



Universidade do Minho
Licenciatura em Ciências da Computação

Unidade Curricular de Bases de Dados

Ano Lectivo de 2024/2025

Sport Org

Alexandra Calafate (A100060)

Hélder Simão Pereira (A102521)

Guilherme Duarte (A101122)

Killian Oliveira (A105532)

Janeiro, 2025

BD

Data de Recepção	
Responsável	
Avaliação	
Observações	

Sport Org

Alexandra Calafate (A100060)

Hélder Simão Pereira (A102521)

Guilherme Duarte (A101122)

Killian Oliveira (A105532)

Janeiro, 2025

Resumo

Este relatório foi realizado no âmbito da unidade curricular de Bases de Dados. O objetivo deste trabalho é desenvolver um sistema de base de dados para um dado caso de estudo no domínio da “Gestão de Atletas, Equipas e Resultados de uma Competição Desportiva Multimodalidade”, com particular ênfase na análise, planeamento, modelação, arquitetura, e implementação deste tipo de sistemas.

Pretendemos criar uma base de dados para os organizadores de um evento desportivo, de forma organizada e de fácil compreensão. Os atletas podem pesquisar outros atletas, os eventos e o seu horário.

Índice

Conteúdo

Resumo	i
Índice	ii
Índice de Figuras	iv
Índice de Tabelas	vi
1. Introdução	1
1.1. Contextualização	1
1.2. Fundamentação	1
1.3. Objetivos	2
1.4. Viabilidade	2
1.5. Recursos a utilizar	2
1.6. Equipa de trabalho	2
1.7. Plano de execução do trabalho	3
2. Definição de Requisitos	4
2.1. Método de levantamento e de análise de requisitos adotado	4
2.2. Organização dos requisitos levantados	4
2.3. Análise e validação geral dos requisitos	7
3. Modelação Concetual	8
3.1. Apresentação da abordagem de modelação realizada	8
3.2. Identificação e caracterização das entidades	8
3.3. Identificação e caracterização dos relacionamentos	9
3.4. Identificação e caracterização da associação dos atributos com as entidades e relacionamentos	14
3.5. Apresentação e explicação do diagrama ER produzido	17
4. Modelação Lógica	18
4.1. Construção e validação do modelo de dados Lógico	18
4.2. Normalização de Dados	20
4.3. Apresentação e explicação do modelo Lógico produzido	20
4.4. Validação do modelo com interrogações do utilizador	21
5. Implementação Física	24
5.1. Apresentação do processo de criação da base de dados	24
5.2. Cálculo do espaço da base de dados(inicial e taxa de crescimento anual)	25

5.3. Realização do povoamento da base de dados	29
5.4. Tradução das interrogações do utilizador para SQL (alguns exemplos)	31
5.5. Definição e caracterização das vistas de utilização em SQL (alguns exemplos)	32
5.6. Definição dos perfis de utilização para cada utilizador da base de dados	34
5.7. Indexação do Sistema de Dados	35
5.8. Procedimentos Implementados	35
5.9. Plano de segurança e recuperação de dados	37
6. Conclusões e Trabalho Futuro	39
7. Referências Bibliográficas	40

Índice de Figuras

Figura 1 – Diagrama de Gantt	3
Figura 2 – Relacionamento Atleta-Modalidade	9
Figura 3 – Relacionamento Atleta-Universidade	10
Figura 4 - Relacionamento Modalidade-Treinador	11
Figura 5 – Relacionamento Atleta-Evento	11
Figura 6 – Relacionamento Evento-Staff	12
Figura 7 – Relacionamento Atleta-Morada	13
Figura 8 – Relacionamento Universidade-Morada	14
Figura 9 – Modelo Conceptual	17
Figura 10 – Modelo Lógico	21
Figura 11 – Árvore correspondente à primeira interrogação	22
Figura 12 – Árvore correspondente à segunda interrogação	22
Figura 13 – Script de criação da tabela Universidade	24
Figura 14 – Script de criação da tabela Atleta	24
Figura 15 – Script de criação da tabela Atleta-Evento	25
Figura 16 – Script de criação da tabela Modalidade-Treinador	25
Figura 17 – Povoamento de Universidade	29
Figura 18 – Povoamento de Evento	30
Figura 19 – Povoamento de Staff	30
Figura 20 – Primeira interrogação: Número de atletas Masculinos	31
Figura 21 – resultado da primeira interrogação	31
Figura 22 – Segunda interrogação: Data de um Evento específico	31
Figura 23 – resultado da segunda interrogação	31
Figura 24 – Terceira interrogação: Quais são os atletas de uma Universidade	31
Figura 25 – resultado da terceira interrogação	32
Figura 26 – Quarta interrogação: Nacionalidade com mais Atletas	32
Figura 27 – resultado da quarta interrogação	32
Figura 28 – Primeira View	32
Figura 29 – resultado da Primeira View	33
Figura 30 – Segunda View	33
Figura 31 – resultado da Segunda View	34
Figura 32 – Definição dos Perfis1	34

Figura 33 – Definição dos Perfis2	35
Figura 34 – Indexes	35
Figura 35 – Primeira Procedure	35
Figura 36 – Primeira Procedure2	36
Figura 37 – Segunda Procedure	36
Figura 38 – Terceira Procedure	37
Figura 39 – <i>Backup</i>	38

Índice de Tabelas

Tabela 1 – Requisitos de Descrição	5
Tabela 2 – Requisitos de Exploração	6
Tabela 3 – Requisitos de Controlo	7
Tabela 4 - Caracterização dos atributos de Atleta	14
Tabela 5 - Caracterização dos atributos de Universidade	15
Tabela 6 - Caracterização dos atributos de Evento	15
Tabela 7 - Caracterização dos atributos de Modalidade	15
Tabela 8 - Caracterização dos atributos de Treinador	16
Tabela 9 - Caracterização dos atributos de Staff	16
Tabela 10 – Calculo dos bytes associados à entidade Universidade	26
Tabela 11 – Calculo dos bytes associados à entidade Atleta	26
Tabela 12 – Calculo dos bytes associados à entidade Morada	26
Tabela 13 – Calculo dos bytes associados à entidade Evento	27
Tabela 14 – Calculo dos bytes associados à entidade Staff	27
Tabela 15 – Calculo dos bytes associados à entidade Modalidade	27
Tabela 16 – Calculo dos bytes associados à entidade Treinador	28
Tabela 17 – Calculo dos bytes associados à entidade Atleta_Evento	28
Tabela 18 – Calculo dos bytes associados à entidade Evento_Staff	28
Tabela 19 – Calculo dos bytes associados à entidade Atleta_Modalidade	28
Tabela 20 – Calculo dos bytes associados à entidade Modalidade_Treinador	29

1. Introdução

1.1. Contextualização

Joana Torres e Tomás Fonseca moram em Braga, e ambos partilham um gosto pelo desporto. Desde que eles eram crianças eles vêm todos os Jogos Olímpicos e imaginavam um dia poder organizar um evento desportivo como o que eles adoravam.

Porém houve um evento na universidade que eles frequentam e depararam-se com vários problemas. Eles viram a necessidade de criar um evento desportivo com uma plataforma online de maneira que a informação esteja toda organizada e que seja de fácil compreensão, e com que os atletas e equipas que participem e consigam ter acesso a certos dados.

Com isto, eles decidiram fundar uma plataforma e chamaram-lhe de "*Sport Org*", com a iniciativa de criar um evento entre Universidades por todo o mundo que incluísse diferentes modalidades.

1.2. Fundamentação

Para que exista uma melhor organização da plataforma, a equipa do projeto "*Sport Org*" optou por criar um Sistema de Base de Dados (SBD) que lhes permitisse armazenar de forma eficiente todas as informações necessárias.

Uma plataforma de desporto é responsável por tratar de uma grande quantidade de informações complexas, incluindo detalhes sobre atletas, modalidades, universidades e o horário de cada evento. Utilizar uma base de dados é a melhor solução encontrada por eles para organizar e armazenar essas informações de maneira estruturada e eficiente.

O SBD exerce uma função essencial no projeto "*Sport Org*", sendo indispensável para armazenar, gerência e ter o acesso a dados complexos de maneira segura e eficiente. A base de dados proporciona uma plataforma eficaz tanto para os organizadores quanto para os seus utilizadores, permitindo a divulgação ágil e eficaz deste evento de desportos da sua universidade.

1.3. Objetivos

Após a Joana e o Tomás terem-se reunido, juntos definiram várias metas tendo como objetivo uma melhoria do sistema que eles pretendem implementar:

- Criar uma plataforma online eficiente que facilite a gestão e organização.
- Melhorar a experiência dos gestores e de quem visita a base de dados.
- Tornar-se um evento desportivo que tenha muita adesão.
- Proteger os dados pessoais dos atletas, cumprindo os regulamentos de segurança e privacidade de dados em vigor.
- Assegurar que os atletas e os organizadores tenham acesso a um suporte técnico ágil e eficiente para resolver problemas e esclarecer dúvidas rapidamente.

1.4. Viabilidade

Joana e Tomás estão confiantes que o seu evento será muito importante para o futuro do desporto entre universidades. Eles acreditam que o projeto “*Sport Org*”:

- Vai ser benéfico em termos de eficiência e trabalho manual, poupando tempo e possivelmente com menos despesas.
- Irá contribuir para a promoção de futuros eventos desportivos também interuniversitários.
- Vai promover visibilidade e notoriedade para o país, atraindo mais participantes e público.
- Terá uma boa segurança e privacidade, deixando os seus utilizadores confortáveis e confiantes.

1.5. Recursos a utilizar

Recursos humanos: 2 gestores, 1 equipa de desenvolvimento, 1 especialista em *Marketing*.

Recursos materiais: *Hardware* (1 servidor, 1 máquina). *Software* (MySQL, brModelo)

1.6. Equipa de trabalho

Pessoal interno:

- Joana e Tomás: fundadores e gestores do projeto “*Sport Org*”, estes lideram o projeto e tomam as decisões finais do mesmo. Supervisionam a equipa.
- Equipa de desenvolvimento: equipa de programadores responsáveis do desenvolvimento e manutenção da plataforma.

