</i>	<i>Tamanho</i>
Elemento	23325 bytes
Equipa	1504 bytes
Seccao	772 bytes
Atividade	5075 bytes
Funcao	6780 bytes
Tamanho do Agrupamento ao fim de um ano	36.57813 Kbytes

5.6. Definição e Caracterização das Vistas de Utilização

Vista de todos os Chefes:

```
-- View para Chefes

CREATE VIEW membroschefes AS

SELECT * FROM Elemento
INNER JOIN Funcao
  ON Funcao.Designacao = Elemento.Funcao_Designacao
 WHERE (Funcao.Designacao = 'Chefe de Agrupamento' || Funcao.Designacao = 'Chefe Tesoureiro' ||
        Funcao.Designacao = 'Chefe Secretário' || Funcao.Designacao = 'Chefe dos Lobitos' ||
        Funcao.Designacao = 'Chefe dos Exploradores' || Funcao.Designacao = 'Chefe dos Pioneiros' || Funcao.Designacao = 'Chefe dos Caminheiros');

-- drop view membroschefes;

SELECT Nome, Idade, Funcao_Designacao AS 'Funções' FROM membroschefes
```

Figura 15 - Vista Chefes

5.7. Definição e Caracterização dos Mecanismos de Segurança

Identificou-se três diferentes utilizadores na base de dados.

- O “admin” tem acesso total à base de dados:
- O “chefe” tem acesso aos elementos, atividade e equipa podendo alterar, remover ou adicionar informação:
- O “elem” tem acesso a tudo mas só para visualização:

```
-- User admin tem autorizaçao para todos os privilegios ate mesmo
-- para atribuir privilegios podendo-se logar de qualquer lado.
GRANT ALL
ON `agrup424`.* TO admin IDENTIFIED BY 'admin'
WITH GRANT OPTION;

-- User adchefe tem autorizaçao para alterar dados nas tabelas elemento, atividade e equipa
GRANT SELECT,INSERT,UPDATE,DELETE ON `agrup424`.`Elemento` TO adchefe@localhost IDENTIFIED BY 'chefe';
GRANT SELECT,INSERT,UPDATE,DELETE ON `agrup424`.`Atividade` TO adchefe@localhost;
GRANT SELECT,INSERT,UPDATE,DELETE ON `agrup424`.`Equipa` TO adchefe@localhost;

-- User adelemento .
GRANT USAGE
ON `agrup424`.`Elemento` TO adelemento@localhost IDENTIFIED BY 'elem';

SELECT user,host from mysql.user;
DROP USER chefeSecretario@localhost;
SHOW GRANTS FOR adelemento@localhost;
```

Figura 16 - Código de Criação e Respetivos Previlégios dos três Utilizadores

5.8. Revisão do Sistema Implementado

Efetuamos uma reunião com o cliente em que testámos as funcionalidades da implementação física com as interrogações, transações e por fim ainda apresentamos uma estimativa de espaço de memória que a base de dados ocupará.

Por fim, obtivemos uma aprovação positiva do cliente.

6. Conclusão

Sendo este o primeiro projeto que tínhamos em mãos, um Sistema de Gestão de Base de Dados para o Agrupamento de Escuteiros 424-Nogueira, concluímos que foi uma boa iniciação à linguagem de SQL. A base de dados que nos propusemos a resolver era relativamente simples e com um nível de complexidade suficiente para os objetivos que temos, ter aprovação à UC com uma boa nota.

Deparamo-nos com várias dificuldades, tais como a normalização e estamos satisfeitos com o resultado final apesar de o erro que temos com um atributo específico em que este viola a primeira forma normal. Contudo, seria necessária uma constante manutenção da base de dados para que esta se mantivesse sempre bem otimizada.

Referências

Thomas M. Connolly, Carolyn E. Begg - Database Systems: A Practical Approach to Design, Implementation and Management - 4th Edition.

Lista de Siglas e Acrónimos

BD Base de Dados

SGBD Sistema de Gestão de Bases de Dados