1.7. Plano de execução do trabalho

Joana e Tomás reuniram com a sua equipa de desenvolvimento e juntos decidiram criar um plano sob a forma de diagrama de GANTT para organizarem se melhor a cerca do projeto e visualizar mais facilmente o plano de execução.

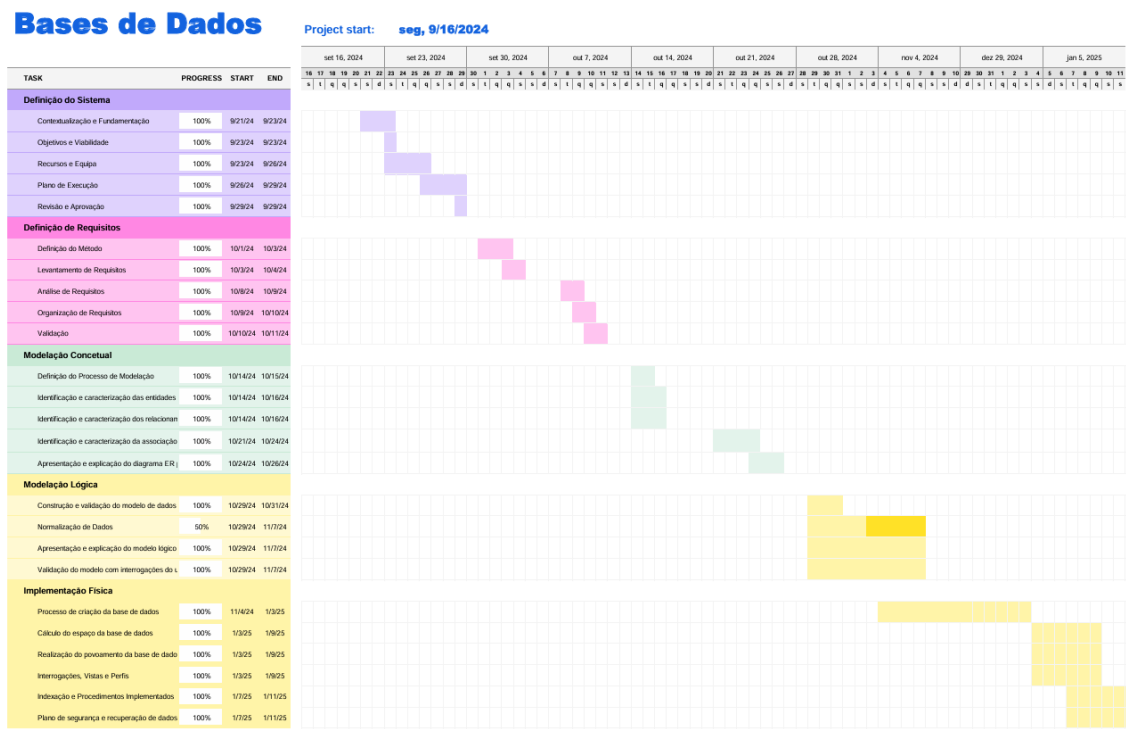


Figura 1 – Diagrama de Gantt

2. Definição de Requisitos

2.1. Método de levantamento e de análise de requisitos adotado

De maneira a aprimorar o funcionamento da sua plataforma, eles usaram diversas maneiras de fazer o levantamento de dados que permitirá que estabeleçam conjuntos de requisitos abrangentes que desejam implementar na sua base de dados.

Além disso, é uma forma que eles arranjam para que corrigem erros já existentes na plataforma.

Eles usaram os seguintes métodos:

- **Questionários:** feitos aos atletas, e vários treinadores. Deste modo foi possível fazer o juntamento dos seus dados.
- **Entrevistas:** Realizadas inicialmente pela Joana e pelo Tomas. Obtiveram assim *feedback* de funcionalidades que a plataforma deveria priorizar, e lacunas que esta poderia vir a ter.
- **Observação Direta:** Joana e Tomás participaram como observadores em outros eventos desportivos universitários antes do começo do projeto para entender o seu funcionamento e os problemas enfrentados pelos atletas.
- **Documentação:** Joana e Tomás decidiram que através deste método iriam encontrar informação mais rapidamente.

2.2. Organização dos requisitos levantados

Após verificarem os requisitos que levantaram, estes foram analisados um a um, tendo em conta o aprimoramento do sistema. Organizando-os por requisitos de: Descrição, Exploração e Controlo.

2.2.1. Requisitos de Descrição

Tabela 1 – Requisitos de Descrição

Numero		Descrição	Fonte
		Entidades	
1	RD01	Um Atleta registado no sistema é constituído por IdAtleta (INT, PRIMARY KEY), NomeAtleta (VARCHAR(75)), Genero (VARCHAR(20)), Data_Nascimento (DATE), Nacionalidade (VARCHAR(50)), Email (VARCHAR(75)), Contacto(INT), IdUniversidade(INT, FOREIGN KEY), IdMorada(INT, FOREIGN KEY)	Entrevista
2	RD02	Uma Universidade registada no sistema contém um IdUniversidade(INT, PRIMARY KEY), Contacto(INT), Pais(VARCHAR(50)), NomeUniversidade (VARCHAR(45)), IdMorada(INT, FOREIGN KEY)	Documentação
3	RD03	Um Evento registado no sistema contém um IdEvento(INT, PRIMARY KEY), Local(VARCHAR(60)), NomeEvento(VARCHAR(45)), Data_Hora(DATETIME)	Questionário
4	RD04	Uma Morada registada no sistema contém um IdMorada(INT, PRIMARY KEY), Rua(VARCHAR(45)), NrPorta(INT), CodPostal(VARCHAR(45))	Entrevista
5	RD05	Um Staff registado no sistema contém um IdStaff(INT, PRIMARY KEY), NomeStaff(VARCHAR(45)), Horario(VARCHAR(60)), Tipo(VARCHAR(20)), Contacto(INT), Email(VARCHAR(75))	Entrevista
6	RD06	Uma Modalidade registada no sistema contém um IdModalidade(INT, PRIMARY KEY), NomeModalidade(VARCHAR(45)), Descricao(VARCHAR(100)), Genero(VARCHAR(20)), Regras(VARCHAR(100))	Questionário
7	RD07	Um Treinador registado no sistema contém um IdTreinador(INT, PRIMARY KEY), NomeTreinador(VARCHAR(45)), Contacto(INT), Email(VARCHAR(75))	Entrevista
		Relacionamentos	
8	RD08	Cada Atleta tem que frequentar uma Universidade, e uma Universidade tem de ter pelo menos um Atleta	Entrevista

		mas pode ter vários.	
9	RD09	Um Atleta tem que praticar pelo menos uma modalidade mas pode praticar mais, uma Modalidade tem de ser praticada por pelo menos um atleta.	Observação Direta
10	RD10	Um Atleta tem que participar pelo menos em um Evento mas pode participar em mais, um Evento tem de ter pelo menos um Atleta.	Observação Direta
11	RD11	Um Evento tem de ter pelo menos um Staff, e um Staff tem que pelo menos participar em um Evento.	Entrevista
12	RD12	Um Atleta tem que ter pelo menos uma Morada, e uma Morada tem que ter pelo menos um Atleta.	Entrevista
13	RD13	Uma Universidade tem que ter pelo menos uma Morada, e uma Morada tem que ter pelo menos uma Universidade.	Entrevista

2.2.2. Requisitos de Exploração

Tabela 2 – Requisitos de Exploração

Numero		Descrição	Fonte
		Exploração	
1	RD01	O sistema deve ser capaz de mostrar o número de atletas Masculinos	Entrevista
2	RD02	O sistema deve ser capaz de mostrar o número de atletas Femininos	Entrevista
3	RD03	O sistema deve ser capaz de listar uma data específica de um certo evento. (Exemplo: Volleyball Feminina)	Entrevista
4	RD04	O sistema deve ser capaz de dado o nome de uma universidade saber quais são os seus atletas (Exemplo: Universidade de Cambridge)	Entrevista
5	RD05	O sistema deve ser capaz de dado o nome de uma modalidade saber quais são os treinadores dessa mesma (exemplo: Atletismo 200m)	Entrevista
6	RD06	O sistema deve ser capaz de saber a nacionalidade com mais atletas	Entrevista
7	RD07	O sistema deve ser capaz de saber a nacionalidade com menos atletas	Entrevista

2.2.3. Requisitos de Controlo

Tabela 3 – Requisitos de Controlo

Numero		Descrição	Fonte
		Controlo	
1	RD01	Apenas a Joana, o Tomás e a Equipa de Desenvolvimento podem realizar operações de modificação ou de remoção de dados relativos ao sistema.	Entrevista
2	RD02	O Sistema deve atribuir a cada eventos uma localização conforme a data e hora.	Entrevista
3	RD03	Os Atletas apenas têm acesso às informações dos atletas e evento.	Entrevista

2.3. Análise e validação geral dos requisitos

Após terem feito o levantamento dos requisitos a equipa toda atrás do sistema decidiu reunir-se para realizar uma análise geral dos requisitos. Após reunirem-se decidiram seguir com a criação da base de dados. Como estava tudo de acordo com as suas ideias continuamos com o trabalho.

3. Modelação Concetual

3.1. Apresentação da abordagem de modelação realizada

Após terem sido recolhido todos os requisitos necessários para a plataforma, começou o planeamento da estrutura da Base de Dados.

Para facilitar a sua construção iniciaram com um Diagrama ER, em que demonstra como as identidades se relacionam entre si. Seguindo pela ordem de entidades, relacionamentos entre si e os seus atributos.

Assim a sua modelação conceptual deve representar uma estrutura de acordo com a ordem referida anteriormente.

3.2. Identificação e caracterização das entidades

Avaliando o funcionamento da plataforma online, consideramos que as entidades seriam: Atleta, Modalidade, Treinador, Evento, Universidade, Staff e Morada. Para cumprir os requisitos que foram apresentados anteriormente, as entidades possuem os seguintes atributos.

- **Atleta:** é fundamental pois representa um dos utilizadores da nossa plataforma online. São eles que participam no evento desportivos e assim estes podem ser realizados.
- **Universidade:** é fundamental pois assim podemos distinguir os atletas pela universidade que frequentam, como se fosse a “equipa” deles.
- **Evento:** é fundamental.
- **Staff:** são as pessoas que estão a trabalhar para o evento.
- **Modalidade:** é fundamental para sabermos o tipo de desporto que os atletas praticam e a modalidade do evento.
- **Treinador :** é fundamental para saber o treinador representante da modalidade de uma certa universidade.
- **Morada:** é fundamental para saber a morada do atleta e da Universidade.

3.3. Identificação e caracterização dos relacionamentos

Na nossa modelação são utilizados diferentes tipos de relacionamentos entre as entidades. Assim segue uma análise e breve explicação individual feita através desses relacionamentos.

- **Relacionamento Atleta-Modalidade**

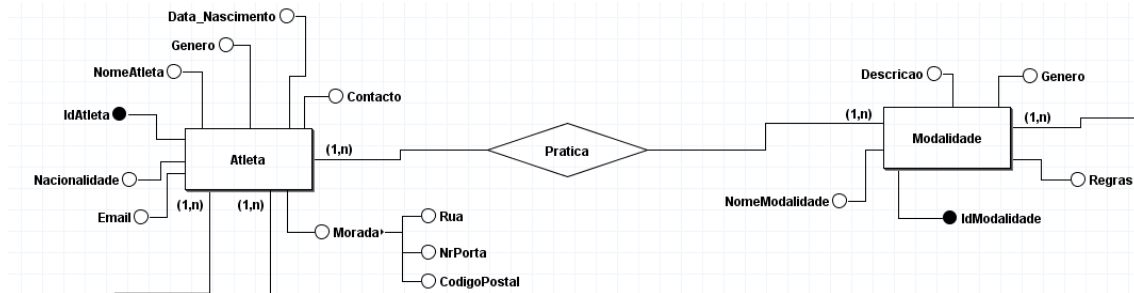


Figura 2 – Relacionamento Atleta-Modalidade

Relacionamento: Atleta pratica Modalidade

Descrição: Este evento serve para saber a modalidade que cada atleta participa.

Multiplicidade: Atleta(1,n) – (1,n)Modalidade

Um atleta pode praticar mais do que uma modalidade, e uma modalidade tem de ter mais do que um atleta.

Atributos: O relacionamento não tem atributos.

- **Relacionamento Atleta-Universidade**

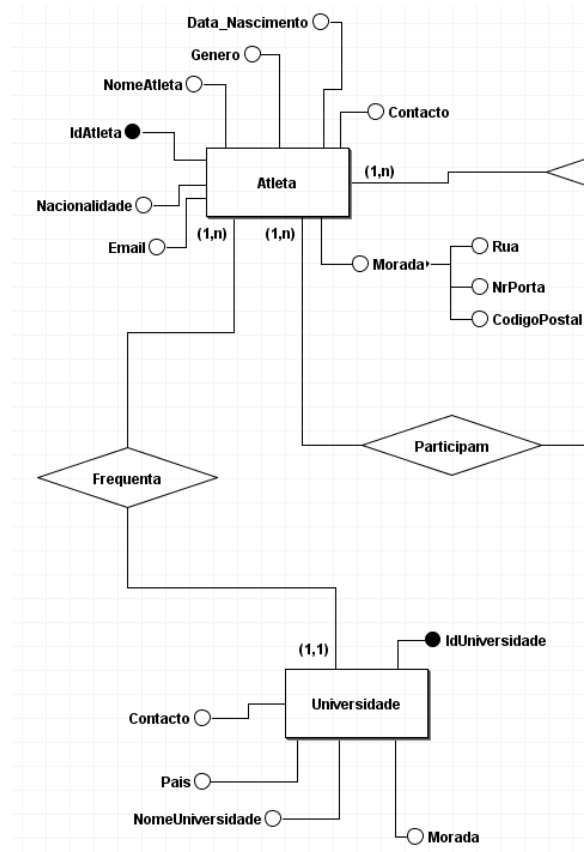


Figura 3 – Relacionamento Atleta-Universidade

Relacionamento: Atleta frequenta Universidade

Descrição: Serve para sabermos em que universidade os atletas frequentam.

Multiplicidade: Atleta(1,n) – (1,1)Universidade

Uma Universidade pode ter mais que um atleta, mas um atleta só pode frequentar uma universidade.

Atributos: O relacionamento não tem atributos.

- **Relacionamento Modalidade-Treinador**

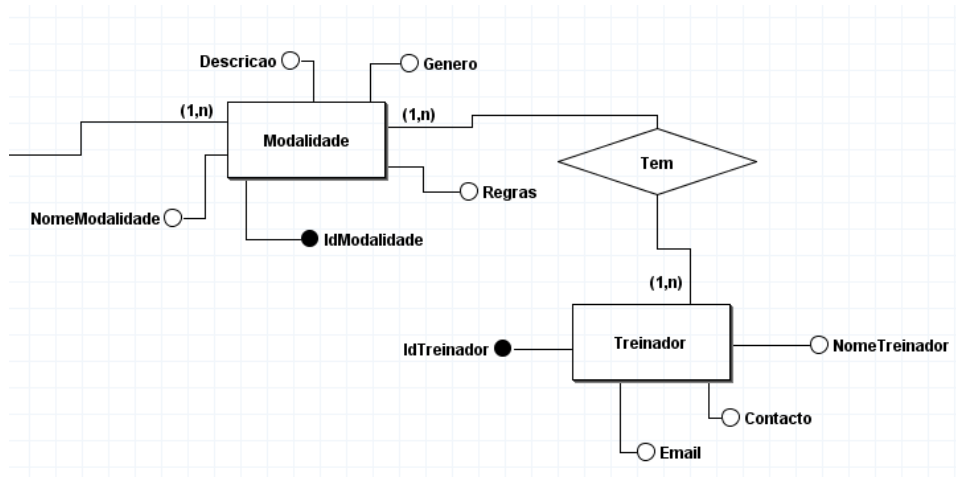


Figura 4 - Relacionamento Modalidade-Treinador

Relacionamento: Modalidade tem Treinador

Descrição: Serve para sabermos quais são os treinadores da modalidade.

Multiplicidade: Modalidade(1,n) – (1,n)Treinador

Uma Modalidade pode ter mais que um Treinador, e um Treinador pode frequentar mais que uma modalidade.

Atributos: O relacionamento não tem atributos.

- **Relacionamento Atleta-Evento**

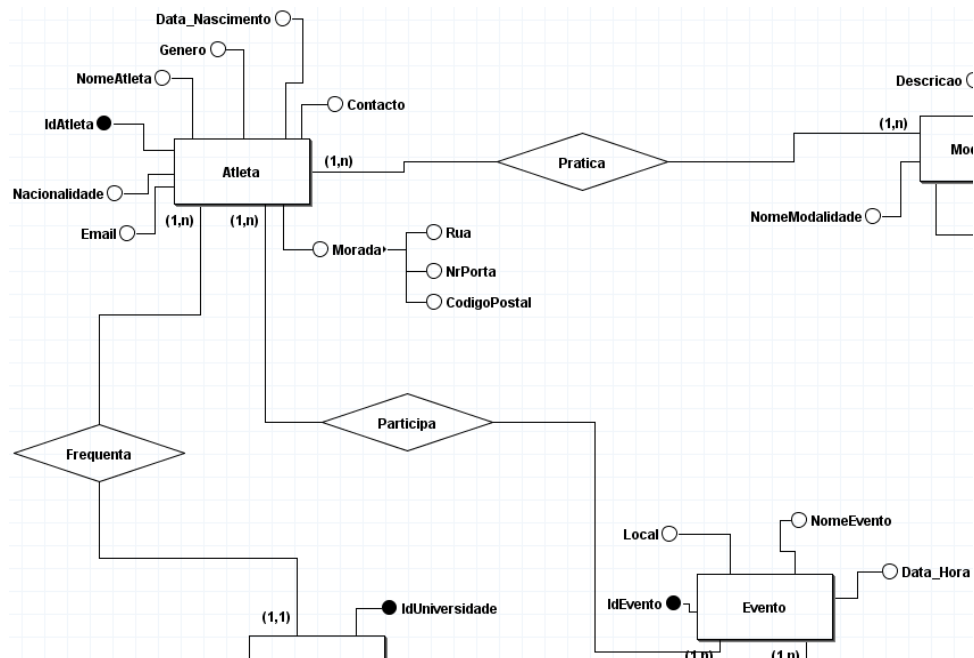


Figura 5 – Relacionamento Atleta-Evento

Relacionamento: Atleta participa Evento

Descrição: Serve para sabermos quais atletas participam naquele Evento.

Multiplicidade: Atleta(1,n) – (1,n)Evento

Um Atleta pode participar em vários eventos, num evento pode ter mais que um atleta.

Atributos: O relacionamento não tem atributos.

- **Relacionamento Evento-Staff**

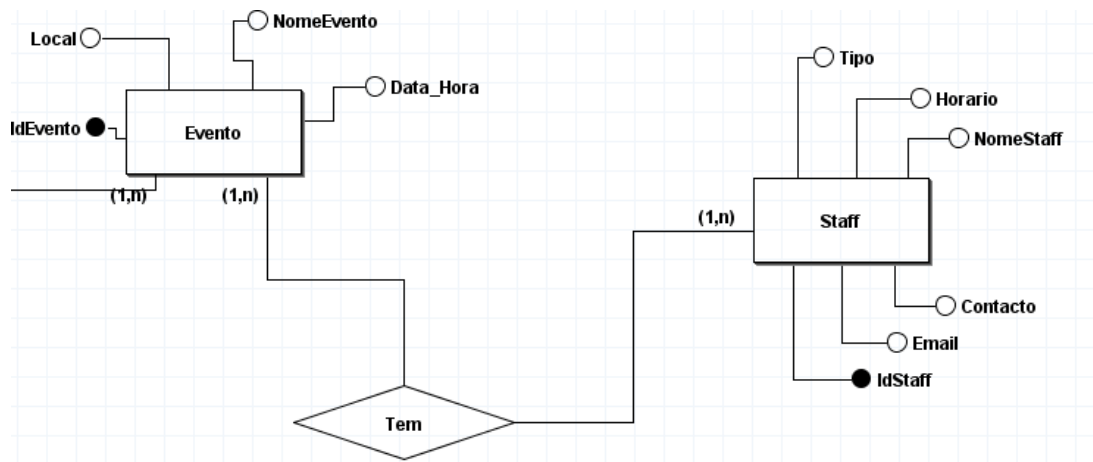


Figura 6 – Relacionamento Evento-Staff

Relacionamento: Evento tem Staff

Descrição: Serve para sabermos quais staffs trabalham no evento.

Multiplicidade: Atleta(1,n) – (1,n)Evento

Um Atleta pode participar em vários eventos, num evento pode ter mais que um atleta.

Atributos: O relacionamento não tem atributos.

- **Relacionamento Atleta-Morada**

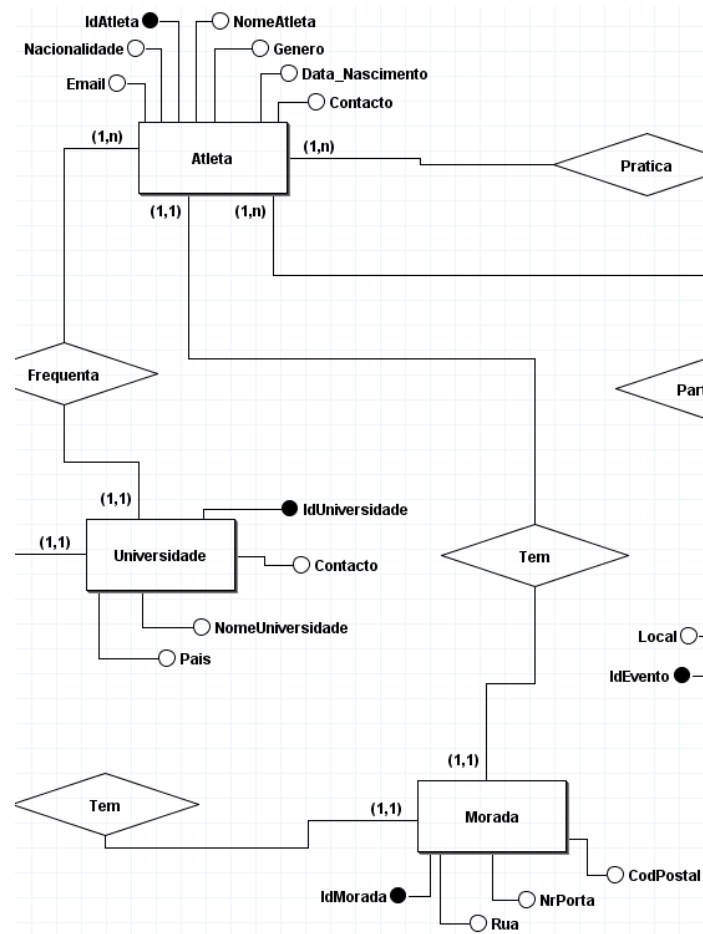


Figura 7 – Relacionamento Atleta-Morada

Relacionamento: Atleta tem Morada

Descrição: Serve para sabermos qual é a Morada do Atleta.

Multiplicidade: Atleta(1,1) – (1,1)Morada

Um Atleta só pode ter uma Morada, uma Morada só pode ter um Atleta.

Atributos: O relacionamento não tem atributos.

- **Relacionamento Universidade-Morada**

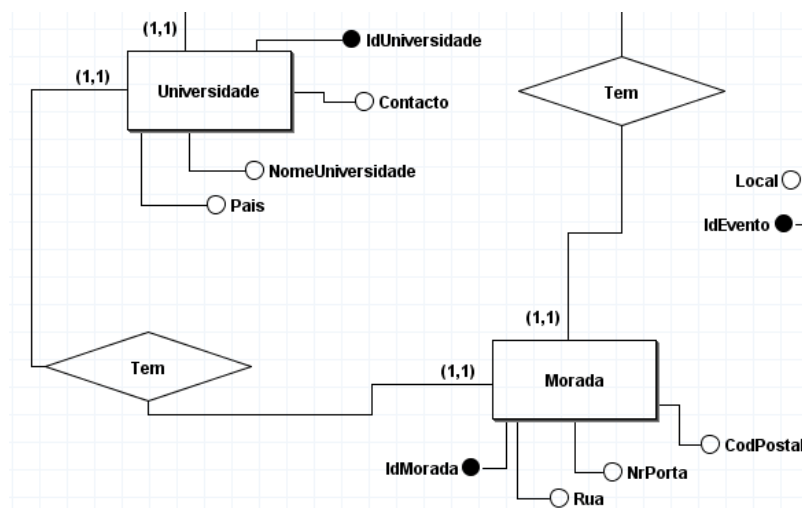


Figura 8 – Relacionamento Universidade-Morada

Relacionamento: Universidade tem Morada

Descrição: Serve para sabermos qual é a Morada da Universidade.

Multiplicidade: Universidade(1,1) – (1,1)Morada

Uma Universidade só pode ter uma Morada, uma Morada só pode ter uma Universidade.

Atributos: O relacionamento não tem atributos.

3.4. Identificação e caracterização da associação dos atributos com as entidades e relacionamentos

- **Entidade Atleta**

Tabela 4 - Caracterização dos atributos de Atleta

Atributos	Tipos de Dados	NULO	Composto	Multivalorado	Chave Primária
Id Atleta	INT	Não	Não	Não	Sim
Nome Atleta	VARCHAR(75)	Não	Não	Não	Não
Nacionalidade	VARCHAR(50)	Não	Não	Não	Não
Email	VARCHAR(75)	Não	Não	Não	Não
Contacto	INT	Não	Não	Não	Não
Género	VARCHAR(20)	Não	Não	Não	Não
Data Nascimento	DATE	Não	Não	Não	Não

- **Entidade Universidade**

Tabela 5 - Caracterização dos atributos de Universidade

Atributos	Tipos de Dados	NULO	Composto	Multivalorado	Chave Primária
Id Universidade	INT	Não	Não	Não	Sim
Nome Universidade	VARCHAR(45)	Não	Não	Não	Não
Contacto	INT	Não	Não	Não	Não
País	VARCHAR(50)	Não	Não	Não	Não

- **Entidade Evento**

Tabela 6 - Caracterização dos atributos de Evento

Atributos	Tipos de Dados	NULO	Composto	Multivalorado	Chave Primária
Id Evento	INT	Não	Não	Não	Sim
Nome Evento	VARCHAR(45)	Não	Não	Não	Não
Local	VARCHAR(60)	Não	Não	Não	Não
Data_Hora	DATETIME	Não	Não	Não	Não

- **Entidade Modalidade**

Tabela 7 - Caracterização dos atributos de Modalidade

Atributos	Tipos de Dados	NULO	Composto	Multivalorado	Chave Primária
Id Modalidade	INT	Não	Não	Não	Sim
Nome Modalidade	VARCHAR(45)	Não	Não	Não	Não
Descrição	VARCHAR(100)	Não	Não	Não	Não
Género	VARCHAR(20)	Não	Não	Não	Não
Regras	VARCHAR(100)	Não	Não	Não	Não

- **Entidade Treinador**

Tabela 8 - Caracterização dos atributos de Treinador

Atributos	Tipos de Dados	NULO	Composto	Multivalorado	Chave Primária
Id Treinador	INT	Não	Não	Não	Sim
Nome Treinador	VARCHAR(45)	Não	Não	Não	Não
Contacto	INT	Não	Não	Não	Não
Email	VARCHAR(75)	Não	Não	Não	Não

- **Entidade Staff**

Tabela 9 - Caracterização dos atributos de Staff

Atributos	Tipos de Dados	NULO	Composto	Multivalorado	Chave Primária
Id Staff	INT	Não	Não	Não	Sim
Nome Staff	VARCHAR(45)	Não	Não	Não	Não
Tipo	VARCHAR(20)	Não	Não	Não	Não
Horário	VARCHAR(60)	Não	Não	Não	Não
Email	VARCHAR(75)	Sim	Não	Não	Não
Contacto	INT	Não	Não	Não	Não

- **Entidade Morada**

Atributos	Tipos de Dados	NULO	Composto	Multivalorado	Chave Primária
Id Morada	INT	Não	Não	Não	Sim
Rua	VARCHAR(45)	Não	Não	Não	Não
NrPorta	VARCHAR(20)	Sim	Não	Não	Não
CodPostal	VARCHAR(45)	Não	Não	Não	Não

3.5. Apresentação e explicação do diagrama ER produzido

Após a explicação de cada uma das entidades, dos relacionamentos entre si e os seus respetivos atributos, apresenta-se o modelo conceptual completo.

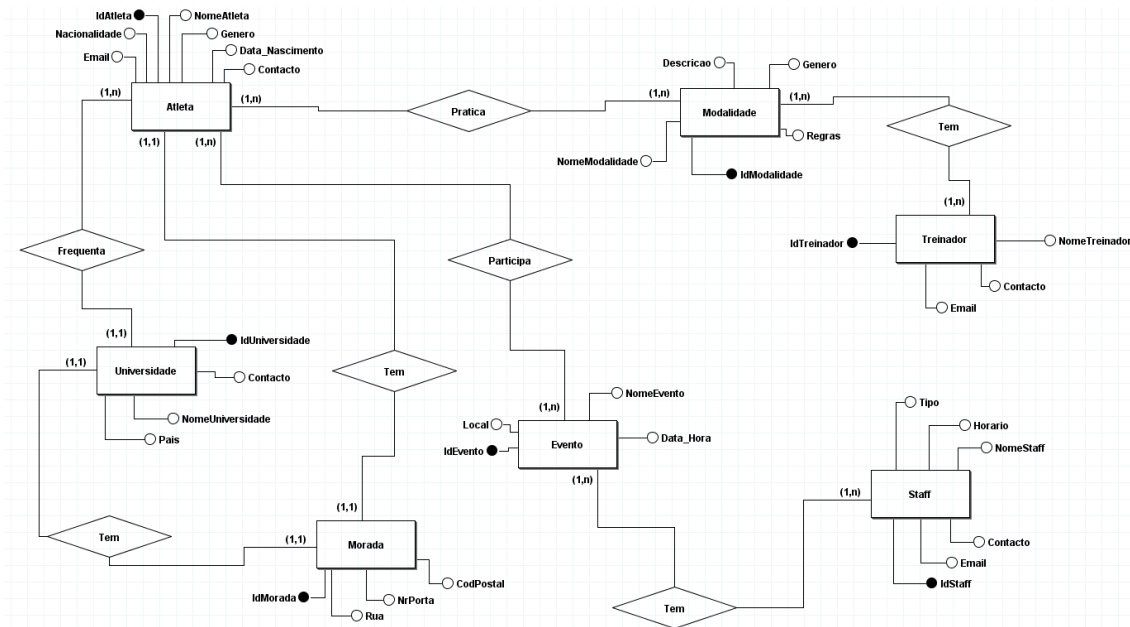


Figura 9 – Modelo Conceptual

4. Modelação Lógica

4.1. Construção e validação do modelo de dados Lógico

O modelo lógico foi construído a partir do modelo conceptual.

Nesta fase, convertemos as entidades e os relacionamentos em tabelas. Os identificadores de uma entidade passam a ser uma Chave Primária (PK), ou uma Chave Estrangeira (FK) caso esse identificador esteja presente noutra sob a forma de tabela. Feita a conversão, obtemos 10 tabelas, 7 originadas pelas entidades, 4 originadas pelos relacionamentos.

1) Atleta

- **Atributos:** Id Atleta (INT), Nome Atleta (VARCHAR(75)), Email (VARCHAR(75)), Contacto (INT), Género (VARCHAR(20)), Morada(VARCHAR(75)), Rua(VARCHAR(30)), Nr Porta (INT), Código Postal (INT), Data Nascimento (DATE), Nacionalidade (VARCHAR(50))
- **Chave Primária:** Id Atleta(INT)
- **Chave Estrangeira:** Id Universidade (INT), Id Morada(INT)

2) Universidade

- **Atributos:** Id Universidade (INT), Nome Universidade (VARCHAR(45)), Contacto (INT), Morada (VARCHAR(50)), País (VARCHAR(50))
- **Chave Primária:** Id Universidade (INT)
- **Chave Estrangeira:** Id Morada(INT)

3) Evento

- **Atributos:** Id Evento (INT), Nome Evento (VARCHAR(45)), Local (VARCHAR(60)), Data Hora (DATETIME)
- **Chave Primária:** Id Evento (INT)
- **Chave Estrangeira:** Não possui

4) Modalidade

- **Atributos:** Id Modalidade(INT), Nome Modalidade(VARCHAR(45)), Descrição(VARCHAR(100)), Género (VARCHAR(20)), Regras (VARCHAR(100))
- **Chave Primária:** Id Modalidade (INT)
- **Chave Estrangeira:** Não possui

5) Staff

- **Atributos:** Id Staff (INT), Nome Staff (VARCHAR(45)), Horario (VARCHAR(60)), Tipo (VARCHAR(20)), Contacto (INT), Email (VARCHAR(75))
- **Chave Primária:** Id Staff (INT)
- **Chave Estrangeira:** Não possui

6) Morada

- **Atributos:** Id Morada (INT), Rua (VARCHAR(45)), NrPorta (INT), CodPostal (VARCHAR(45))
- **Chave Primária:** Id Morada (INT)
- **Chave Estrangeira:** Não possui

7) Treinador

- **Atributos:** Id Treinador (INT), Nome Treinador (VARCHAR(45)), Contacto (INT), Email (VARCHAR(75))
- **Chave Primária:** Id Treinador (INT)
- **Chave Estrangeira:** Não possui

8) Atleta_Evento

- **Atributos:** Não possui
- **Chave Primária:** Não possui
- **Chave Estrangeira:** Id Atleta (INT), Id Evento (INT)

9) Atleta_Modalidade

- **Atributos:** Não possui
- **Chave Primária:** Não possui
- **Chave Estrangeira:** Id Atleta (INT), Id Modalidade (INT)

10)Evento_Staff

- **Atributos:** Não possui
- **Chave Primária:** Não possui
- **Chave Estrangeira:** Id Evento (INT), Id Staff (INT)

11) Modalidade_Treinador

- **Atributos:** Não possui
- **Chave Primária:** Não possui
- **Chave Estrangeira:** Id Modalidade (INT), Id Treinador (INT)

4.2. Normalização de Dados

A normalização de dados é uma prática fundamental em bancos de dados que visa eliminar a redundância de informações. Para validar essa normalização do modelo temos de verificar se satisfaz 3 regras, as FN.

A Primeira Fórmula Normal (1FN): Indica que os atributos têm de ser atômicos, isto é, as tabelas não podem ter valores repetidos nem atributos multivalorados.

A Segunda Fórmula Normal (2FN): Só pode ser cumprida se for cumprida a primeira, e indica que os atributos que não sejam chave primária devem depender apenas da chave primária dessa tabela.

A Terceira Fórmula Normal (3FN): Só pode ser cumprida se forem cumpridas as duas anteriores, e indica que numa tabela não pode haver nenhum atributo que seja gerado a partir de outro. Fazendo uma análise nas nossas tabelas, vemos que isso nunca acontece. Logo respeita a 3FN.

Analisando todas as regras de normalização, podemos concluir que o nosso modelo está normalizado.

4.3. Apresentação e explicação do modelo Lógico produzido

Após serem feitas as tabelas, uma por uma, definimos as relações entre elas.

Começamos por criar as tabelas das entidades com os seus respetivos atributos, e depois criamos novas tabelas que surgiram através dos relacionamentos de N-N.

Logo, surgiram quatro novas tabelas: Atleta_Evento que como chaves estrangeiras tem IdAtleta e IdEvento, Atleta_Modalidade que como chaves estrangeiras tem IdAtleta e IdModalidade, Evento_Staff que como chaves estrangeiras tem IdEvento e IdStaff e Modalidade_Treinador que como chaves estrangeiras tem IdModalidade e IdTreinador.

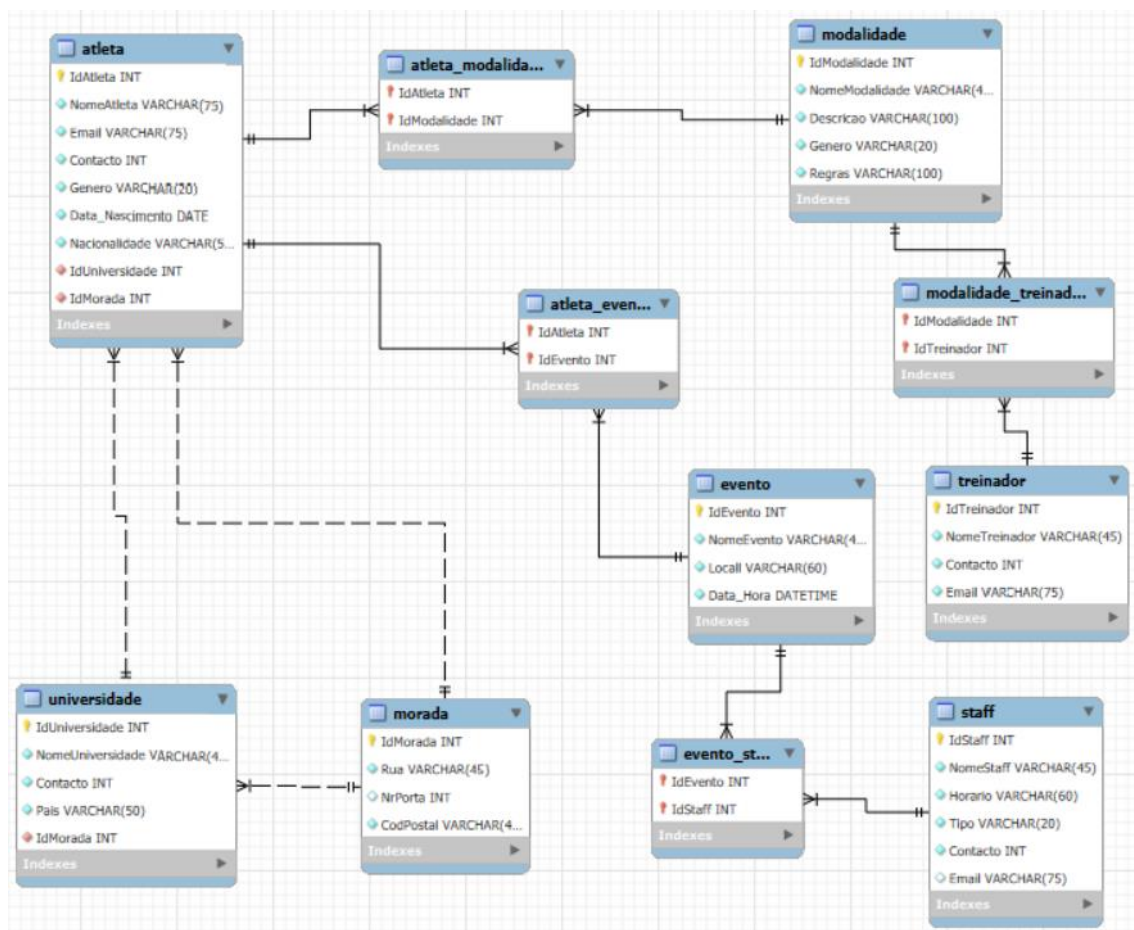


Figura 10 – Modelo Lógico

4.4. Validação do modelo com interrogações do utilizador

Para validar o modelo lógico foram seleccionadas algumas interrogações dos requisitos de exploração e usufruindo da álgebra relacional conseguimos mostrar de que forma o modelo a faz.

- Dado o nome de uma universidade saber quais são os seus atletas



Figura 11 – Árvore correspondente à primeira interrogação

Começamos por aceder às tabelas Atleta e Universidade unimo-las em uma só tabela nova r1, após isso selecionamos apenas o NomeUniversidade e projetamos só os que são 'Universidade do Minho'

- Dado o nome de uma modalidade saber quais são os treinadores dessa mesma

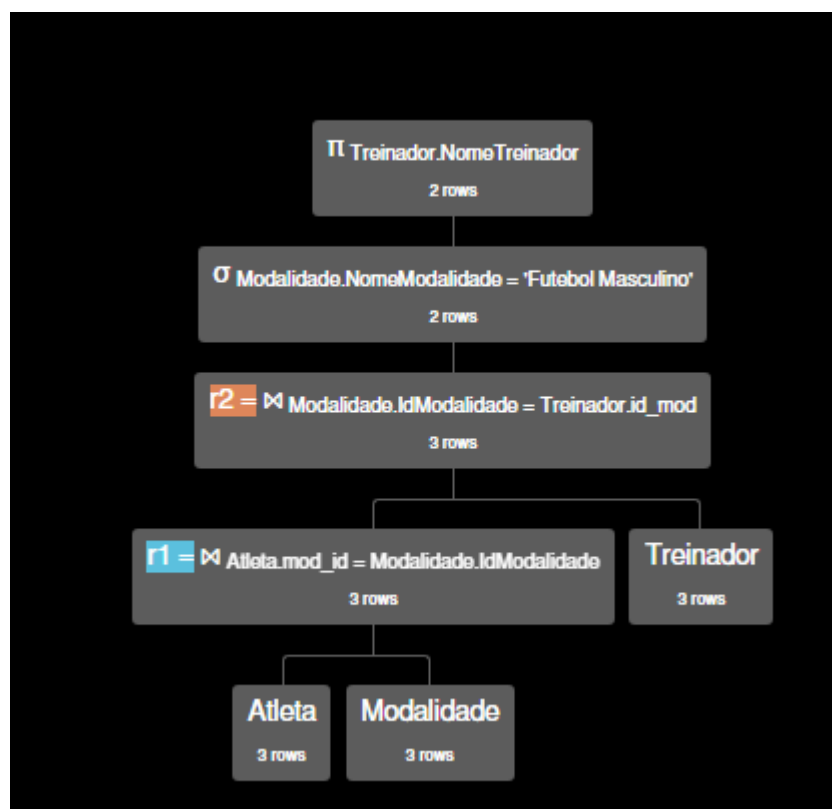


Figura 12 – Árvore correspondente à segunda interrogação

Começamos por aceder às tabelas Atleta e Modalidade unimo-las em uma só tabela nova r1, depois acedemos à tabelo Treinador e unimos estes dois como r2, após isso selecionamos apenas o NomeModalidade e projetamos só os que são 'Futebol Masculino' e projetamos todos os NomeTreinador

5. Implementação Física

5.1. Apresentação do processo de criação da base de dados

Alguns exemplos do processo de criação da base de dados em que eles concordaram estabelecer são:

- Criação da tabela Universidade

```
-- Criação da tabela "Universidade"
CREATE TABLE IF NOT EXISTS Universidade(
    IdUniversidade INT AUTO_INCREMENT,
    NomeUniversidade VARCHAR(45) NOT NULL,
    Contacto INT NOT NULL,
    Pais VARCHAR(50) NOT NULL,
    IdMorada INT NOT NULL,
    PRIMARY KEY(IdUniversidade),
    FOREIGN KEY (IdMorada) REFERENCES Morada(IdMorada)
);
-- Remover a tabela "Universidade"
-- DROP TABLE Universidade;
```

Figura 13 – Script de criação da tabela Universidade

- Criação da tabela Atleta

```
-- Criação da tabela "Atleta"
CREATE TABLE IF NOT EXISTS Atleta(
    IdAtleta INT AUTO_INCREMENT,
    NomeAtleta VARCHAR(75) NOT NULL,
    Email VARCHAR(75) NOT NULL ,
    Contacto INT NOT NULL ,
    Genero VARCHAR(20) NOT NULL,
    Data_Nascimento DATE NOT NULL,
    Nacionalidade VARCHAR(50) NOT NULL,
    IdUniversidade INT NOT NULL,
    IdMorada INT NOT NULL,
    PRIMARY KEY(IdAtleta),
    FOREIGN KEY (IdUniversidade) REFERENCES Universidade(IdUniversidade),
    FOREIGN KEY (IdMorada) REFERENCES Morada(IdMorada)
);
-- Remover a tabela "Atleta"
-- DROP TABLE Atleta;
```

Figura 14 – Script de criação da tabela Atleta

- Criação da tabela **Atleta_Evento**

```
-- Criação da tabela do relacionamento "Atleta - Evento"
CREATE TABLE IF NOT EXISTS Atleta_Evento (
    IdAtleta INT,
    IdEvento INT,
    PRIMARY KEY (IdAtleta, IdEvento),
    FOREIGN KEY (IdAtleta) REFERENCES Atleta(IdAtleta),
    FOREIGN KEY (IdEvento) REFERENCES Evento(IdEvento)
);
-- Remover a tabela "Atleta - Evento"
-- DROP TABLE Atleta_Evento;
```

Figura 15 – Script de criação da tabela Atleta-Evento

- Criação da tabela **Modalidade_Treinador**

```
-- Criação da tabela do relacionamento "Modalidade - Treinador"
CREATE TABLE IF NOT EXISTS Modalidade_Treinador(
    IdModalidade INT,
    IdTreinador INT,
    PRIMARY KEY (IdModalidade, IdTreinador),
    FOREIGN KEY (IdModalidade) REFERENCES Modalidade(IdModalidade),
    FOREIGN KEY (IdTreinador) REFERENCES Treinador(IdTreinador)
);
-- Remover a tabela "Modalidade - Treinador"
-- DROP TABLE Modalidade_Treinador;
```

Figura 16 – Script de criação da tabela Modalidade-Treinador

5.2. Cálculo do espaço da base de dados(inicial e taxa de crescimento anual)

Para calcular o espaço da Base de Dados, iremos estimar o tamanho, em bytes, de cada atributo de cada tabela. Seguimos a documentação Oficial do MySQL, em que o tipo de dados INT ocupa 4bytes, DATE ocupa 3bytes, DATETIME ocupa 8bytes e VARCHAR(X) ocupa X+1bytes.

- **Universidade**

Tabela 10 – Calculo dos bytes associados à entidade Universidade

Atributos	Tipos de Dados	Tamanho(Bytes)
IdUniversidade	INT	4
NomeUniversidade	VARCHAR(45)	46
Contacto	INT	4
Pais	VARCHAR(50)	51
Total	-	105
Crescimento Anual	-	7*105 = 735

- **Atleta**

Tabela 11 – Calculo dos bytes associados à entidade Atleta

Atributos	Tipos de Dados	Tamanho(Bytes)
IdAtleta	INT	4
NomeAtleta	VARCHAR(75)	76
Email	VARCHAR(75)	76
Contacto	INT	4
Genero	VARCHAR(20)	21
Data_Nascimento	DATE	3
Nacionalidade	VARCHAR(50)	51
Total	-	235
Crescimento Anual	-	50*235 = 11750

- **Morada**

Tabela 12 – Calculo dos bytes associados à entidade Morada

Atributos	Tipos de Dados	Tamanho(Bytes)
IdMorada	INT	4
Rua	VARCHAR(45)	46
NrPorta	INT	4
CodPostal	VARCHAR(45)	46
Total	-	100
Crescimento Anual	-	60 * 100 = 6000

- **Evento**

Tabela 13 – Calculo dos bytes associados à entidade Evento

Atributos	Tipos de Dados	Tamanho(Bytes)
IdEvento	INT	4
NomeEvento	VARCHAR(45)	46
Locall	VARCHAR(60)	61
Data_Hora	DATETIME	8
Total	-	119
Crescimento Anual	-	18 * 119 = 2142

- **Staff**

Tabela 14 – Calculo dos bytes associados à entidade Staff

Atributos	Tipos de Dados	Tamanho(Bytes)
IdStaff	INT	4
NomeStaff	VARCHAR(45)	46
Horario	VARCHAR(60)	61
Tipo	VARCHAR(20)	21
Contacto	INT	4
Email	VARCHAR(75)	76
Total	-	212
Crescimento Anual	-	15 * 212 = 3180

- **Modalidade**

Tabela 15 – Calculo dos bytes associados à entidade Modalidade

Atributos	Tipos de Dados	Tamanho(Bytes)
IdModalidade	INT	4
NomeModalidade	VARCHAR(45)	46
Descricao	VARCHAR(100)	101
Genero	VARCHAR(20)	21
Regras	VARCHAR(100)	101
Total	-	273
Crescimento Anual	-	18 * 273 = 4914

- **Treinador**

Tabela 16 – Calculo dos bytes associados à entidade Treinador

Atributos	Tipos de Dados	Tamanho(Bytes)
IdTreinador	INT	4
NomeTreinador	VARCHAR(45)	46
Contacto	INT	4
Email	VARCHAR(75)	76
Total	-	130
Crescimento Anual	-	15 * 130 = 1950

- **Atleta_Evento**

Tabela 17 – Calculo dos bytes associados à entidade Atleta_Evento

Atributos	Tipos de Dados	Tamanho(Bytes)
IdAtleta	INT	4
IdEvento	INT	4
Total	-	8
Crescimento Anual	-	110 * 8 = 880

- **Evento_Staff**

Tabela 18 – Calculo dos bytes associados à entidade Evento_Staff

Atributos	Tipos de Dados	Tamanho(Bytes)
IdEvento	INT	4
IdStaff	INT	4
Total	-	8
Crescimento Anual	-	115 * 8 = 920

- **Atleta_Modalidade**

Tabela 19 – Calculo dos bytes associados à entidade Atleta_Modalidade

Atributos	Tipos de Dados	Tamanho(Bytes)
IdAtleta	INT	4
IdModalidade	INT	4
Total	-	8
Crescimento Anual	-	110 * 8 = 880

- **Modalidade_Treinador**

Tabela 20 – Calculo dos bytes associados à entidade Modalidade_Treinador

Atributos	Tipos de Dados	Tamanho(Bytes)
IdModalidade	INT	4
IdTreinador	INT	4
Total	-	8
Crescimento Anual	-	18 * 8 = 144

- **Crescimento Anual Total**

Após o calculo de todas as tabelas o tamanho total da nossa base de dados seria, sem povoamento, 1206 bytes.

No entanto, para uma estimativa real, tendo em conta o povoamento do nosso modelo seria um total de:

$$7*105 + 50*235 + 60 * 100 + 18 * 119 + 15 * 212 + 18 * 273 + 15 * 130 + 110 * 8 + 115 * 8 + 110 * 8 + 18 * 8 = 33495 \text{ bytes}$$

5.3. Realização do povoamento da base de dados

```
INSERT INTO Universidade (IdUniversidade, NomeUniversidade, Contacto, Pais, IdMorada)
VALUES
(1,'Universidade do Minho',253601109,'Portugal',1),
(2,'Universidade de Harvard',255601109,'Estados Unidos',2),
(3,'Universidade de Cambridge',257601109,'Inglaterra',3),
(4,'Universidade de Tokyo',259601109,'Japão',4),
(5,'Universidade de Peking',260601109,'China',5),
(6,'Universidade Nacional de Singapura',263601109,'Singapura',6),
(7,'Universidade de Munich',258601109,'Alemanha',7);
```

Figura 17 – Povoamento de Universidade

```

INSERT INTO Evento (IdEvento, NomeEvento, Local, Data_Hora)
VALUES
(1, 'Natação Feminina', 'Ginásio 5', '2024-06-12 15:30:00'),
(2, 'Natação Masculina', 'Ginásio 5', '2024-06-13 15:00:00'),
(3, 'Volleyball Feminina', 'Ginásio 7', '2024-06-14 09:00:00'),
(4, 'Volleyball Masculino', 'Ginásio 7', '2024-06-15 10:30:00'),
(5, 'BasketBall Feminino', 'Ginásio 1', '2024-06-16 15:30:00'),
(6, 'BasketBall Masculino', 'Ginásio 2', '2024-06-17 15:30:00'),
(7, 'Atletismo 100m Masculino', 'Estádio Olímpico', '2024-06-18 10:00:00'),
(8, 'Atletismo 200m Masculino', 'Estádio Olímpico', '2024-06-18 11:00:00'),
(9, 'Futebol Feminino', 'Campo Principal', '2024-06-19 14:00:00'),
(10, 'Futebol Masculino', 'Campo Principal', '2024-06-19 16:00:00'),
(11, 'Hóquei Feminino', 'Ginásio 2', '2024-06-20 12:00:00'),
(12, 'Hóquei Masculino', 'Ginásio 2', '2024-06-20 14:00:00'),
(13, 'Ginástica Artística Feminina', 'Ginásio 3', '2024-06-21 09:00:00'),
(14, 'Ginástica Artística Masculina', 'Ginásio 3', '2024-06-21 11:00:00'),
(15, 'Ténis de Mesa Feminino', 'Ginásio 1', '2024-06-22 15:00:00'),
(16, 'Ténis de Mesa Masculino', 'Ginásio 1', '2024-06-22 17:00:00'),
(17, 'Atletismo 100m Feminino', 'Estádio Olímpico', '2024-06-18 15:00:00'),
(18, 'Atletismo 200m Feminino', 'Estádio Olímpico', '2024-06-18 16:00:00');

```

Figura 18 – Povoamento de Evento

```

INSERT INTO Staff (IdStaff, NomeStaff, Horario, Tipo, Contacto, Email)
VALUES
(1, 'Ricardo Pereira', '8H-16H', 'Segurança', '923567890', 'ricardopereira@gmail.com'),
(2, 'Beatriz Mendes', '9H-18H', 'Recepcionista', '924678901', 'beatrizmendes@gmail.com'),
(3, 'João Carvalho', '7H-15H', 'Técnico', '925789012', 'joaocarvalho@gmail.com'),
(4, 'Patrícia Gomes', '12H-20H', 'Limpeza', '926890123', 'patriciagomes@gmail.com'),
(5, 'Miguel Fonseca', '6H-14H', 'Manutenção', '927901234', 'miguelfonseca@gmail.com'),
(6, 'Carolina Dias', '14H-22H', 'Funcionária', '928012345', 'carolinadias@gmail.com'),
(7, 'Fábio Silva', '10H-18H', 'Auxiliar', '929123456', 'fabiosilva@gmail.com'),
(8, 'Sara Fernandes', '15H-23H', 'Segurança', '930234567', 'sarafernandes@gmail.com'),
(9, 'André Matos', '9H-17H', 'Técnico', '931345678', 'andrematos@gmail.com'),
(10, 'Diana Oliveira', '13H-21H', 'Recepcionista', '932456789', 'dianaoliveira@gmail.com'),
(11, 'Rafael Nogueira', '8H-16H', 'Manutenção', '933567890', 'rafaelnogueira@gmail.com'),
(12, 'Mariana Costa', '9H-18H', 'Limpeza', '934678901', 'marianacosta@gmail.com'),
(13, 'Pedro Lopes', '7H-15H', 'Segurança', '935789012', 'pedrolopes@gmail.com'),
(14, 'Camila Rocha', '12H-20H', 'Auxiliar', '936890123', 'camilarocha@gmail.com'),
(15, 'Bruno Tavares', '10H-18H', 'Funcionário', '937901234', 'brunotavares@gmail.com');

```

Figura 19 – Povoamento de Staff

5.4. Tradução das interrogações do utilizador para SQL (alguns exemplos)

Para mostrar que o modelo físico desenvolvido se encontra válido os requisitos foram convertidos em interrogações e traduzidos para SQL.

- **Mostrar o número de atletas Masculinos**

```
-- 1. Mostrar o número de atletas Masculinos:
SELECT COUNT(*) AS NumeroDeAtletasMasculinos
FROM Atleta
WHERE Genero = 'M';
```

Figura 20 – Primeira interrogação: Número de atletas Masculinos

	NumeroDeAtletasMasculinos
▶	26

Figura 21 – resultado da primeira interrogação

- **Listar uma data especifica de um certo evento (exemplo: Volleyball Feminina)**

```
-- 3. Listar uma data especifica de um certo evento (exemplo: Volleyball Feminina):
SELECT Data_Hora
FROM evento
WHERE NomeEvento = 'Volleyball Feminina';
```

Figura 22 – Segunda interrogação: Data de um Evento específico

	Data_Hora
▶	2024-06-14 09:00:00

Figura 23 – resultado da segunda interrogação

- **Dado o nome de uma universidade saber quais são os seus atletas (exemplo: Universidade de Cambridge)**

```
-- 4. dado o nome de uma universidade saber quais são os seus atletas (exemplo: Universidade de Cambridge):
SELECT a.IdAtleta, a.NomeAtleta, a.Genero, a.Nacionalidade, a.Email
FROM universidade u
JOIN atleta a ON u.IdUniversidade = a.IdUniversidade
WHERE u.NomeUniversidade = 'Universidade de Cambridge';
```

Figura 24 – Terceira interrogação: Quais são os atletas de uma Universidade

	IdAtleta	NomeAtleta	Genero	Nacionalidade	Email
▶	6	Michael Brown	M	Ingles	michaelbrown@gmail.com
	14	Laura Taylor	F	Ingles	laurataylor@gmail.com
	23	Daniel Cooper	M	Ingles	danielcooper@gmail.com
	27	Oliver Davis	M	Ingles	oliverdavis@gmail.com
	34	Isabella Lewis	F	Ingles	isabellalewis@gmail.com
	39	Mia Clarke	F	Ingles	miac Clarke@gmail.com
	47	Emma Watson	F	Ingles	emmawatson@gmail.com

Figura 25 – resultado da terceira interrogação

- **Saber a nacionalidade com mais atletas**

```
-- 6. permitir saber a nacionalidade com mais atletas :
SELECT Nacionalidade, COUNT(*) AS TotalAtletas
FROM atleta
GROUP BY Nacionalidade
ORDER BY TotalAtletas DESC
LIMIT 1;
```

Figura 26 – Quarta interrogação: Nacionalidade com mais Atletas

	Nacionalidade	TotalAtletas
▶	Portugues	15

Figura 27 – resultado da quarta interrogação

5.5. Definição e caracterização das vistas de utilização em SQL (alguns exemplos)

- **View sobre o número de participantes e de Staffs em cada Evento**

```
-- 2. View para visualizar o numero de participantes e de staffs em cada evento:
CREATE VIEW DetalhesEventos AS
SELECT
    E.IdEvento,
    E.NomeEvento,
    E.Local,
    E.Data_Hora,
    COUNT(DISTINCT AE.IdAtleta) AS TotalParticipantes,
    COUNT(DISTINCT ES.IdStaff) AS TotalStaff
FROM
    Evento E
LEFT JOIN
    Atleta_Evento AE
ON
    E.IdEvento = AE.IdEvento
LEFT JOIN
    Evento_Staff ES
ON
    E.IdEvento = ES.IdEvento
GROUP BY
    E.IdEvento, E.NomeEvento, E.Local, E.Data_Hora;

SELECT * FROM DetalhesEventos;
```

Figura 28 – Primeira View

	IdEvento	NomeEvento	Local	Data_Hora	TotalParticipantes	TotalStaff
▶	1	Natação Feminina	Ginásio 5	2024-06-12 15:30:00	9	7
	2	Natação Masculina	Ginásio 5	2024-06-13 15:00:00	9	7
	3	Volleyball Feminina	Ginásio 7	2024-06-14 09:00:00	9	7
	4	Volleyball Masculino	Ginásio 7	2024-06-15 10:30:00	9	7
	5	BasketBall Feminino	Ginásio 1	2024-06-16 15:30:00	6	7
	6	BasketBall Masculino	Ginásio 2	2024-06-17 15:30:00	7	7
	7	Atletismo 100m Masculino	Estádio Olímpico	2024-06-18 10:00:00	6	5
	8	Atletismo 200m Masculino	Estádio Olímpico	2024-06-18 11:00:00	6	5
	9	Futebol Feminino	Campo Principal	2024-06-19 14:00:00	6	7
	10	Futebol Masculino	Campo Principal	2024-06-19 16:00:00	6	7
	11	Hóquei Feminino	Ginásio 2	2024-06-20 12:00:00	6	6
	12	Hóquei Masculino	Ginásio 2	2024-06-20 14:00:00	7	6
	13	Ginástica Artística Feminina	Ginásio 3	2024-06-21 09:00:00	4	6
	14	Ginástica Artística Masculina	Ginásio 3	2024-06-21 11:00:00	4	6
	15	Tênis de Mesa Feminino	Ginásio 1	2024-06-22 15:00:00	4	6
	16	Tênis de Mesa Masculino	Ginásio 1	2024-06-22 17:00:00	4	6
	17	Atletismo 100m Feminino	Estádio Olímpico	2024-06-18 15:00:00	4	6
	18	Atletismo 200m Feminino	Estádio Olímpico	2024-06-18 16:00:00	4	6

Figura 29 – resultado da Primeira View

Esta vista permite a visualização do Evento, com o número de Atletas participantes nele e o número dos Staffs que iram trabalhar nele, de uma forma simples e com mais “cosmética”.

- **View que mostra quais universidades têm atletas a participar no maior número de modalidades**

```
-- 4. View que mostra quais universidades têm atletas a participar no maior número de modalidades:
CREATE VIEW UniversidadesMaisModalidades AS
SELECT
    U.NomeUniversidade,
    COUNT(DISTINCT am.IdModalidade) AS TotalModalidades
FROM
    universidade U
JOIN
    Atleta A ON U.IdUniversidade = A.IdUniversidade
JOIN
    Atleta_Modalidade AM ON A.IdAtleta = AM.IdAtleta
GROUP BY
    U.NomeUniversidade
ORDER BY
    TotalModalidades DESC;

SELECT * FROM UniversidadesMaisModalidades;
```

Figura 30 – Segunda View

	NomeUniversidade	TotalModalidades
►	Universidade do Minho	18
	Universidade de Tokyo	14
	Universidade de Cambridge	13
	Universidade de Harvard	13
	Universidade de Munich	10
	Universidade de Peking	8
	Universidade Nacional de Singapura	4

Figura 31 – resultado da Segunda View

Esta view apresenta uma forma organizada quais universidades têm atletas a participar no maior número de modalidades.

5.6. Definição dos perfis de utilização para cada utilizador da base de dados

```
-- Perfis de utilização

CREATE USER 'admin'@'localhost' IDENTIFIED BY 'password_admin';
CREATE USER 'gestor'@'localhost' IDENTIFIED BY 'password_gestor';
CREATE USER 'treinador'@'localhost' IDENTIFIED BY 'password_treinador';
CREATE USER 'staff'@'localhost' IDENTIFIED BY 'password_staff';
CREATE USER 'aluno'@'localhost' IDENTIFIED BY 'password_aluno';

GRANT ALL PRIVILEGES ON *.* TO 'admin'@'localhost';

-- Gestor pode ver e alterar
GRANT SELECT, UPDATE ON sportorg.Atleta TO 'gestor'@'localhost';
GRANT SELECT, UPDATE ON sportorg.Evento TO 'gestor'@'localhost';
GRANT SELECT, UPDATE ON sportorg.Modalidade TO 'gestor'@'localhost';
GRANT SELECT, UPDATE ON sportorg.Universidade TO 'gestor'@'localhost';

-- Staff pode apenas ver
GRANT SELECT ON sportorg.Evento TO 'staff'@'localhost';
GRANT SELECT ON sportorg.Staff TO 'staff'@'localhost';

-- Treinador pode apenas ver
GRANT SELECT ON sportorg.Atleta TO 'treinador'@'localhost';
GRANT SELECT ON sportorg.Modalidade TO 'treinador'@'localhost';
```

Figura 32 – Definição dos Perfis1

```
-- Se um aluno deve acessar apenas seus próprios dados, você pode usar views
CREATE VIEW aluno_view AS
SELECT *
FROM Atleta
WHERE IdAtleta = (SUBSTRING_INDEX(USER(), '@', 1));
-- SUBSTRING_INDEX Obtém o nome do usuário logado (por exemplo, aluno1) e filtra os dados correspondentes ao ID do aluno.

GRANT SELECT ON aluno_view TO 'aluno'@'localhost';

-- Remove as permissões de leitura e atualização na tabela atletas (exemplo) para o usuário gestor
REVOKE SELECT, UPDATE ON sportorg.Atleta FROM 'gestor'@'localhost';

-- Após criar ou alterar permissões, é necessário atualizar os privilégios no servidor
FLUSH PRIVILEGES;
```

Figura 33 – Definição dos Perfis2

5.7. Indexação do Sistema de Dados

```
-- Indexes
-- Melhora buscas por nome e nacionalidade juntos :
CREATE INDEX idx_atleta_nome ON Atleta(NomeAtleta);
-- DROP INDEX idx_atleta_nome ON Atleta(NomeAtleta);

-- Otimiza buscas por eventos em um local e data específica :
CREATE INDEX idx_evento_nome_data ON Evento(Local, Data_Hora);
-- DROP INDEX idx_evento_nome_data ON Evento(Local, Data_Hora);
```

Figura 34 – Indexes

5.8. Procedimentos Implementados

- Procedure de adicionar um Atleta novo

```
-- 1. Procedure de adicionar um Atleta novo
-- DROP PROCEDURE AdicionarAtleta;
DELIMITER $$

CREATE PROCEDURE AdicionarAtleta(
    IN p_NomeAtleta VARCHAR(75),
    IN p_Nacionalidade VARCHAR(50),
    IN p_DataNascimento DATE,
    IN p_Email VARCHAR(100),
    IN p_Contacto INT,
    IN p_IdUniversidade INT,
    IN p_Genero VARCHAR(20)
)
BEGIN
    INSERT INTO atleta (NomeAtleta, Nacionalidade, Data_Nascimento, Email, Contacto, IdUniversidade, Genero)
    VALUES (p_NomeAtleta, p_Nacionalidade, p_DataNascimento, p_Email, p_Contacto, p_IdUniversidade, p_Genero);
END$$

DELIMITER ;
```

Figura 35 – Primeira Procedure

```

CALL AdicionarAtleta(
    'João Silva',
    'Portugues',
    '1999-06-12',
    'joaosilva@email.com',
    987654321,
    1,
    'M'
);

SELECT * FROM Atleta;

```

Figura 36 – Primeira Procedure2

- Procedure de remover um atleta da base de dados, incluindo os registos associados

```

-- 2. Remove um atleta da base de dados, incluindo os registos associados:

DELIMITER $$

CREATE PROCEDURE RemoverAtleta(
    IN p_IdAtleta INT
)
BEGIN
    -- Remove as associações com eventos
    DELETE FROM atleta_evento WHERE IdAtleta = p_IdAtleta;

    -- Remove associações com modalidades
    DELETE FROM atleta_modalidade WHERE IdAtleta = p_IdAtleta;

    -- Remove o próprio atleta
    DELETE FROM atleta WHERE IdAtleta = p_IdAtleta;
END$$

DELIMITER ;

CALL RemoverAtleta (
    51
);

SELECT * FROM Atleta;

```

Figura 37 – Segunda Procedure

- **Procedure de estabelecer a relação entre um treinador e uma modalidade nova**

```
-- 3. Estabelece a relação entre um treinador e uma modalidade:

DELIMITER $$

CREATE PROCEDURE AssociarTreinadorAModalidade(
    IN p_IdTreinador INT,
    IN p_IdModalidade INT
)
BEGIN
    INSERT INTO modalidade_treinador (IdTreinador, IdModalidade)
    VALUES (p_IdTreinador, p_IdModalidade);
END$$

DELIMITER ;

CALL AssociarTreinadorAModalidade(
    10,
    3
);

SELECT * FROM Modalidade_Treinador;
```

Figura 38 – Terceira Procedure

5.9. Plano de segurança e recuperação de dados

Para que uma base de dados seja utilizada de forma eficaz, é essencial garantir certos aspetos, como a disponibilidade, a integridade e a confidencialidade dos dados armazenados.

- **Confidencialidade:** A base de dados contém informações sensíveis, como dados de clientes e detalhes de encomendas. Para proteger essas informações contra acessos não autorizados, é crucial manter os dados encriptados. Assim, mesmo em caso de invasão dos servidores, os dados permanecerão inacessíveis. O MySQL oferece a funcionalidade de “criptografia de chave pública” como uma solução eficaz para garantir a confidencialidade das informações.
- **Integridade:** Para assegurar a integridade da base de dados, é necessário implementar uma prática fundamental: *Backups* regulares: É imprescindível realizar cópias de segurança frequentes dos dados para evitar perdas significativas. Os *backups* também devem seguir os mesmos padrões de segurança aplicados à base de dados principal. O comando “*mysqldump*”, disponível no MySQL, é ferramenta usada para este propósito.

```
-- Para o backup manual, abrir cmd como admin
-- ir até a pasta aonde está o mysql bin
-- código: cd C:\Program Files\MySQL\MySQL Server 9.0
-- cd bin
-- mysqldump -u admin -p sportorg > backup_sportorg.sql
-- password: password_admin
-- e cria o ficheiro backup.
```

Figura 39 – *Backup*

- **Disponibilidade:** O número de utilizadores com permissões para modificar a base de dados deve ser reduzido ao mínimo necessário. Além disso, as permissões devem ser configuradas de forma restritiva, garantindo que os utilizadores possam desempenhar suas funções sem comprometer a segurança. Para tal, recomenda-se a criação de contas de utilizadores, com as permissões ajustadas por meio dos comandos “GRANT” e “REVOKE”.

6. Conclusões e Trabalho Futuro

Para concluir, consideramos que os objetivos principais que nos foram propostos para este projeto foram alcançados.

No entanto, às vezes encontramos alguns obstáculos pelo caminho mas a nosso ver eles foram resolvidos.

Sentimos que o trabalho foi uma maneira muito importante para melhorar a nossa compreensão sobre a matéria e sobre a utilização da plataforma MySQL.

Ficou também evidente que existe espaço para melhorar.

7. Referências Bibliográficas

Ferramentas Utilizadas:

- Brmodelo;
- MYSQL Workbench;
- Microsoft Word;
- Relax;
- Microsoft Excel;